New Features Guide

FortiOS 7.4.0

FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO LIBRARY
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/training-certification

FORTINET TRAINING INSTITUTE
https://training.fortinet.com

FORTIGUARD LABS
https://www.fortiguard.com

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

FORTINET®

April 2, 2024
FortiOS 7.4.0 New Features Guide
01-740-876933-20240417

TABLE OF CONTENTS

Change Log

Date	Change Description
2024-04-17	Updated Configuring FortiClient EMS and FortiClient EMS Cloud on a per-VDOM basis on page 688.
2024-04-02	Updated Configuring FortiClient EMS and FortiClient EMS Cloud on a per-VDOM basis on page 688 and FGCP multi-version cluster upgrade 7.4.1 on page 636.
2024-03-14	Added Unauthorized firmware modification attempt reporting 7.4.1 on page 684.
2024-03-11	Added Specify how RADIUS request attributes are formatted 7.4.2 on page 590.
2024-03-07	Added Customizing the FortiFlex license token activation retry parameters 7.4.2 on page 729 and Set the priority for dynamic or egress VLAN assignment 7.4.2 on page 589.
2024-03-06	Added SAML-based authentication for FortiClient remote access dialup IPsec VPN clients on page 460.
2024-02-27	Added Allow multiple Netflow collectors 7.4.2 on page 119.
2024-02-21	Updated Support for LAN extension VDOM simplifications 7.4.2 on page 116.
2024-02-15	Updated Prevent firmware upgrade depending on the current firmware license's expiration date 7.4.2 on page 626 and Automatic firmware upgrade enhancements 7.4.1 on page 615.
2024-02-09	Added Prevent firmware upgrade depending on the current firmware license's expiration date 7.4.2 on page 626.
2024-02-08	Added Memory usage reduced on FortiGate models with 2 GB RAM 7.4.2 on page 625.
2024-02-07	Initial release of FortiOS 7.4.3.
2024-01-12	Updated Optimize virtual patching on the local-in interface 7.4.2 on page 312, Add SNMP trap for memory usage on FortiGates 7.4.2 on page 644, and Introduce new log fields for long-live sessions 7.4.2 on page 710.
2024-01-11	Added Configure Purdue Levels for Fabric devices 7.4.2 on page 704.
2024-01-09	Added Support multiple compartments and regions with single OCI SDN connector on page 721.
2023-12-20	Initial release of FortiOS 7.4.2.
2023-12-07	Updated Improve DVLAN QinQ performance for NP7 platforms over virtual wire pairs on page 64.
2023-12-06	Added View batch transaction commands through the REST API 7.4.1 on page 619.
2023-11-20	Added Support for WPA3 security modes on FortiWiFi units operating in Client Mode on page 515.
2023-11-15	Added Support domain name in XFF with ICAP 7.4.1 on page 413.

Date	Change Description
2023-11-09	Added Support port block allocation for NAT64 on page 300.
2023-10-12	Updated Support inter-VLAN routing by managed FortiSwitch units 7.4.1 on page 574 and Support security rating recommendations for tier-2 and tier-3 MCLAGs 7.4.1 on page 577.
2023-10-10	Updated Add built-in entropy source 7.4.1 on page 682.
2023-10-04	Updated Synchronize the FortiOS interface description with the FortiSwitch VLAN description 7.4.1 on page 585.
2023-09-26	Added Synchronize the FortiOS interface description with the FortiSwitch VLAN description 7.4.1 on page 585.
2023-09-18	Added Support automatic federated firmware updates of managed FortiAPs and FortiSwitches 7.4.1 on page 661.
2023-09-14	Added Support LTE / BLE airplane mode for FGR-70F-3G4G 7.4.1 on page 105.
2023-09-11	Updated Improve the performance of the GUI policy list on page 295.
2023-09-08	Added Configuring multiple DDNS entries in the GUI on page 74.
2023-09-07	Updated Active dynamic BGP neighbor triggered by ADVPN shortcut 7.4.1 on page 236 and Support inter-VLAN routing by managed FortiSwitch units 7.4.1 on page 574.
2023-09-04	Added Support for the authentication and encryption of fabric links 7.4.1 on page 581 and Support security rating recommendations for tier-2 and tier-3 MCLAGs 7.4.1 on page 577.
2023-09-01	Added Securely exchange serial numbers between FortiGates connected with IPsec VPN 7.4.1 on page 469.
2023-08-31	Initial release of FortiOS 7.4.1.
2023-07-24	Updated Enhance BIOS-level signature and file integrity checking on page 676. Added Real-time file system integrity checking on page 680.
2023-07-19	Updated Active SIM card switching available on FortiGates with cellular modem and dual SIM card support on page 64.
2023-07-05	Updated Using MP-BGP EVPN with VXLAN on page 47.
2023-06-30	Added Support the new AWS c7gn instance family on page 718.
2023-06-07	Added Support OT and IoT virtual patching on NAC policies on page 378.
2023-05-31	Added FGCP HA between FortiGates of the same model with different AC and DC PSUs on page 627.
2023-05-17	Added Explicit proxy logging enhancements on page 138.
2023-05-12	Updated IPsec SA key retrieval from a KMS server using KMIP on page 446.
2023-05-11	Initial release.

Overview

This guide provides details of new features introduced in FortiOS 7.4. For each feature, the guide provides detailed information on configuration, requirements, and limitations, as applicable. Features are organized into the following sections:

- GUI
- Network
- SD-WAN
- Policy and objects
- Zero Trust Network Access
- Security profiles
- VPN
- User and authentication
- LAN Edge
- System
- Security Fabric
- Log and report
- Cloud
- Operational Technology

For features introduced in 7.4.1 and later versions, the version number is appended to the end of the topic heading. For example, GUI enhancements for FortiGuard DLP service 7.4.1 on page 28 was introduced in 7.4.1. If a topic heading has no version number at the end, the feature was introduced in 7.4.0.

For a list of features organized by version number, see Index on page 734.

GUI

This section includes information about FortiOS GUI related new features:

General usability enhancements

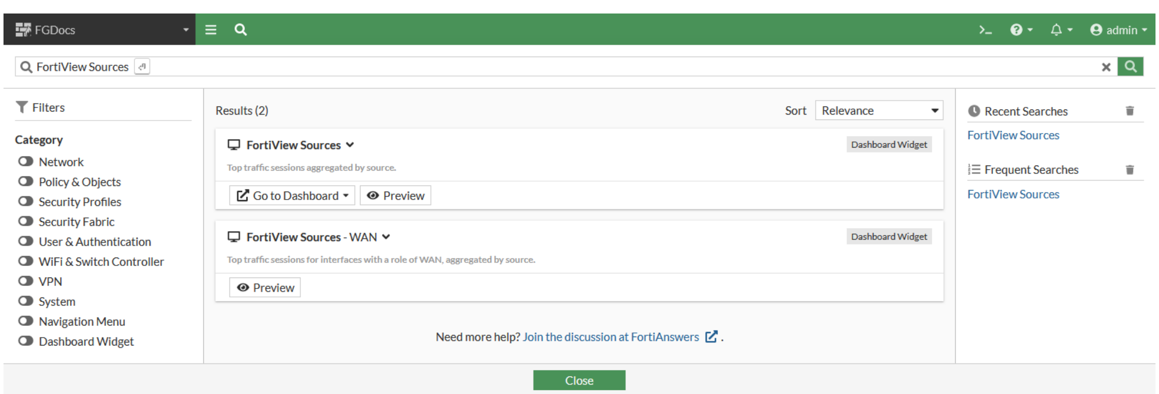
This section includes new features related to general usability enhancements:

Updated Dashboard and FortiView

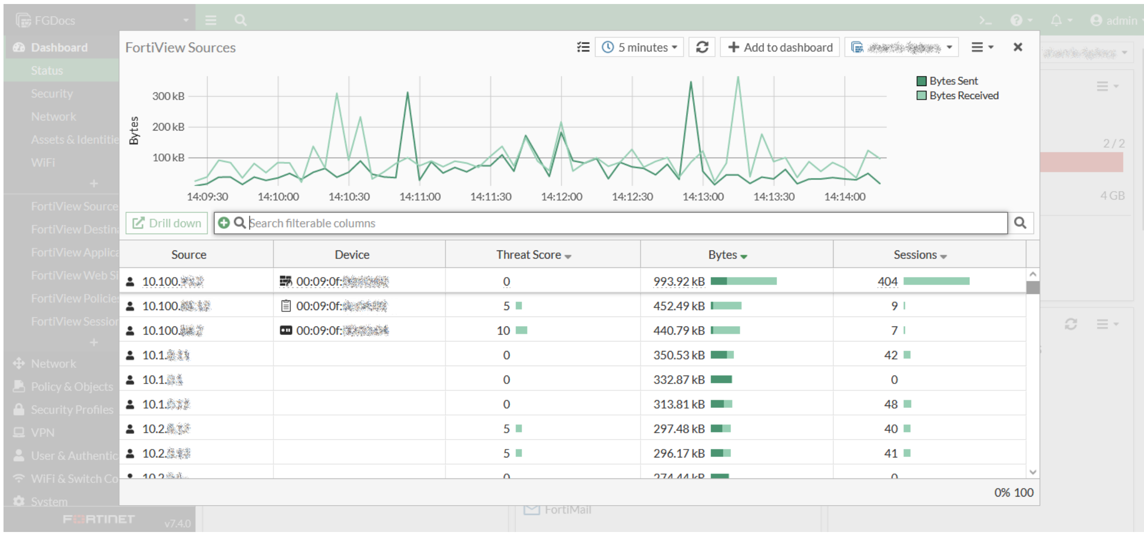
Dashboard widgets and FortiView monitors are updated with new graphs, faster performance, and other updates that improve the user experience.

FortiView

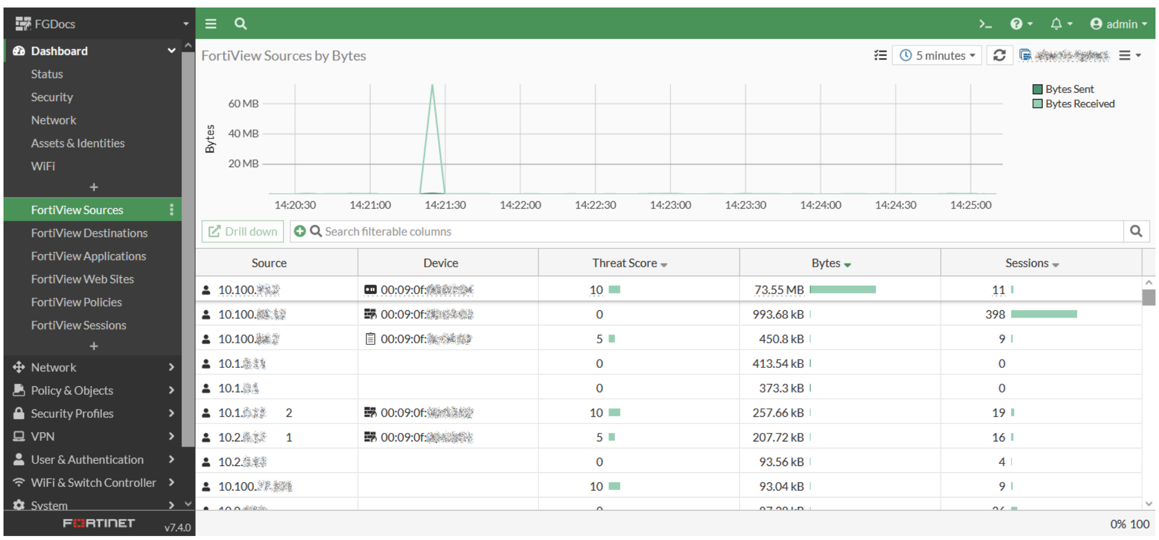
FortiView pages can be found using the global search.



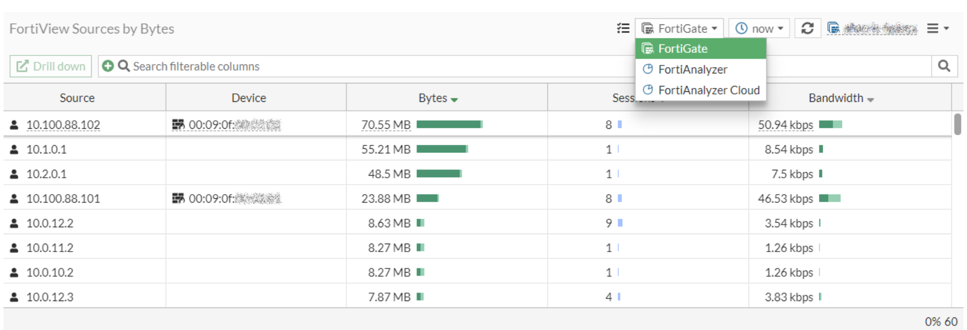
Click the *Preview* button to preview the page.



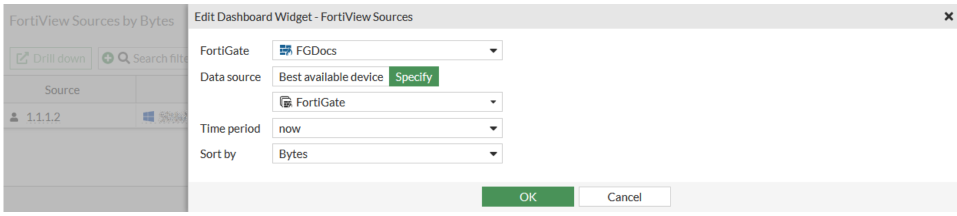
Click *Go to Dashboard* to go to that dashboard. The button is not available if that dashboard has not been added.



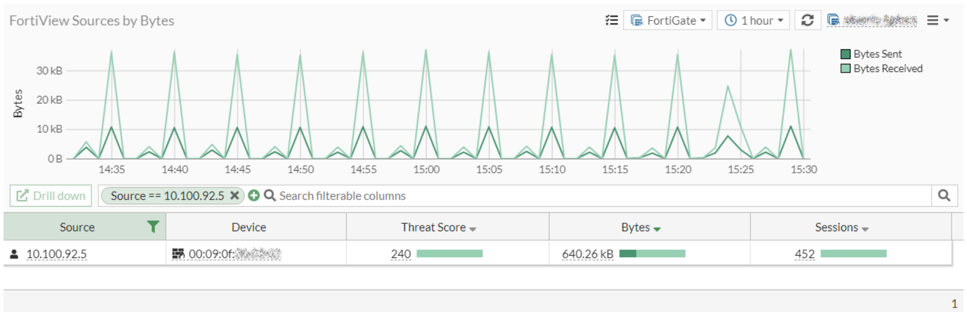
On a standalone FortiGate, the FortiView data source can be selected from the drop down.



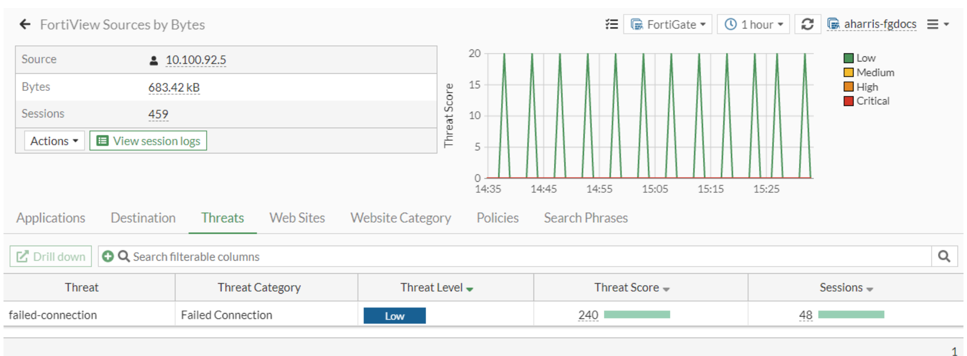
Only the table view is supported; there are no visualization settings.



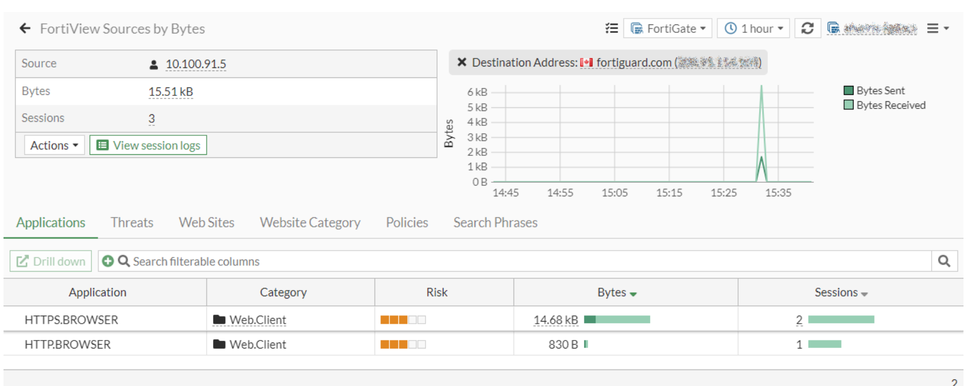
Filters can be applied using the filter bar or column headings, and are not cleared after refreshing the page or logging out then back in.



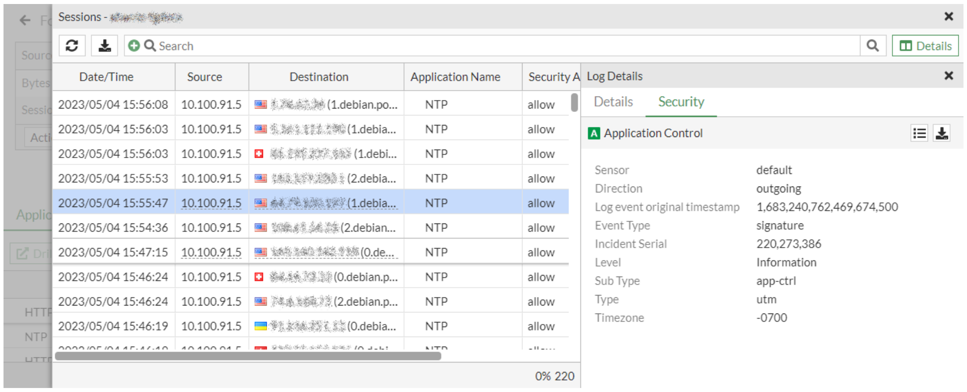
Drill down on any entry by double-clicking on it, or selecting it and clicking *Drill down*.



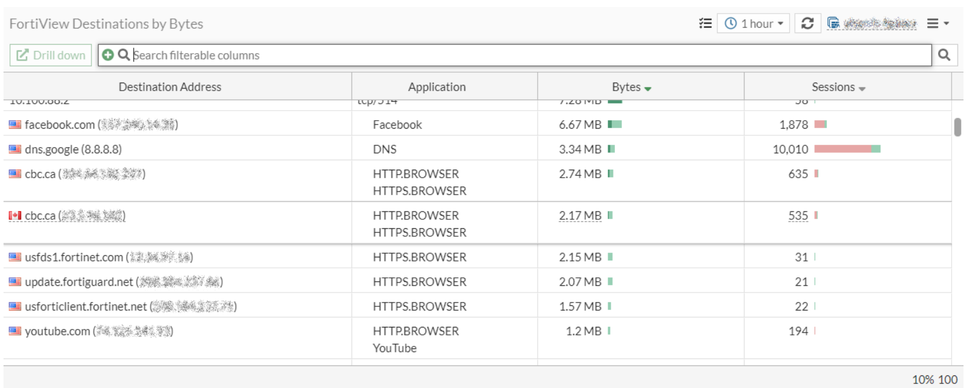
Select a tab, for example *Destination*, and drill down again to apply a second level filter (listed in the summary). That tab will be removed for the available tabs. Click the *X* to remove the filter and show that tab again.



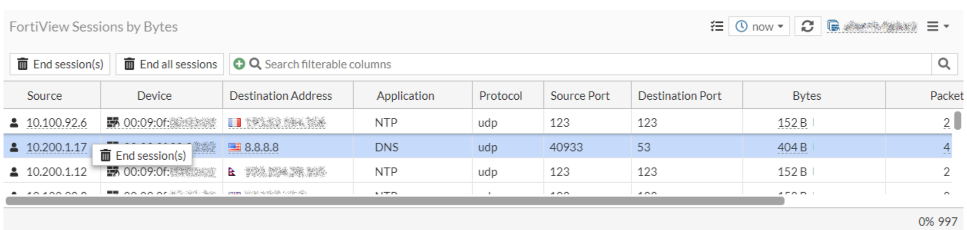
Click *View session logs* to see the log list details.



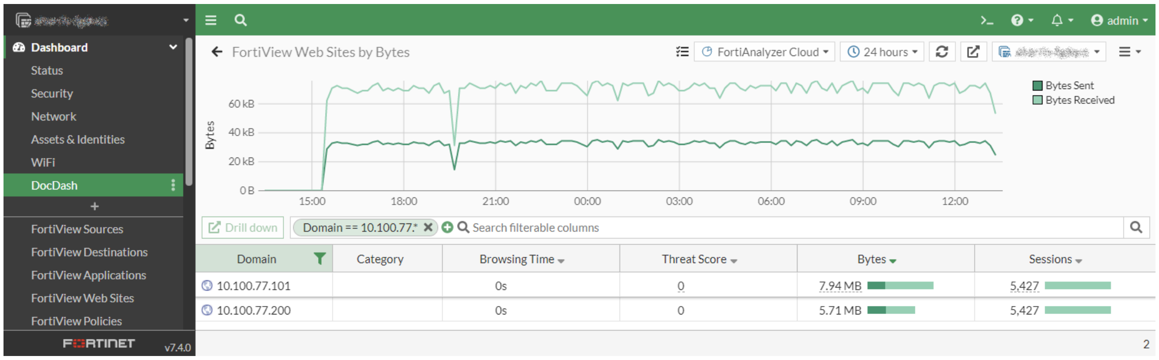
On a FortiView page with enough entries, the graph in the historical view automatically hides when you scroll down the page.



On the FortiView Sessions page, sessions can be ended by selecting the session or sessions then clicking *End session (s)* in the toolbar or right-click menu. Click *End all sessions* in the toolbar to end all of the sessions.

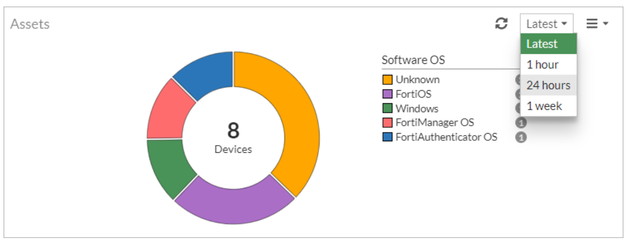


FortiView widgets can be added to custom dashboards. Filters that are applied to the expanded widgets will remain after refreshing the browser.

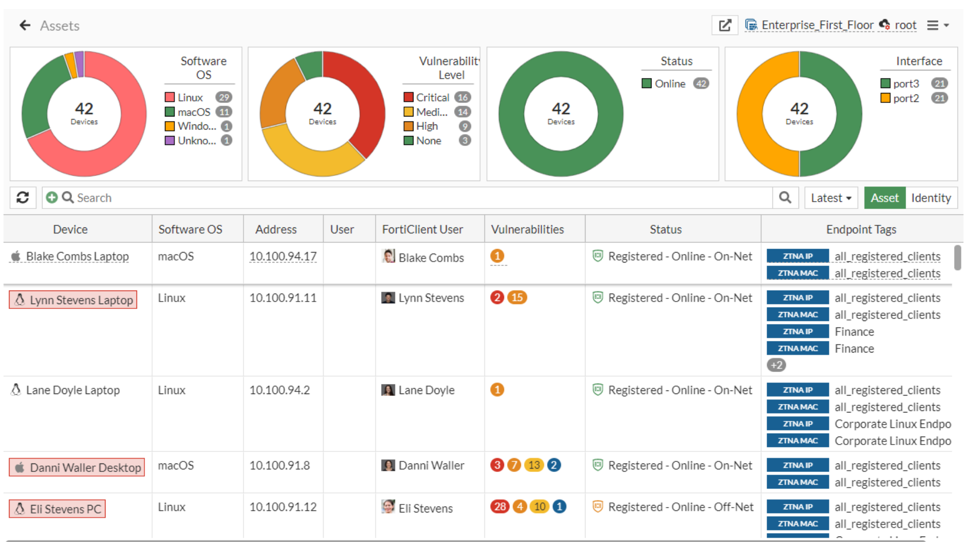


Assets & Identities dashboard

A time range can be specified in the *Assets* widget.



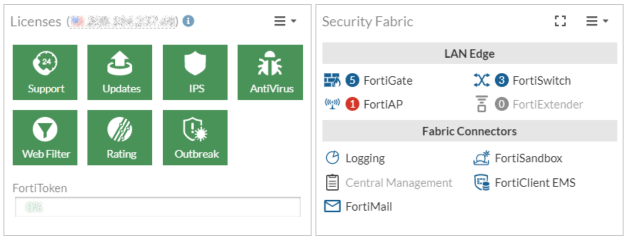
The expanded *Assets* widget is updated.



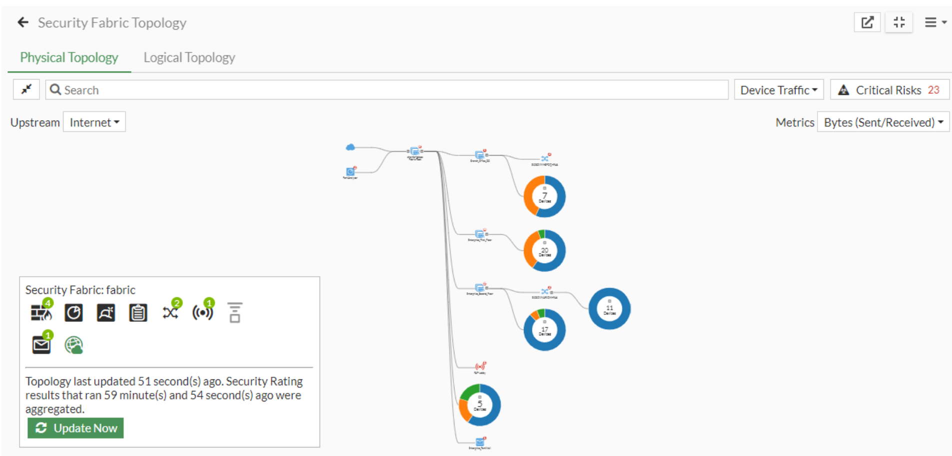
The expanded *Identities* widget is updated.

Status dashboard

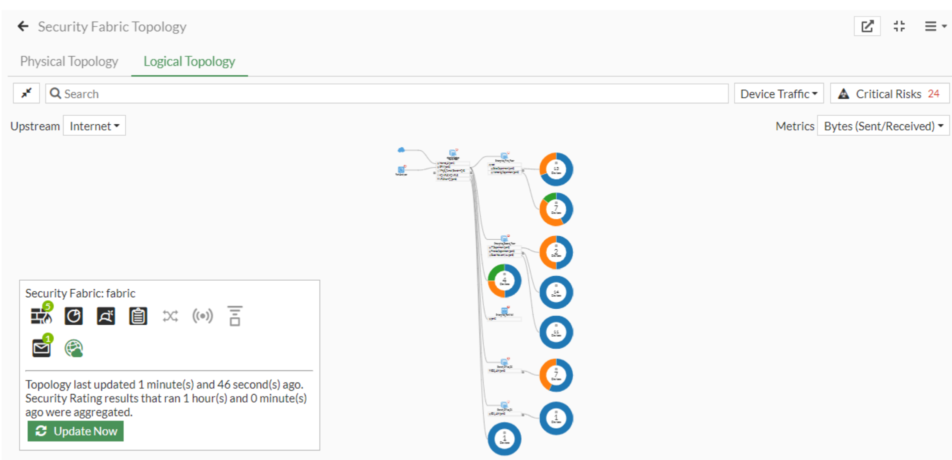
The *Licenses* and *Security Fabric* widgets are updated.



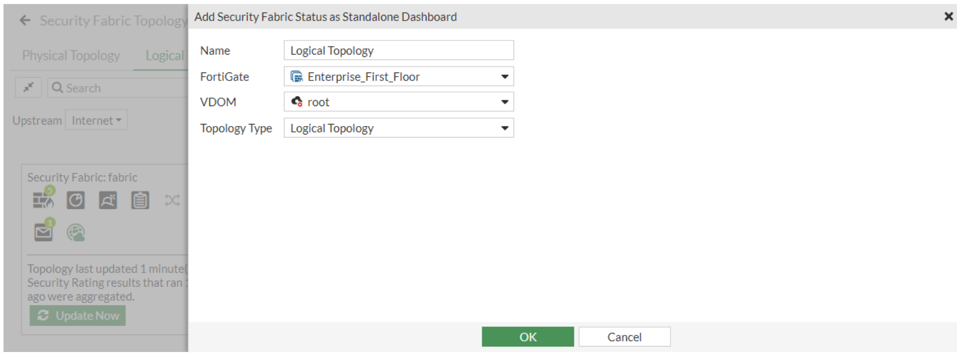
Expand the *Security Fabric* widget to see the *Physical Topology*.



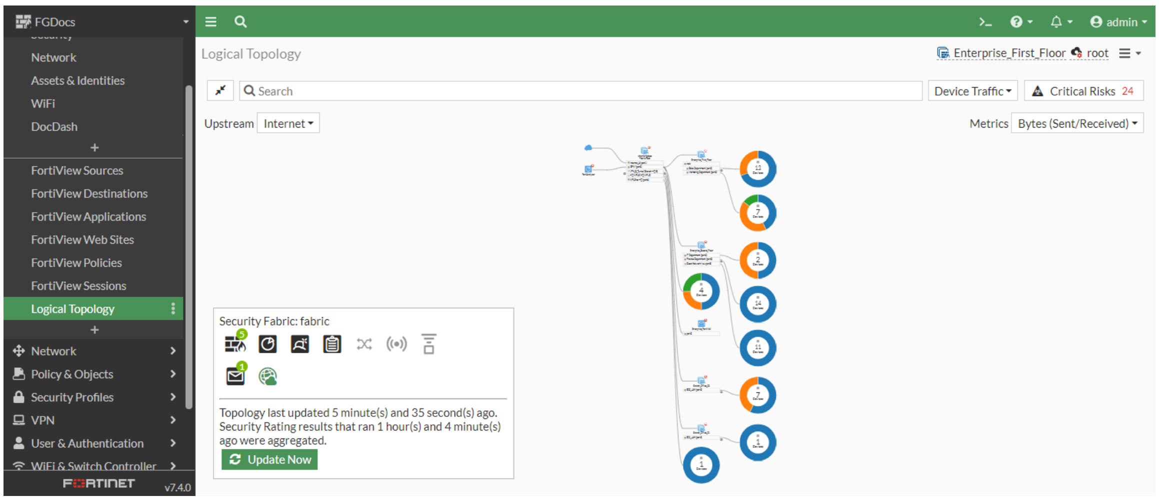
Click the *Logical Topology* tab to see the logical topology.



Click the *Save as Monitor* button, , to save the topology as a dashboard monitor.

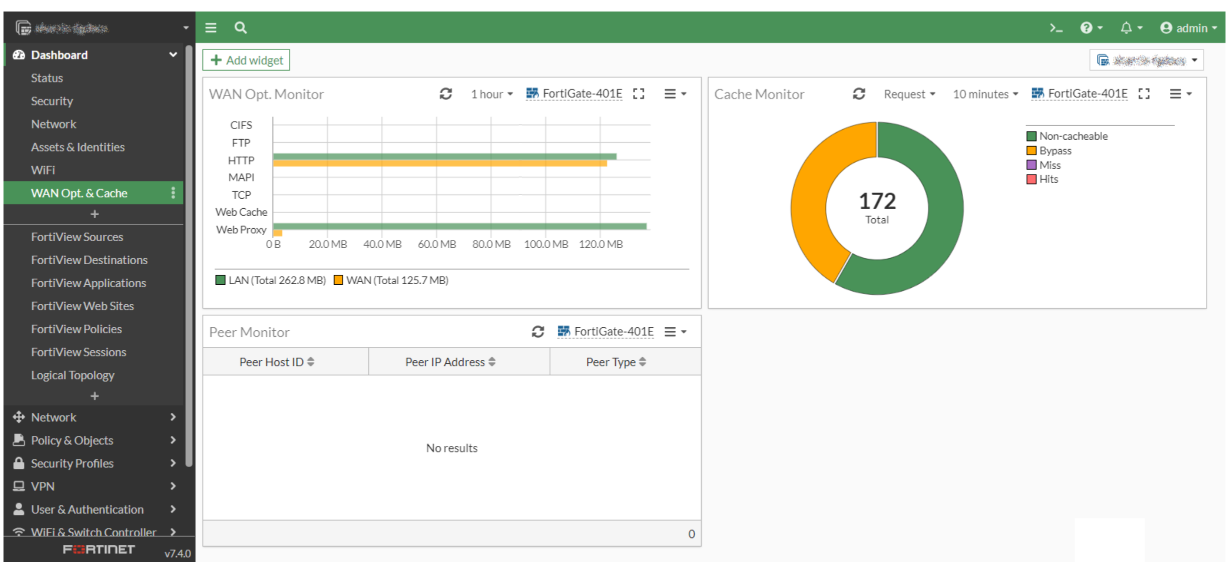


Confirm that the monitor is created and contains the expected data.



WAN Opt. & Cache dashboard

The WAN Opt. & Cache dashboard is updated.



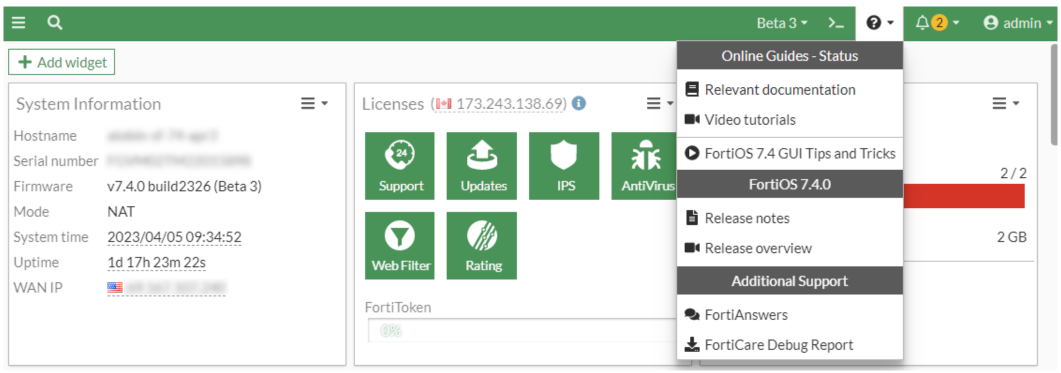
Accessing additional support resources

This information is also available in the FortiOS 7.4 Administration Guide:
- Accessing additional support resources

More integration of additional support resources has been added to the GUI to troubleshoot issues and get the most out of FortiOS. Online guides, FortiOS documentation, and additional support can now be accessed straight from the help menu.

To access support resources:

1. Click *Help* in the top menu. A dropdown menu is displayed.



2. Select the support resource you are looking for:
   - *Online Guides* lists resources for help documentation and videos.
   - *FortiOS <version>* contains release information.
   - *Additional Support* contains a link to access to download the *FortiCare Debug Report*.

Run simultaneous packet captures and use the command palette

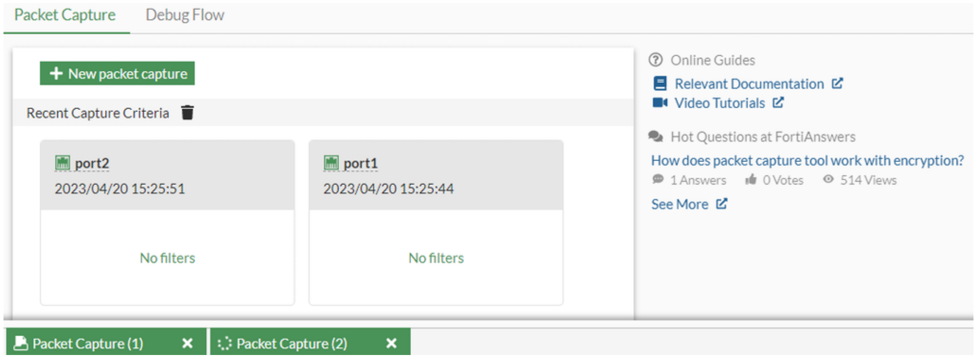
This information is also available in the FortiOS 7.4 Administration Guide:
- Multiple packet captures
- Command palette

The *Network > Diagnostics* page now supports launching multiple packet captures at a time. In addition, a new command palette feature is available for quickly changing between pages and actions using keyboard shortcuts.

Multiple packet captures

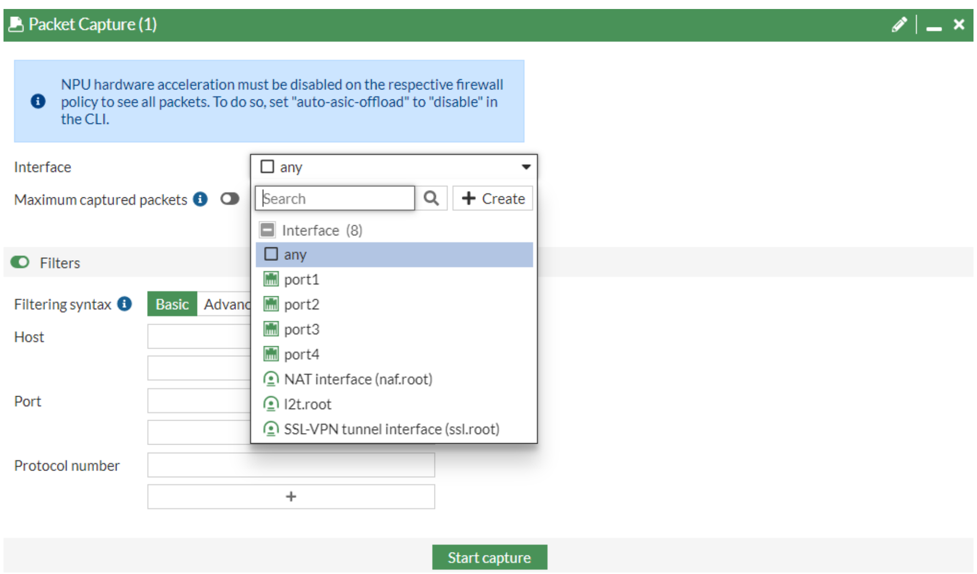
Multiple packet captures can be run simultaneously for when many packet captures are needed for one situation. For example, ingress and egress interfaces can be captured at the same time to compare traffic or the physical interface and VPN interface can be captured using different filters to see if packets are leaving the VPN.

The packet capture dialog can be docked and minimized to run in the background. The minimized dialog aligns with other CLI terminals that are minimized.



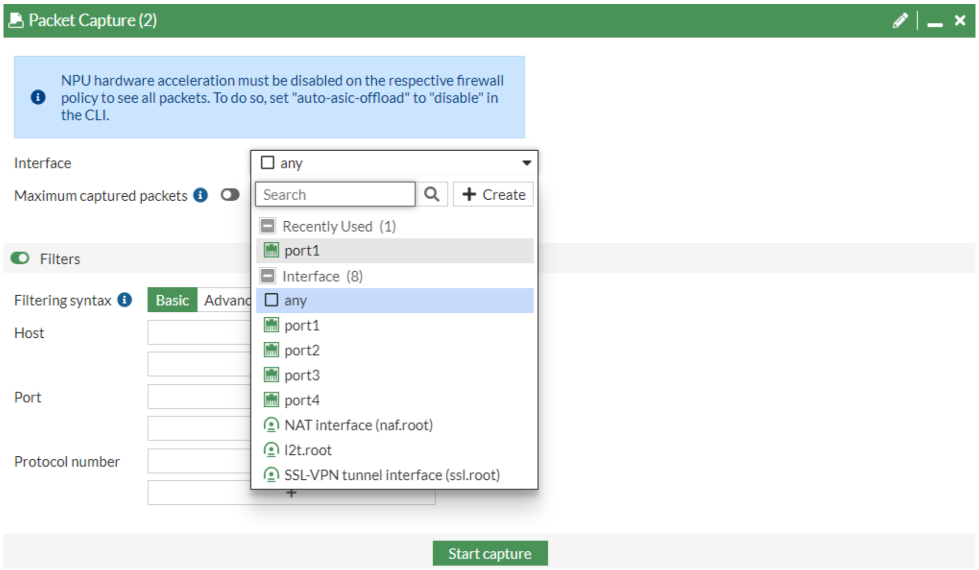
To run multiple packet captures at the same time:

1. Go to *Network > Diagnostics*.
2. Configure the first packet capture:
   a. Click *New packet capture*. The *Packet Capture (1)* dialog is displayed.



   b. Select the *Interface* and configure other settings as needed.
   c. Click *Start capture*. The first packet capture begins.
3. Minimize the packet capture. The packet capture continues to run.
4. Configure the second packet capture:

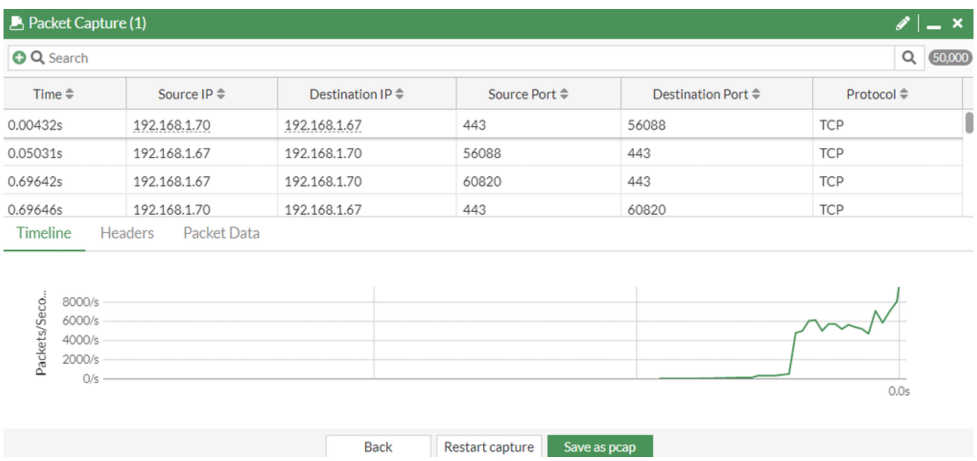
a. Click *New packet capture*. The *Packet Capture (2)* dialog is displayed.



b. Select the *Interface* and configure other settings as needed.

c. Click *Start capture*. The second packet capture begins.

5. When the captures are complete, expand the dialog and select *Save as pcap* for each packet capture.

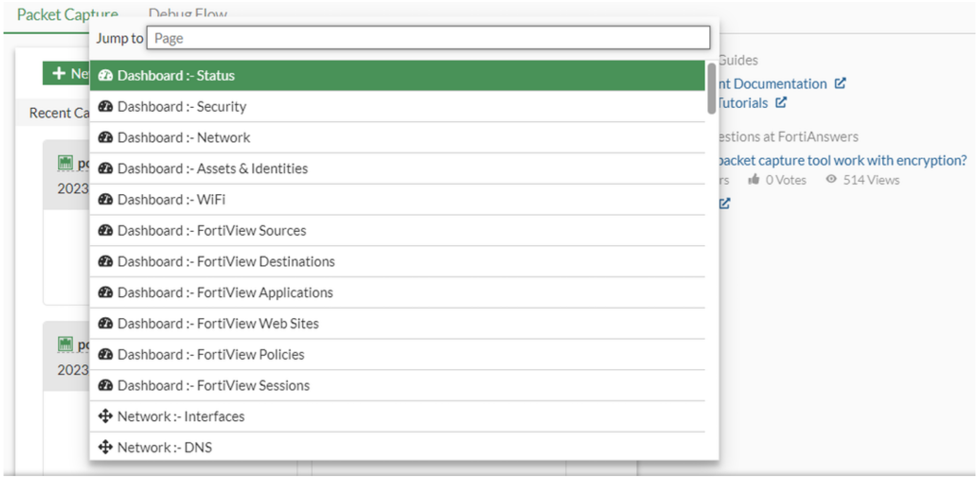


Command palette

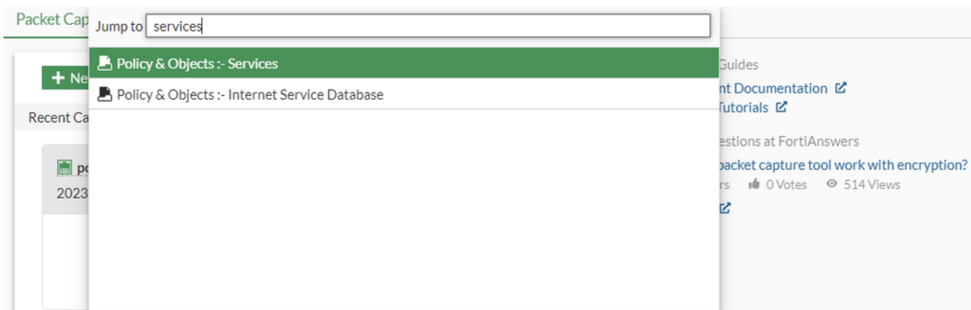
The command palette is a keyboard shortcut menu that can be used to quickly navigate to GUI pages or run specific actions, such as opening the CLI console or restoring a system configuration.

To navigate to a new GUI page using the command palette:

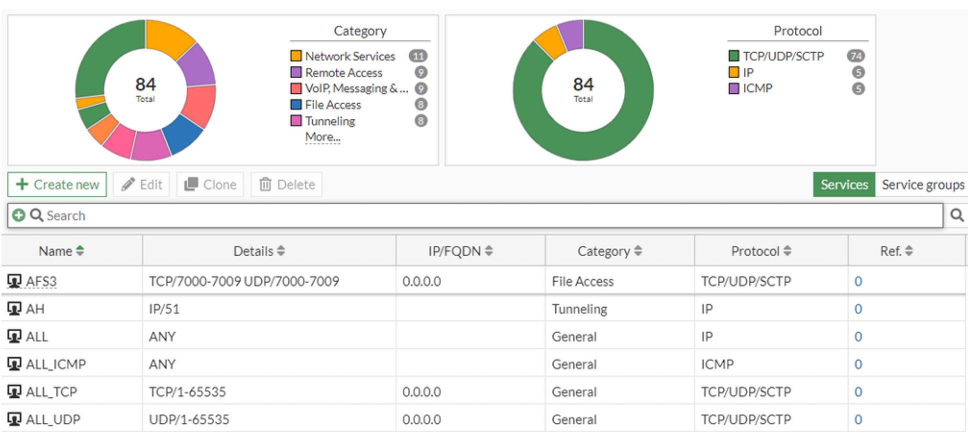
1. Press `ctrl+p` (or `cmd+p` for Mac). The command palette is displayed with available navigation links.



2. Enter the destination.



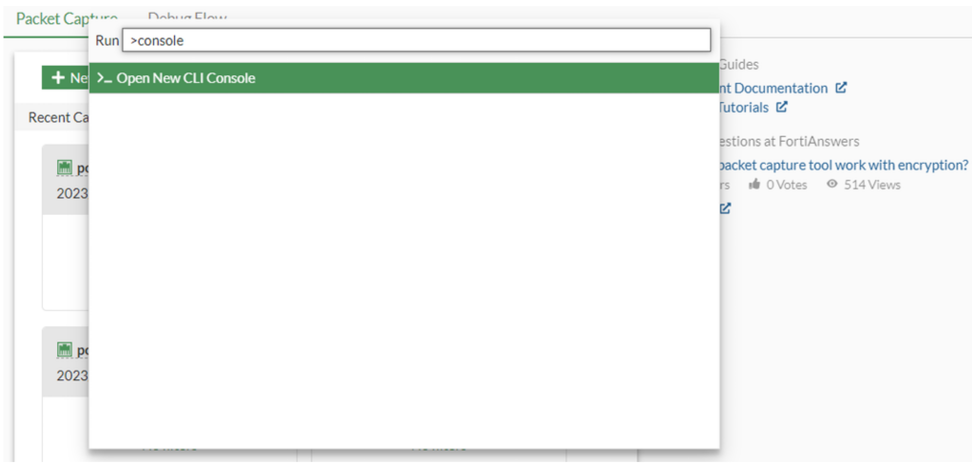
3. Press `Enter` to jump to the page. The set GUI destination page is displayed.



To activate an action using the command palette:

1. Press `ctrl+shift+p` (or `cmd+shift+p` for Mac). The command palette is displayed with a runnable command list.

2. Enter the command key word.



3. Press `Enter` to run the action.

Update FortiSandbox Files FortiView monitor

This information is also available in the FortiOS 7.4 Administration Guide:
- FortiSandbox Files FortiView monitor

The following enhancements have been made to the *FortiSandbox Files* (formerly *Top FortiSandbox Files*) FortiView monitor:

- Add a pie chart with different file statuses for disk data sources.
- Add the *Reports* view, which lists PDF reports after they are downloaded successfully.

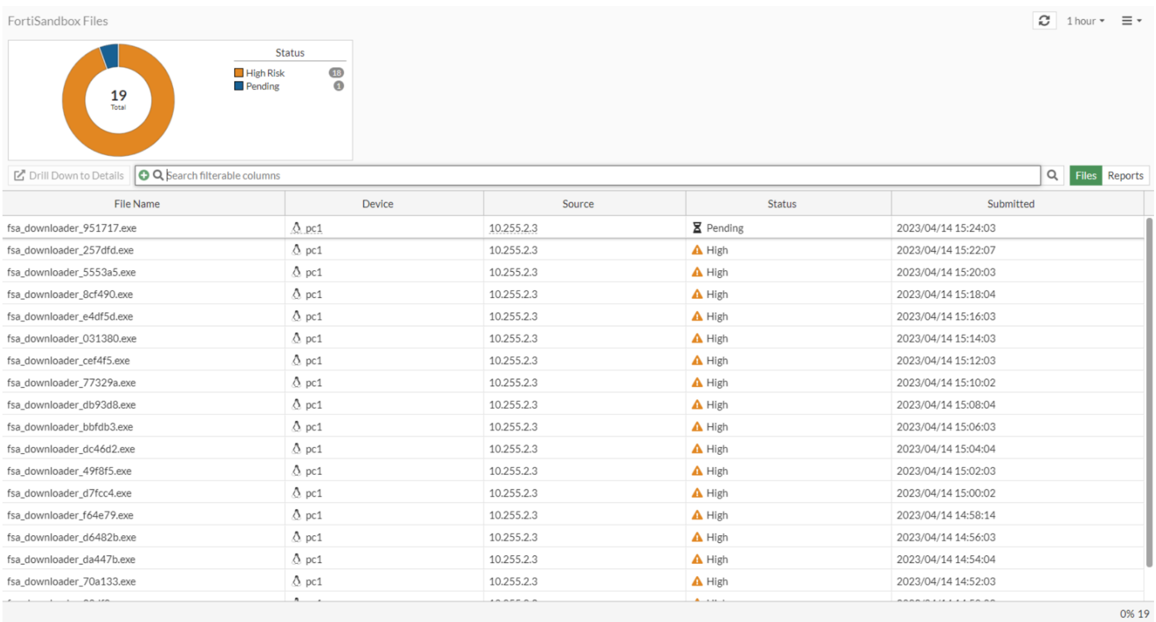
- PDF reports are downloaded on-demand. By default, only 10 are kept in memory.
- PDFs are deleted from memory after 24 hours.

Prerequisites:

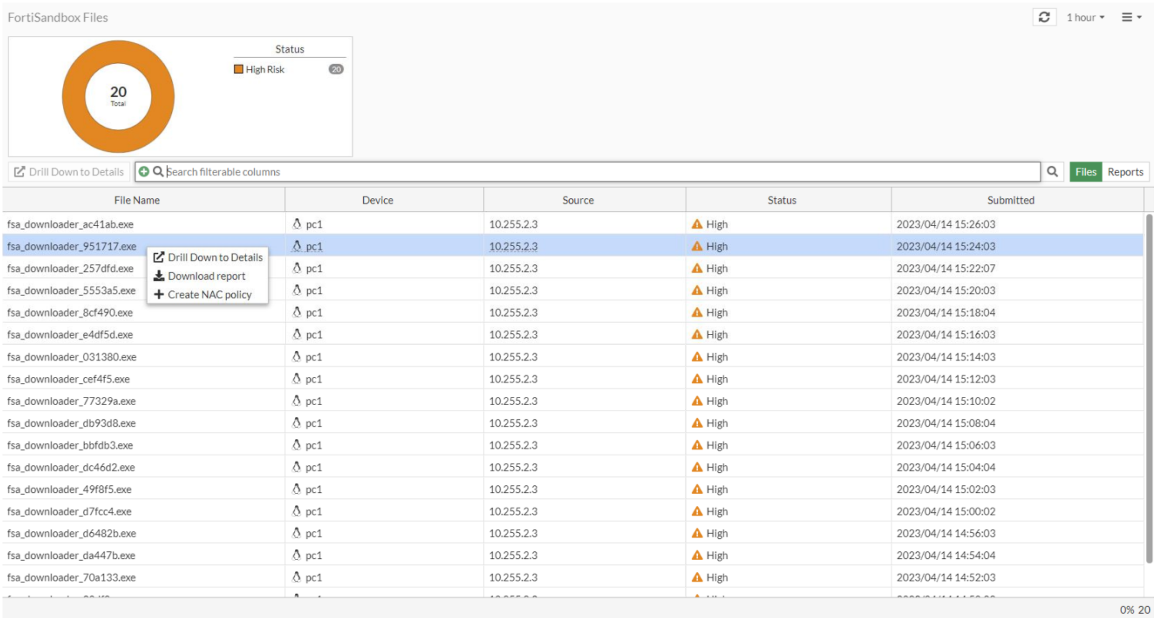
1. Add FortiSandbox running version 3.2.1 or later to the Security Fabric (see Configuring sandboxing in the FortiOS Administration Guide). This feature works with FortiGate Cloud Sandbox, FortiSandbox Cloud, and FortiSandbox appliance.
2. Configure an AV profile with *Send files to FortiSandbox for inspection* enabled (see Using FortiSandbox post-transfer scanning with antivirus in the FortiOS Administration Guide).
3. Configure a firewall policy with the AV profile that allows traffic to the internet.
4. Add the *FortiSandbox Files* FortiView monitor (see Adding FortiView monitors in the FortiOS Administration Guide).
5. On a client PC, attempt to download a suspicious file.

To view the FortiSandbox analysis and download the PDF:

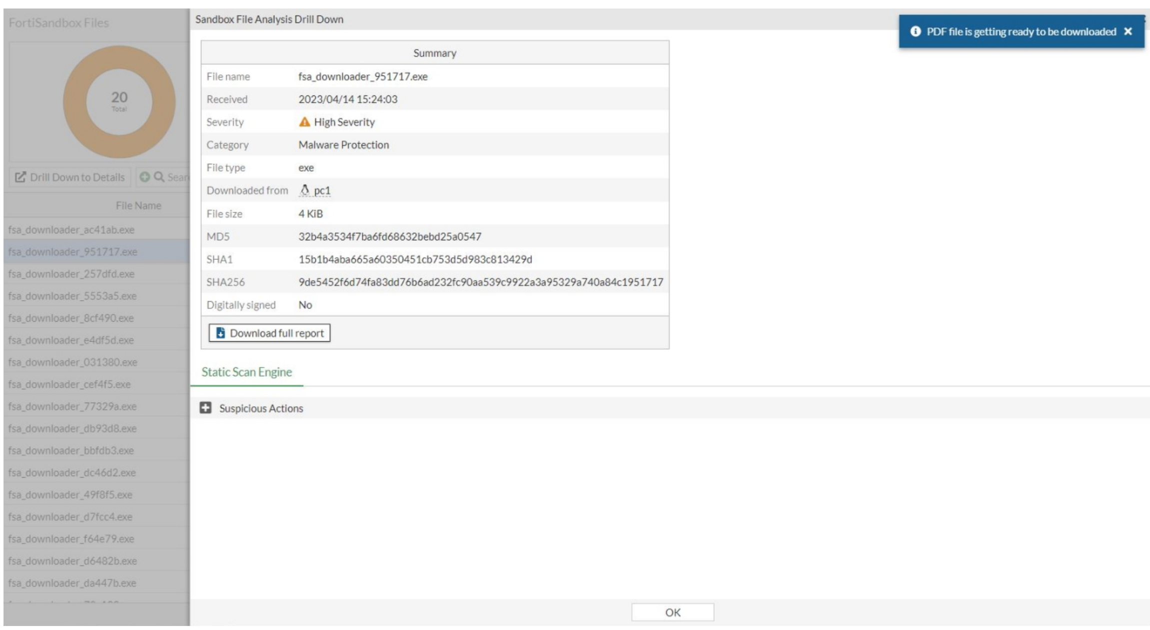
1. Go to *Dashboard > FortiSandbox Files*. The entry appears in the table, but the analysis is not available yet because the *Status* is *Pending*. The default view is *Files*.



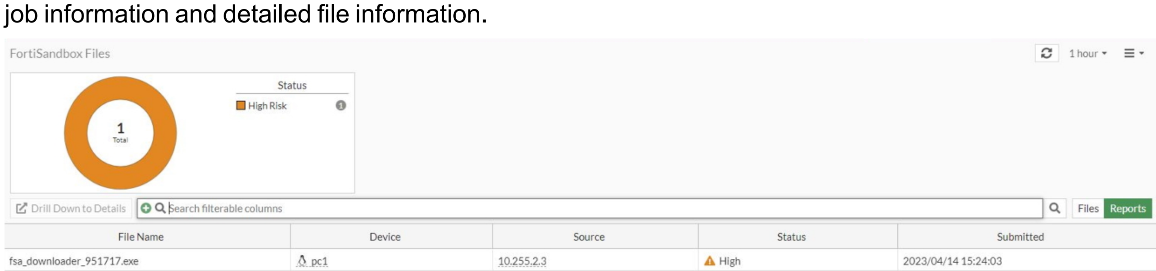
2. After about five to ten minutes, refresh the table. The analysis is now available.
3. Select the entry, then right-click and select *Drill Down to Details*.



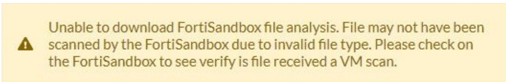
The *Sandbox File Analysis Drill Down* pane opens.



4. Click *Download full report* to download the detailed PDF report.
5. Change the view to *Reports* to verify that the file was downloaded successfully. The reports contains FortiSandbox job information and detailed file information.



When the file type is not supported, a warning message appears that the file was not scanned when the *Sandbox File Analysis Drill Down* pane opens.



To change the maximum number of PDFs kept in memory:

```
# diagnose test analytics-pdf-report max <integer>
```

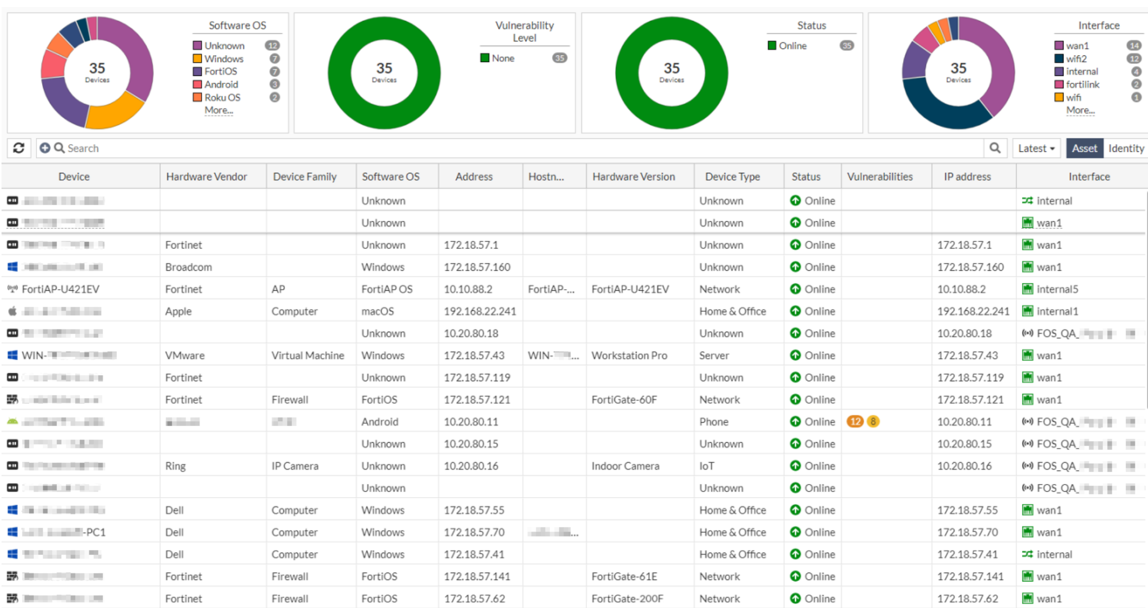
The range is 1 - 10, and the default is 10. After the FortiGate is restarted, this value will revert to the default.

Combine the Device Inventory widget and Asset Identity Center page

The *Device Inventory* widget and *Asset Identity Center* page have been combined to create a more streamlined appearance and to conserve resources. The *Security Fabric > Asset Identity Center* page offers a unified view of asset information, consolidates data from various sources, and can handle significantly larger sets of data.

Note the following updates:

- There are four donut charts available in the *Asset* view: *Software OS*, *Vulnerability Level*, *Status*, and *Interface*. These charts used to be included in the *Device Inventory* widget, which is now replaced with the *Asset* widget.
- Device information is available in the *Hardware Vendor*, *Device Family*, *Software OS*, and *Hardware Version* columns.
- The *Interface* column is included in the table.
- The *IoT Vulnerabilities* and *Endpoint Vulnerabilities* columns from previous versions have been merged into the *Vulnerabilities* column.



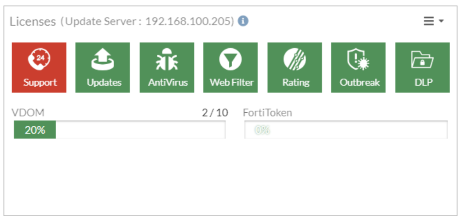
GUI enhancements for FortiGuard DLP service - 7.4.1

The FortiOS GUI has been enhanced to support the FortiGuard DLP service, which includes the following changes:

- Add a new item in the *Licenses* widget (*Dashboard* > *Status*) and *License Information* list (*System* > *FortiGuard*) for the new DLP service.
- Use a FortiGuard icon for DLP patterns that are dynamically retrieved from FDS.
- Clearly distinguish DLP dictionaries and sensors by grouping them as *Managed Locally* and *Managed by FortiGuard*.
- Show an inline message in the tooltip header for the data types, dictionaries, and sensors managed by FortiGuard.

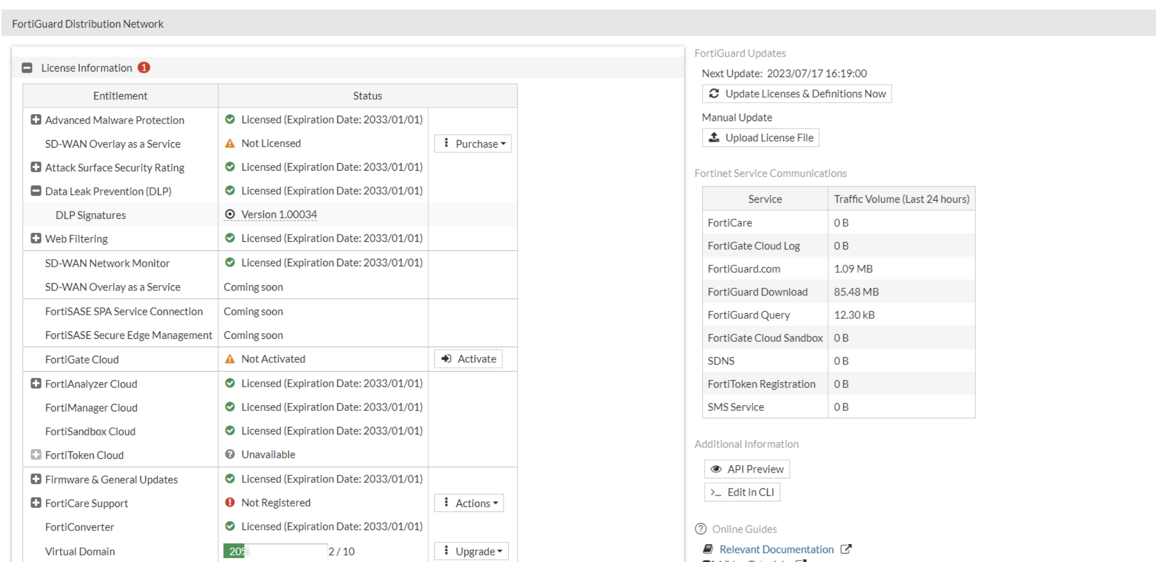
DLP service icon

Once the DLP database is downloaded by a scheduled or manual update, an icon is visible in the *Dashboard* > *Status* > *Licenses* widget.



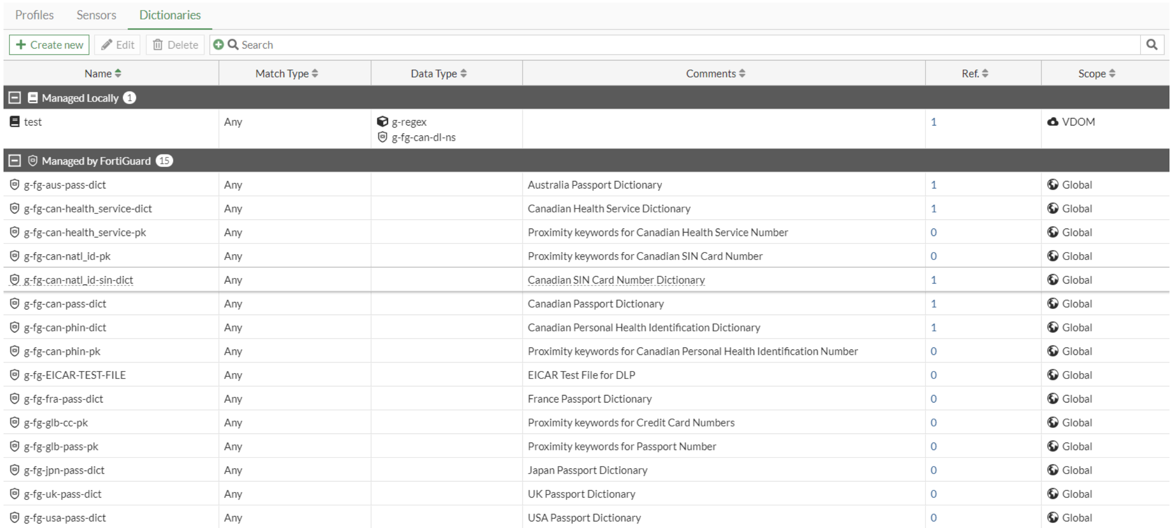
FortiGuard license information

To view the entitlement information, go to *System* > *FortiGuard*. In the *License Information* list, expand *Data Leak Prevention (DLP)* to view the *DLP Signatures* version details.



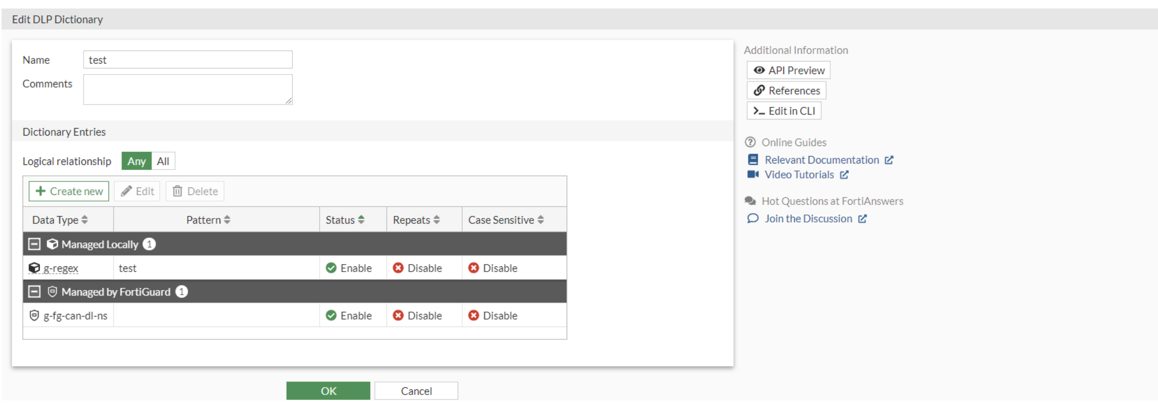
Dictionary classification

In the *Dictionaries* tab of the *Security Profiles* > *Data Leak Prevention* page, all dictionaries are grouped as *Managed Locally* and *Managed by FortiGuard*. In the following example, the *test* dictionary is classified as *Managed Locally* and has a book icon beside its name. Several dictionaries, such as *g-fg-aus-pass-dict*, are classified as *Managed by FortiGuard* and have a FortiGuard shield icons beside their names.



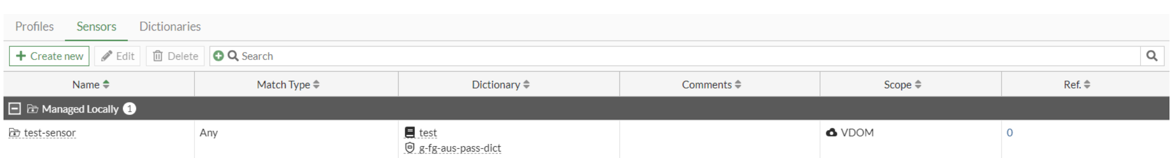
The items listed in the *Data Type* column include icons: a cube icon for local data types, and a FortiGuard shield icon for data types managed by FortiGuard.

When editing a dictionary, the *Dictionary Entries* table entries are grouped as *Managed Locally*and *Managed by FortiGuard*.



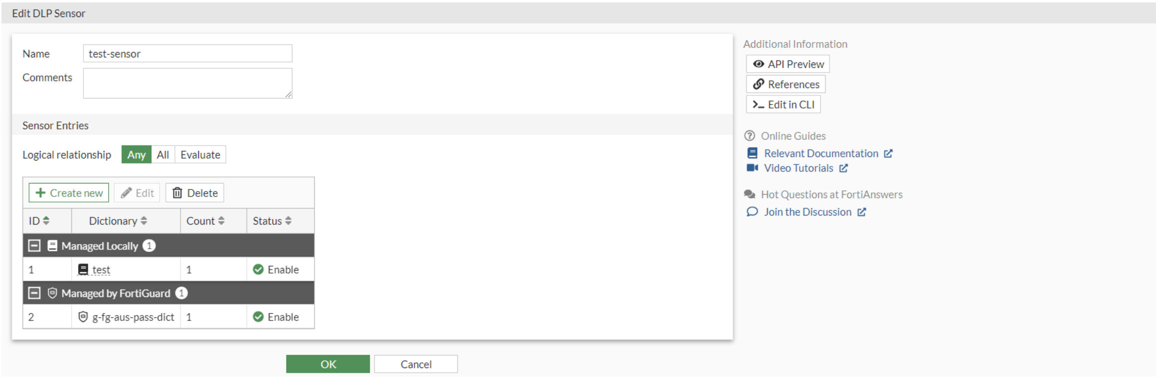
Sensor classification

In the *Sensors* tab of the *Security Profiles > Data Leak Prevention* page all sensors are grouped as *Managed Locally* and *Managed by FortiGuard*. In the following example, there is one local sensor, *test-sensor*. The sensor has a folder icon beside its name since it is a local sensor.



The items listed in the *Dictionary* column include icons: a book icon for a local dictionary, and a FortiGuard shield icon for a dictionary managed by FortiGuard.

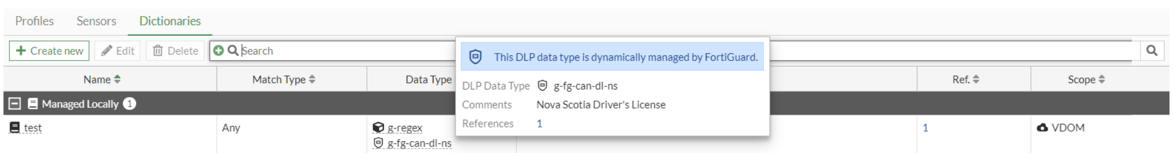
When editing a sensor, the *Sensor Entries* table entries are grouped as *Managed Locally* and *Managed by FortiGuard*.



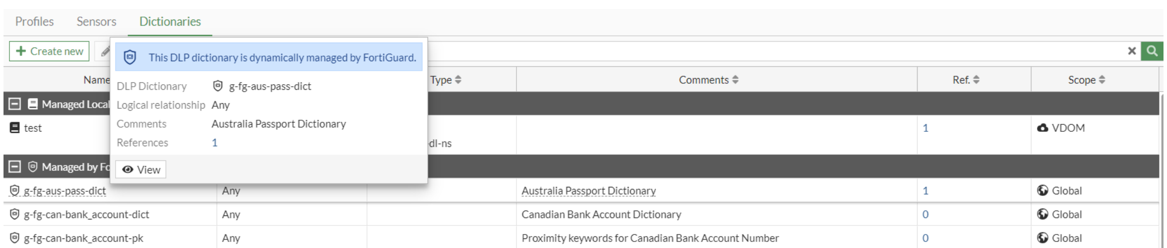
Tooltip header

A header is included in the tooltips for data types, dictionaries, and sensors that are managed by FortiGuard.

Example tooltip for a data type:



Example tooltip for a dictionary:



FortiConverter usability improvements - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

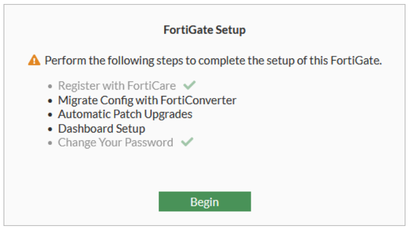
- Migrating a configuration with FortiConverter

Migrating a configuration from one FortiGate to another directly in the GUI without having to access the FortiConverter Service Portal has been updated.

Both FortiGates must be registered under the same FortiCare account, have internet connectivity to reach the FortiConverter server, and the target FortiGate must have a valid FortiConverter license.

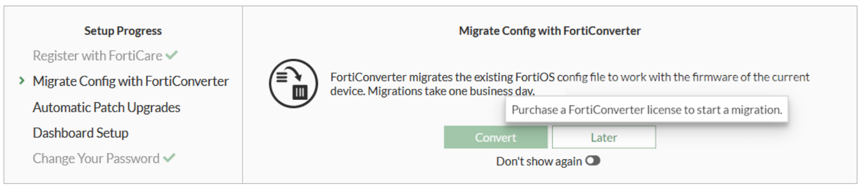
To migrate a configuration with FortiConverter:

1. On the GUI startup menu, after registering with FortiCare, click *Begin* to *Migrate Config with FortiConverter*.



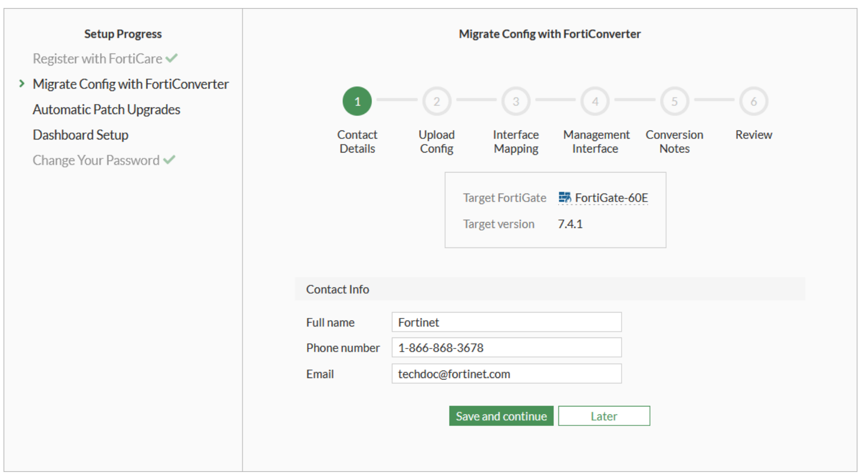
2. Click *Convert* to start the conversion process.

   If the device does not have a FortiConverter license, a warning will be shown and the *Convert* button will be unclickable. The license status is shown in the GUI on the *System > FortiGuard* page in the *License Information* table.



   You can toggle the *Don't show again* option and click *Later* to turn off reminders about the migration process.

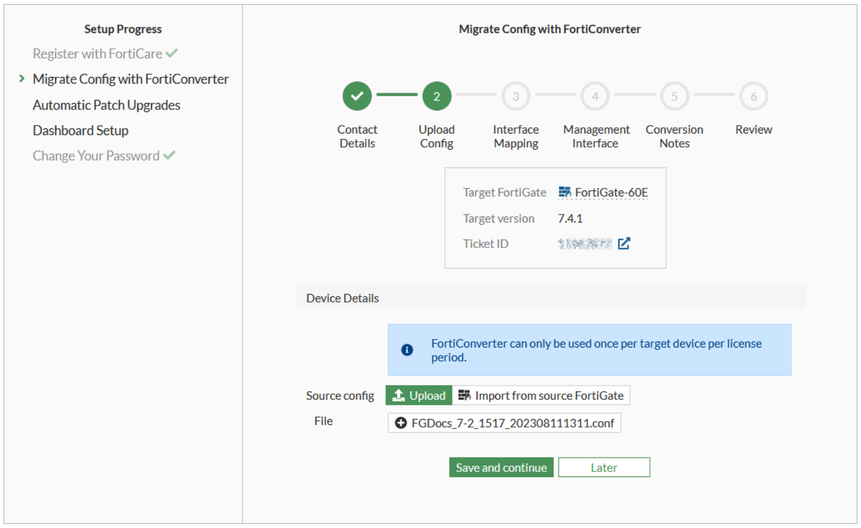
3. Enter the user contact information, then click *Save and continue*.



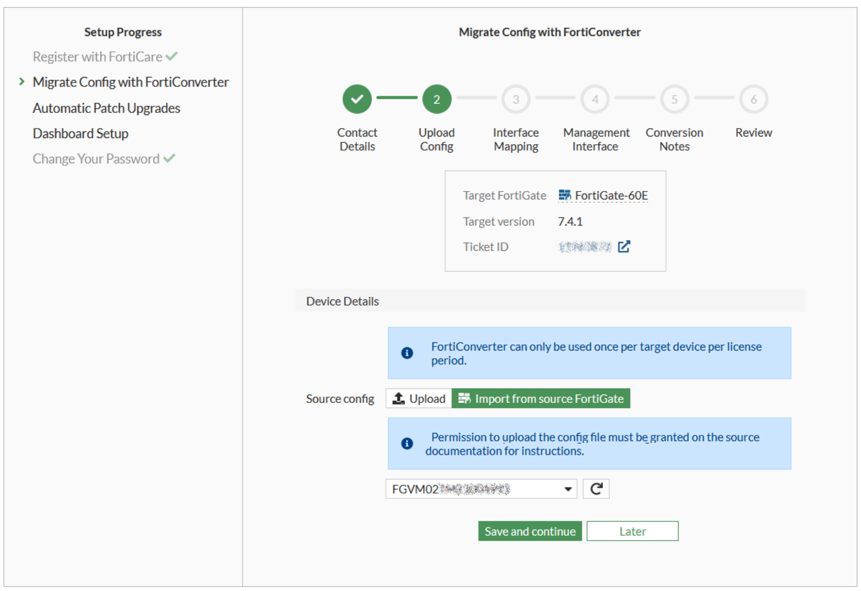
   The FortiConverter ticket is created.

4. The source configuration can be uploaded from a file, or from another FortiGate.

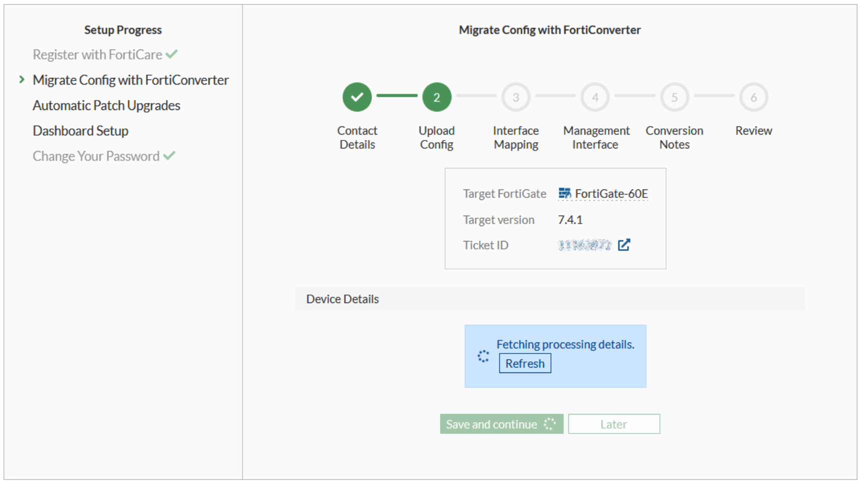
- To upload from a file, set *Source config* to *Upload* then click *Browse* to locate the file.



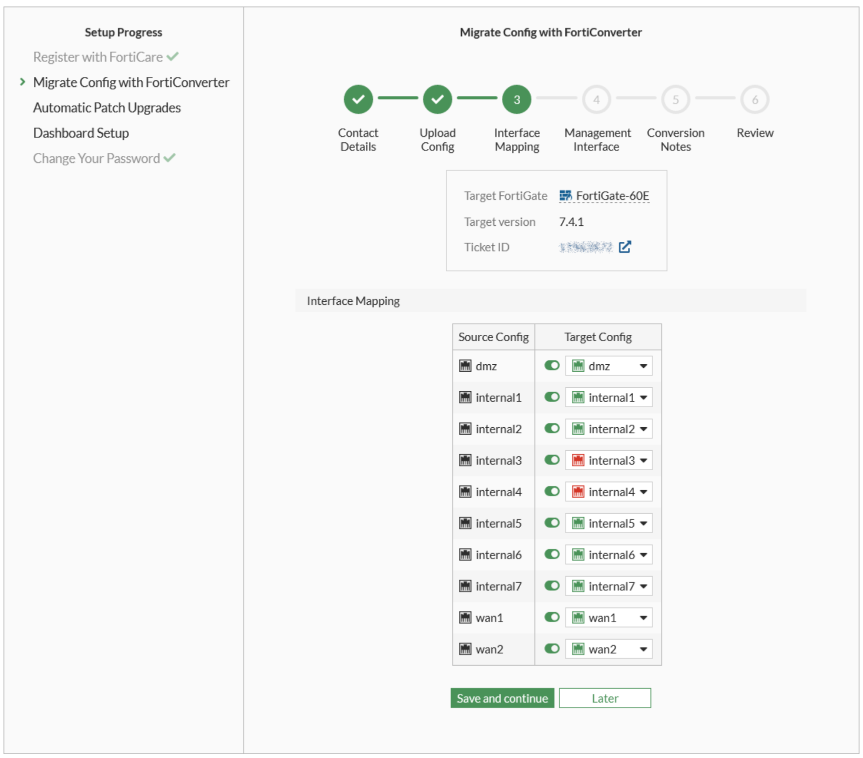
- To import from another FortiGate, set *Source config* to *Import from source FortiGate* then select the source FortiGate. *Allow FortiConverter to obtain config file once* must be enabled in *System* > *Settings* on the source FortiGate.



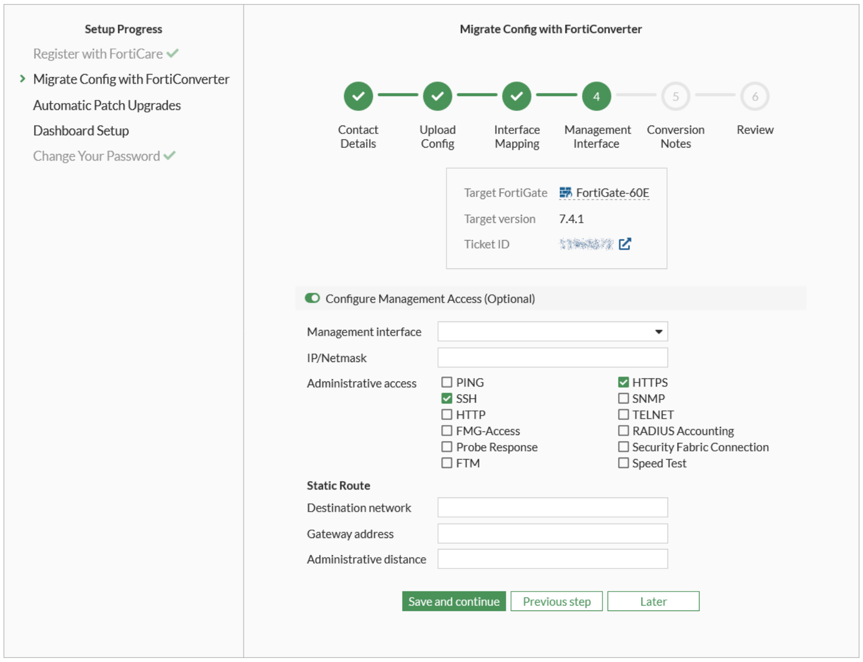
5. Click *Save and continue*, then wait for the configuration file to be uploaded to FortiConverter and processed.



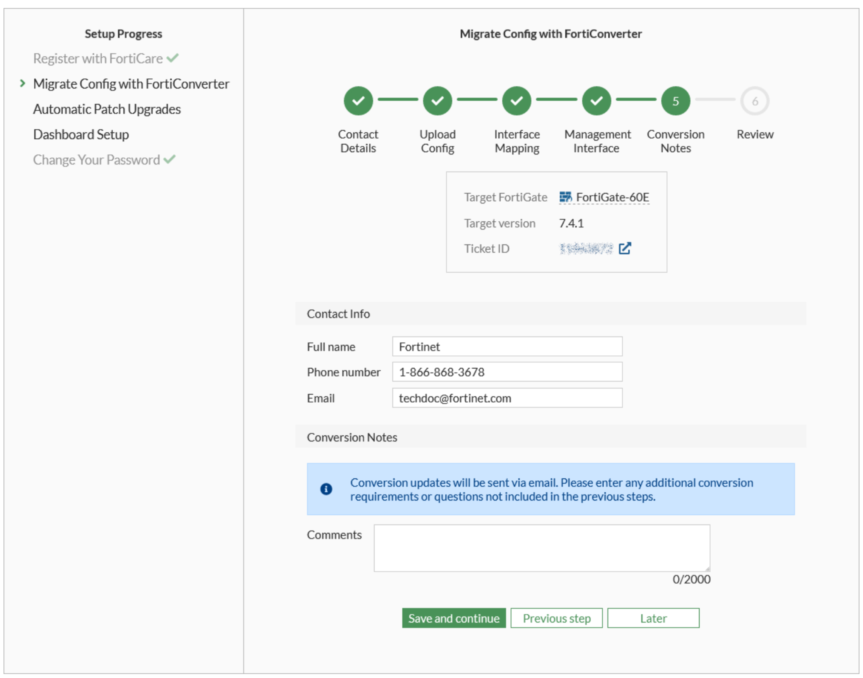
6. Define the interface mapping between the source and target configuration, then click *Save and continue*. The target interfaces are prepopulated.



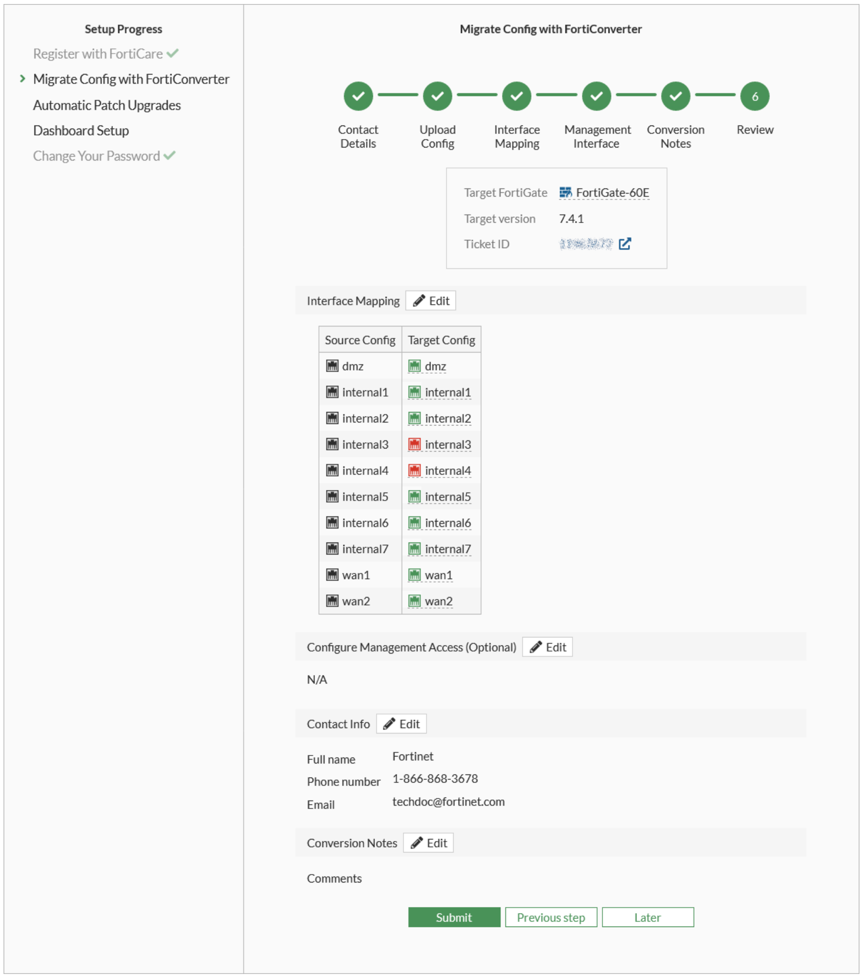
7. Optionally, configure management access on the FortiGate, then click *Save and continue*. The administrative distance can now be configured.



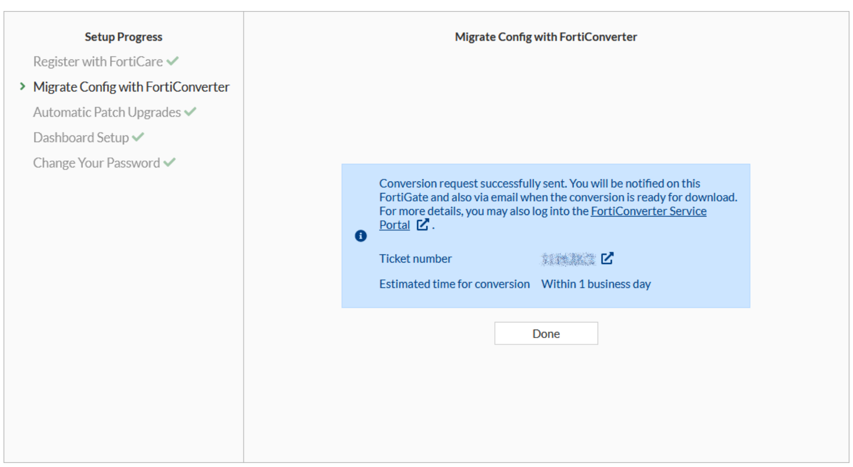
8. Enter conversion notes in the *Comments* field, then click *Save and continue*.



9. Review the content, then click *Submit*.



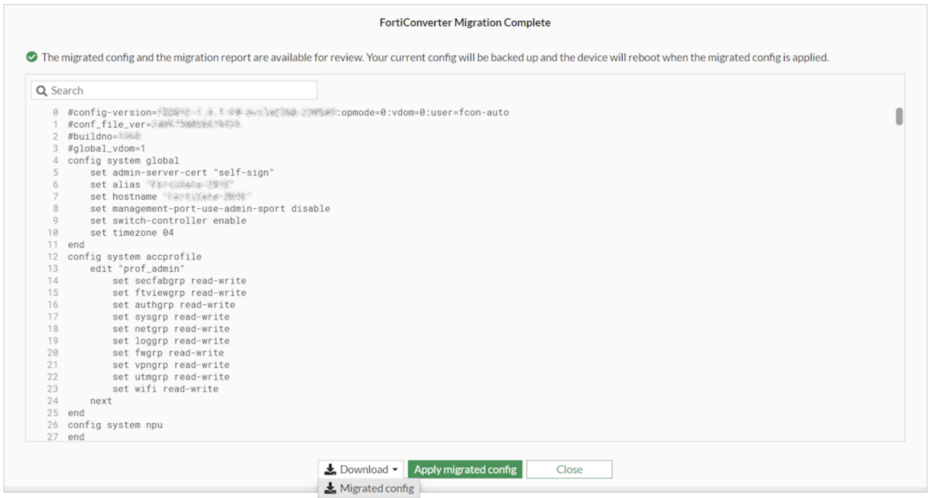
The conversion request is sent, an email is sent to confirm that the conversion process has started in FortiConverter, and the ticket status is shown. The estimated conversion time is one business day.



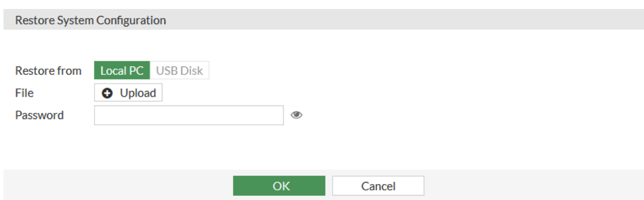
10. Click *Done*.

    When the conversion process completes, you will receive an email and a notifications in the FortiGate GUI.

11. In the GUI, click your administrator name and select *Configurations> FortiConverter*. The migrated configuration is shown for review, and can be downloaded.



12. Click *Apply migrated config* to apply the converted configuration to the FortiGate. This will cause the FortiGate to reboot. The existing configuration will be backed up before the converted configuration is applied.
13. To manually load to configuration file:
    a. Click your administrator name and select *Configuration* > *Restore*.



    b. Upload the converted configuration file, then click *OK*. This will cause the FortiGate to reboot.

To see the visibility status of the FortiConverter wizard:

```
diagnose sys forticonverter get-prompt-visibility
```

To set the visibility status of the FortiConverter wizard:

```
diagnose sys forticonverter set-prompt-visibility {visible | hidden}
```

Update FortiGuard License Information widget - 7.4.1

The *System* > *FortiGuard* > *License Information* widget has been updated to align with current FortiGuard services and entitlements, as well as corresponding definitions, signatures, engines, and databases associated with each service entitlement.

The *Licenses* widget on the *Dashboard* > *Status* page has also been updated to display the major services that are licensed.

License Information widget

The service entitlements and the license statuses are listed on the *System* > *FortiGuard* page. Upon expanding each entitlement, the corresponding definitions associated with the service are listed.

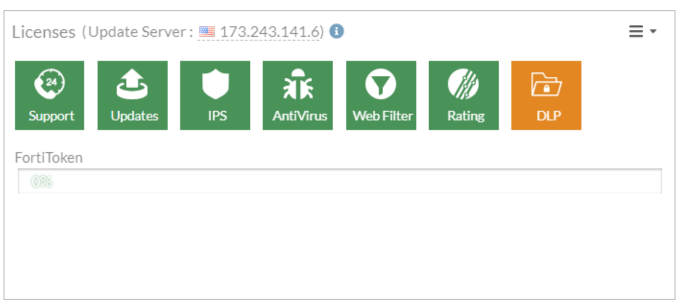
The following table list the available FortiGuard services and entitlements with a brief description.

Entitlement	FortiGuard service description
Advanced Malware Protection AI Malware Detection Model AntiVirus Definitions AntiVirus Engine Mobile Malware Outbreak Prevention	The Advanced Malware Protection service includes various engines, databases, and definitions used in the AV profile. See the AntiVirus section in the FortiOS Administration Guide for details.
Attack Surface Security Rating IoT Detection Definitions Outbreak Package Definitions Security Rating & CIS Compliance	The Attack Surface Security service includes: • Running all the built-in free and paid security rating rules • Displaying CIS compliance information within security ratings • IoT Detection and IoT Query
Data Leak Prevention (DLP) DLP Signatures	The Data Loss Prevention service offers a database of predefined DLP patterns such as data types, dictionaries, and sensors that are used in the DLP profile.
Email Filtering	Email Filtering includes spam and DNS filtering by FortiGuard.
Intrusion Prevention IPS Definitions IPS Engine Malicious URLs Botnet IPs Botnet Domains	The IPS service includes engines, databases, and definitions used in the IPS and application control profiles. See the Intrusion prevention and Application control sections in the FortiOS Administration Guide for details.
Operational Technology (OT) Security Service OT Threat Definitions OT Detection Definitions OT Virtual Patching Signatures	The OT Security service includes OT-related threat definitions used in IPS and application control profiles. It also includes OT Detection Definitions and Virtual Patching Signatures used in the virtual patching profile.
Web Filtering Blocked Certificates DNS Filtering Video Filtering	The Web Security service includes: • FortiGuard categories used in web filter profiles • Malicious certificates used in SSL/SSH inspection profiles • FortiGuard categories used in DNS filter profiles • FortiGuard categories used in video filter profiles
SD-WAN Network Monitor	SD-WAN Underlay Bandwidth and Quality Monitoring service
SD-WAN Overlay as a Service	SD-WAN Overlay as a Service
FortiSASE SPA Service Connection	SD-WAN Connector for FortiSASE Secure Private Access
FortiSASE Secure Edge Management	Allows the FortiGate to act as the FortiSASE Secure Edge

Entitlement	FortiGuard service description
FortiGate Cloud	FortiGate Cloud management, analysis, and log retention services
FortiAnalyzer Cloud SoCaaS	FortiAnalyzer Cloud service The SoCaaS entitlement includes cloud-based managed log monitoring, incident triage, and SOC escalation services.
FortiManager Cloud	FortiManager Cloud service
FortiToken Cloud	FortiToken Cloud service
Firmware & General Updates Application Control Signatures Device & OS Identification FortiGate Virtual Patch Signatures Inline-CASB Application Definitions Internet Service Database Definitions PSIRT Package Definitions FortiCare Support FortiCloud Account Enhanced Support	The FortiCare support entitlement includes firmware and general updates that come with various default signatures and definitions: • Application control signatures used in application control profiles • Device & OS identification used for device detection and asset management • Virtual patch signatures used in local-in policies • Inline CASB application definitions used in inline CASB profiles • ISDB destinations that can be applied in various policies and rules • PSIRT vulnerability definitions used in security ratings
FortiConverter	FortiConverter service

Licenses widget

On the *Dashboard* > *Status* page, the *Licenses* widget lists the status of major entitlements. Licensed entitlement icons are green, and unlicensed entitlement icons are orange.



Optimize policy and objects pages and dialogs - 7.4.2

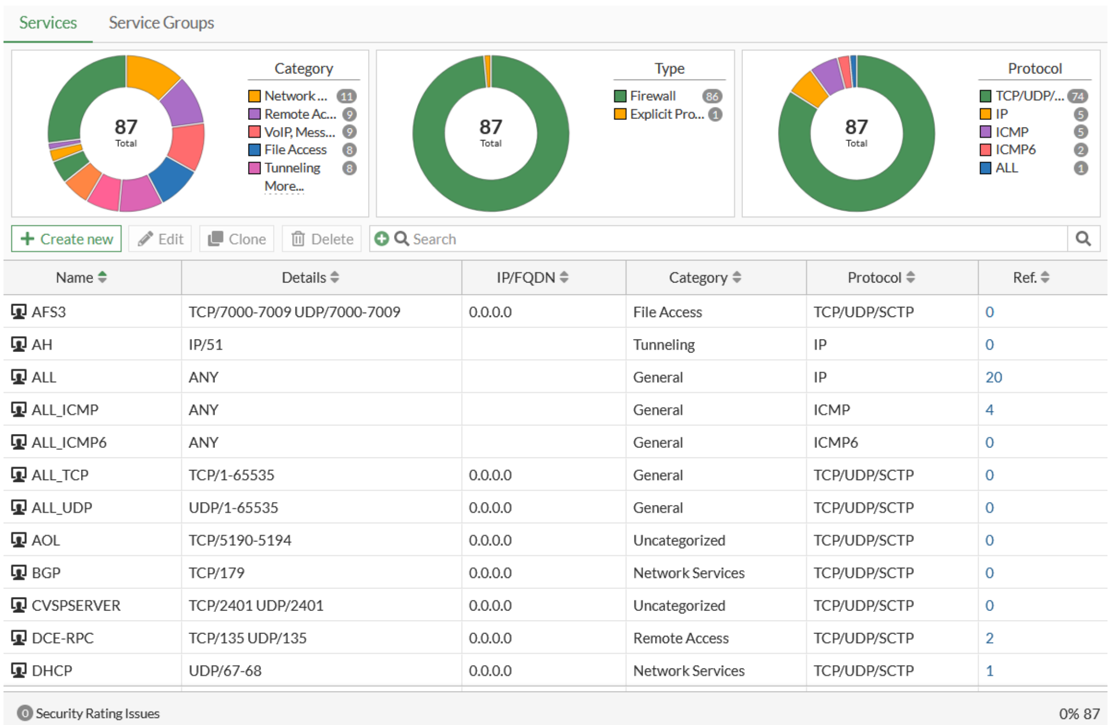
The *Policy & Objects* pages have been optimized for loading large datasets. For example, instead of loading an entire dataset of address objects on the *Addresses* page or within the address object dialog inside a firewall policy, data is lazily-loaded. Different types of address objects are also loaded separately.

Enhancements include:

- Add a tabbed design for firewall object list pages.
- Apply lazy-loading of the firewall address list and introduce sub-tabs for each type of address object.
  - Pages are lazily-loaded based on the dropdown selection.
  - Paged results are returned by the backend in order to return results quickly.
- Update the *Address* dialog page.
- Update the *Policy* dialogs and use new address dialogs with a lazily-load selection widget.

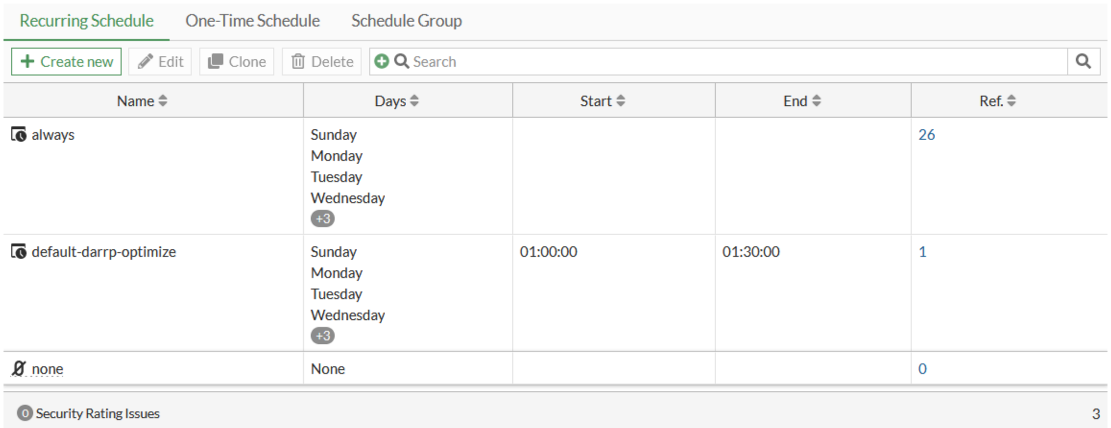
Services page

The *Policy & Objects* > *Services* page includes tabs for *Services* and *Service Groups*.



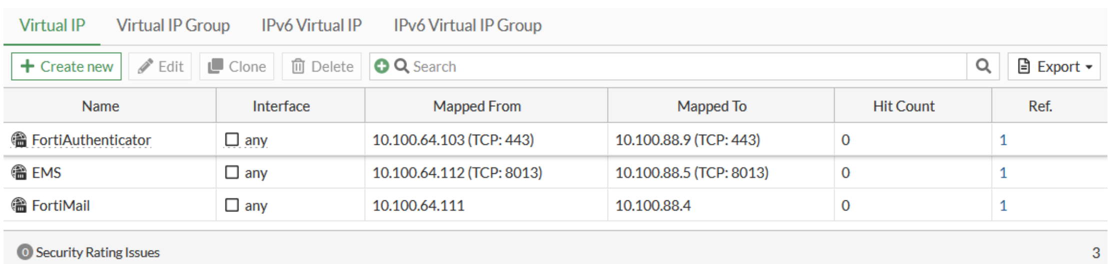
Schedules page

The *Policy & Objects* > *Schedules* page includes tabs for *Recurring Schedule*, *One-Time Schedule*, and *Schedule Group*.



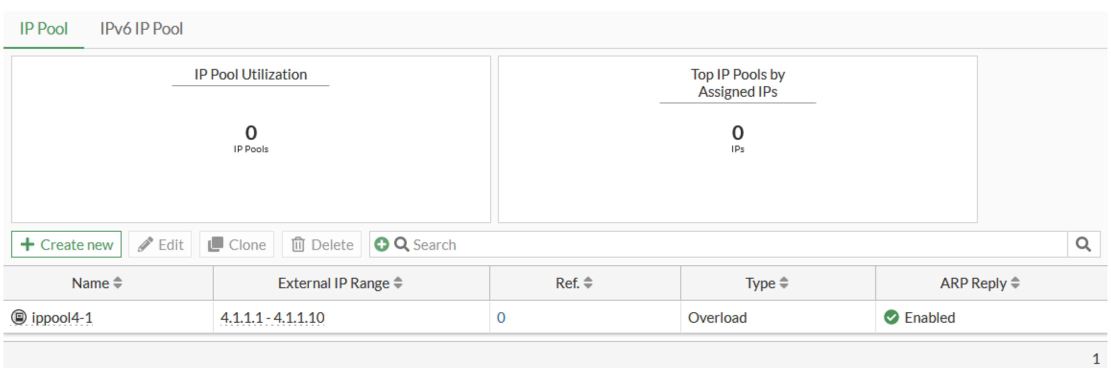
Virtual IPs page

The *Policy & Objects* > *Virtual IPs* page includes tabs for *Virtual IP*, *Virtual IP Group*, *IPv6 Virtual IP*, and *IPv6 Virtual IP Group*.



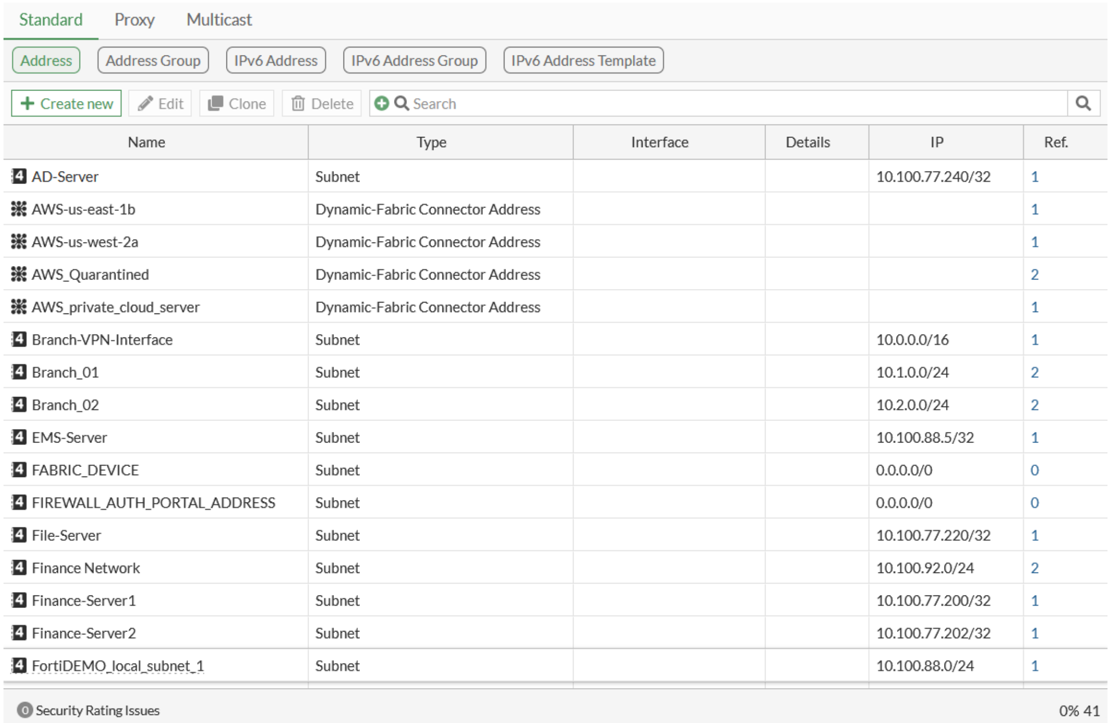
IP Pools page

The *Policy & Objects* > *IP Pools* page includes tabs for *IP Pool* and *IPv6 IP Pool*.

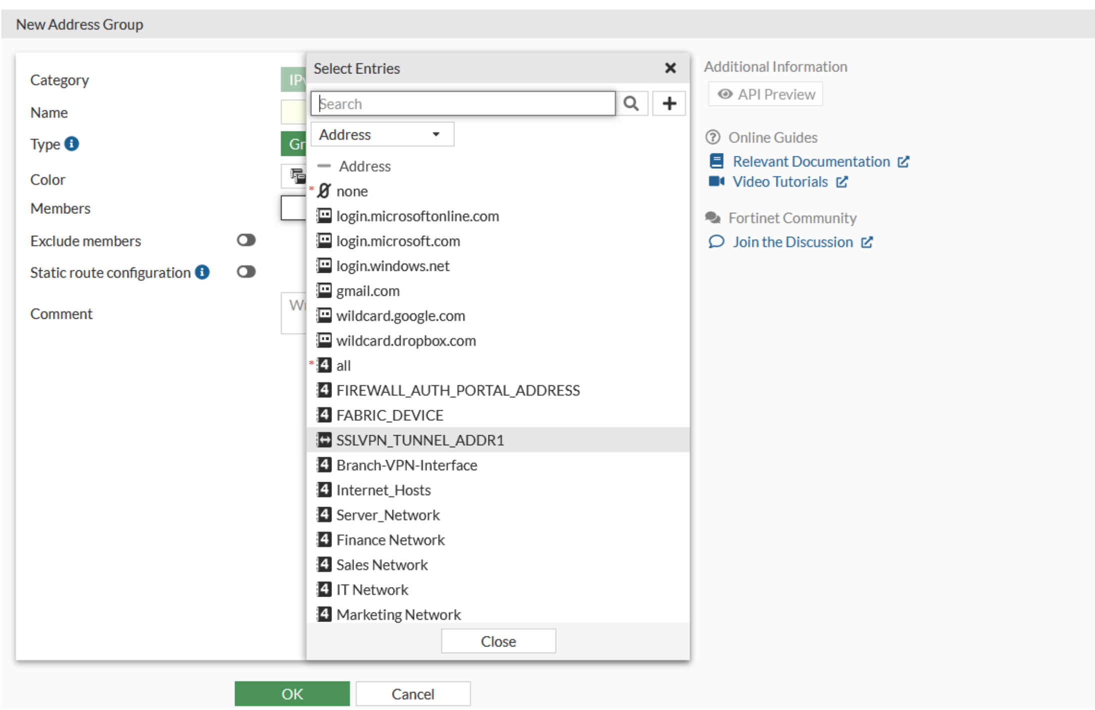


Addresses page

The *Policy & Objects* > *Addresses* page includes tabs for *Address*, *Address Group*, *IPv6 Address*, *IPv6 Address Group*, and *IPv6 Address Template*. If *Explicit Proxy* and/or *Multicast Policy* are enabled from the *System* > *Feature Visibility* page, sub-tabs will appear for *Standard*, *Proxy*, and *Multicast* related address objects.

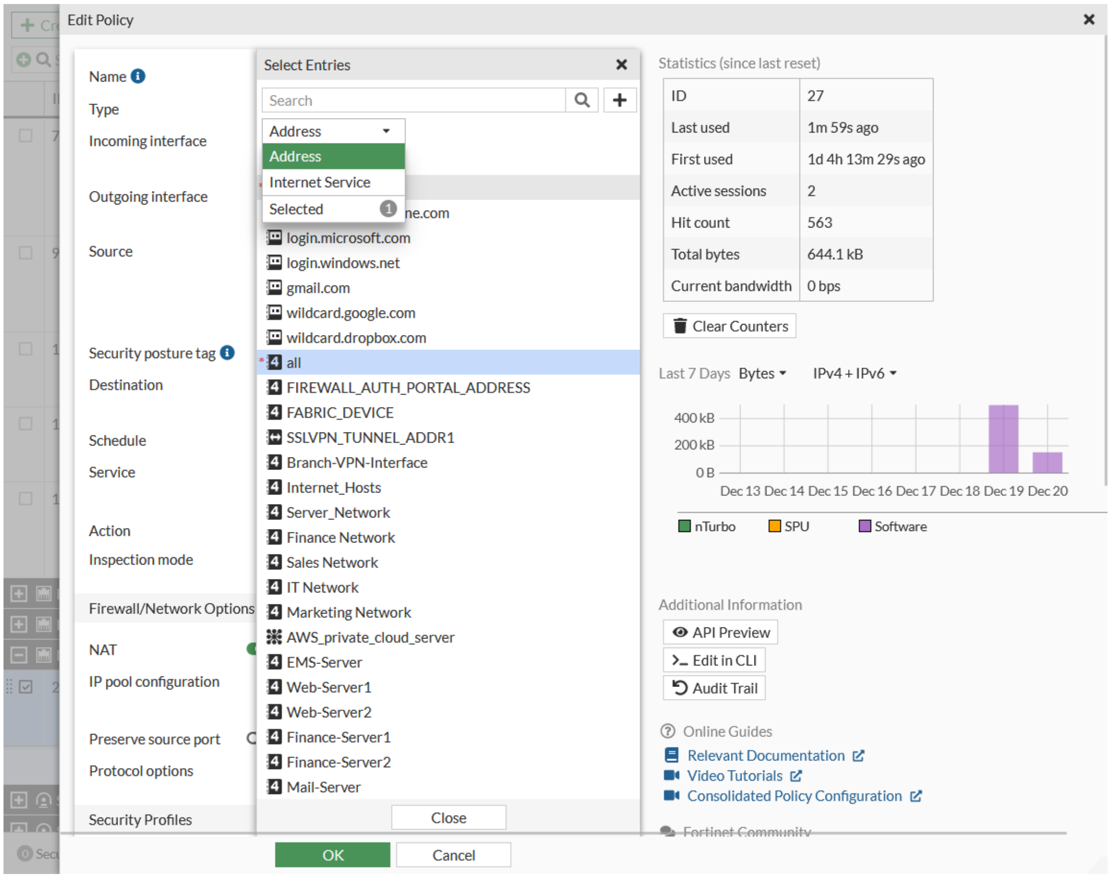


When adding members to an address group, a dropdown is included in the *Select Entries* pane to display specific options.



Firewall Policy dialog

When selecting a source or destination address from the *Policy & Objects > Firewall Policy* page, a dropdown is included in the *Select Entries* pane to display specific options.



Indicate Special Technical Support builds - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- Special Technical Support firmware

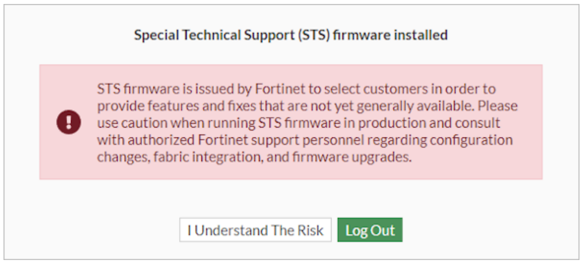
Special Technical Support firmware was formerly known as Top3 builds. When Special Technical Support (STS) firmware is running on FortiGate instead of General Availability (GA) firmware, it is labeled as STS in the FortiOS GUI and CLI, and warning messages about the risks are displayed. STS builds are signed by Fortinet.

Example

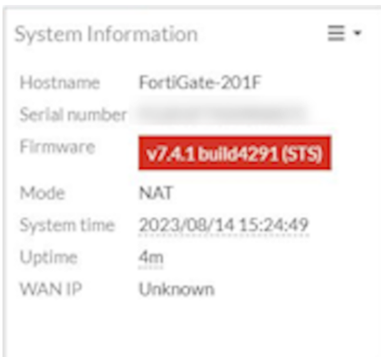
This example shows how to use the FortiOS GUI and CLI to identify when FortiGate is running an STS build of firmware.

To view an STS build in the GUI:

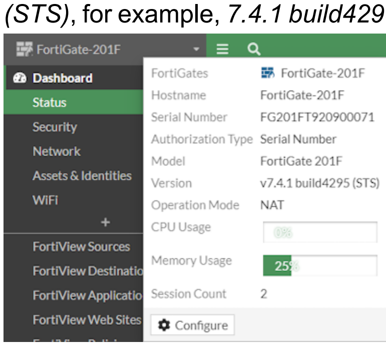
1. Log in to FortiOS. A warning message is displayed.



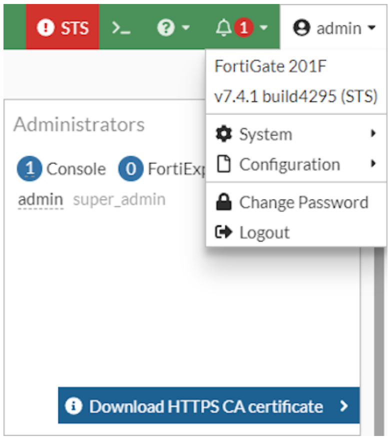
2. Click *I Understand The Risk* to acknowledge the warning and complete the login process.
3. View the STS build label and warnings:
   - Go to *Dashboard > Status > System Information* widget to view the red (STS) label in the *Firmware* field, for example, *7.4.1 build4291 (STS)*.



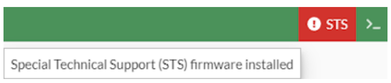
   - Hover over the FortiGate name on the left corner of the banner to display a tooltip. The *Version* fields displays *(STS)*, for example, *7.4.1 build4295 (STS)*.



   - Click *admin* on the top-right of the banner to display a menu. The *(STS)* label appears at the end of the version, for example, *7.4.1 build4295 (STS)*.



- Hover over the exclamation mark beside *STS* on the top-right of the banner. A *Special Technical Support (STS) firmware installed* warning is displayed.



To view an STS build in the CLI:

1. Log in to the CLI. The console prints the following warning message:

```
Login: admin
Password
******WARNING: This is a Special Technical Support (STS) firmware.******
STS firmware is issued by Fortinet to select customers in order to provide features and
fixes that are not yet generally available. Plesae use caution when running STS firmware
in production and consult with authorized Fortinet support personnel regarding
configuration changes, fabric integration, and firmware upgrades.

Welcome!
```

2. Get the system status.

   In this example, the STS build is identified by (STS) and a certified firmware signature.

```
# get system status
Version: FortiGate-201F v7.4.1,build4295,230817 (STS)
Security Level: 2
Firmware Signature: certified
...
```

Network

This section includes information about network related new features:

General

This section includes information about general network related new features:

Using MP-BGP EVPN with VXLAN

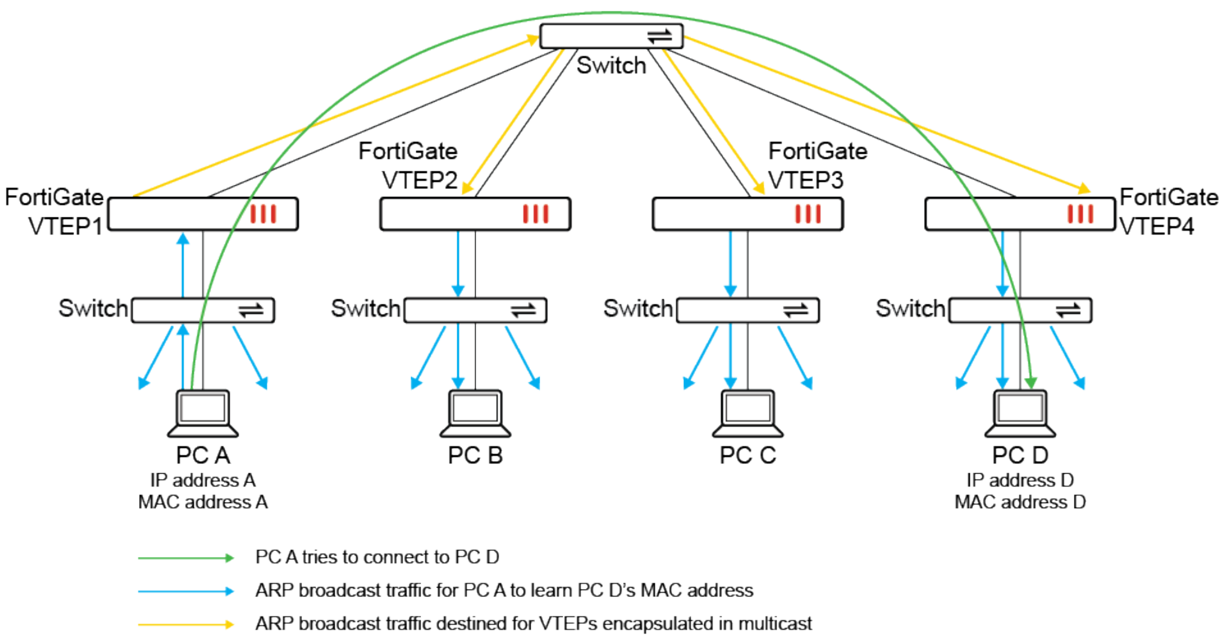
This information is also available in the FortiOS 7.4 Administration Guide:
- VXLAN with MP-BGP EVPN

FortiOS supports VXLAN as implemented according to RFC 7348. Currently, VXLAN relies on determining the MAC address of the destination host by using address resolution protocol (ARP) broadcast frames encapsulated in multicast packets.

- A multicast group is maintained with all the VXLAN tunnel endpoints (VTEPs) associated with the same VXLAN, namely, with the same VXLAN network identifier (VNI).
- The multicast packets that encapsulate ARP broadcast frames are sent to this multicast group, and then the destination host replies to the source host using unicast IP packet encapsulated using VXLAN.
- The source and destination FortiGates as VTEPs each maintain a mapping of MAC addresses to remote VTEPs.

As with non-VXLAN traffic, VXLAN relies on the preceding ARP process, commonly known as flood-and-learn that floods the network with broadcast frames encapsulated as multicast packets to learn MAC addresses. In the RFC 7348 implementation of VXLAN, the data plane is simultaneously used as a control plane.

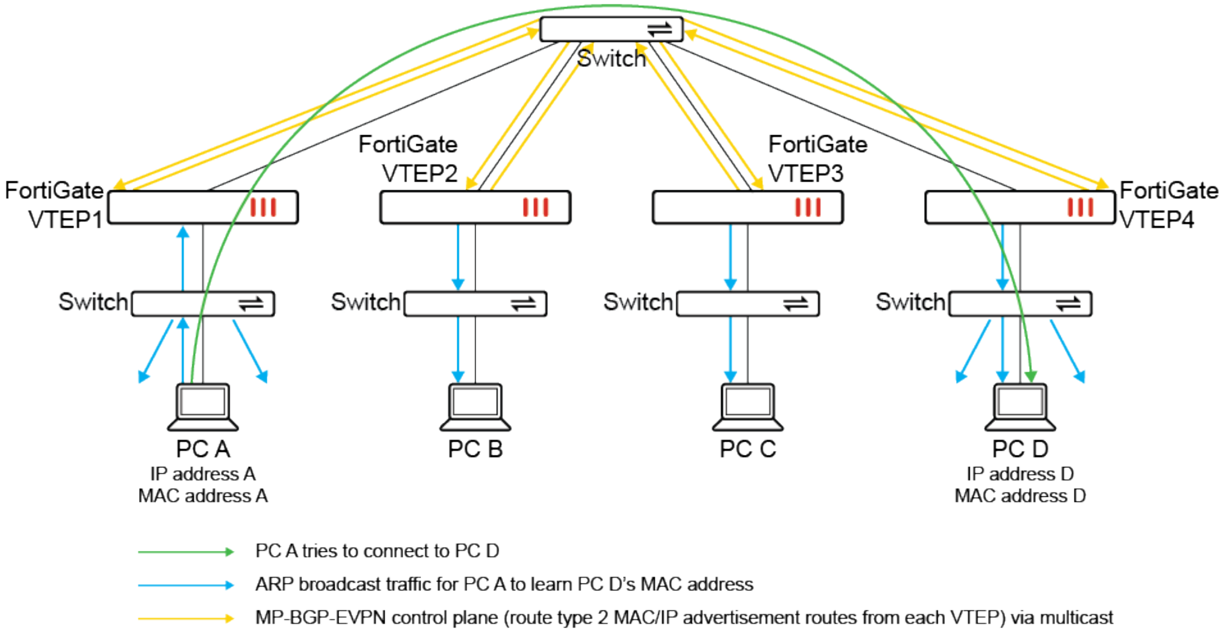
The following topology demonstrates how flood-and-learn uses ARP broadcast traffic flooded throughout the VXLAN for PC A to learn PC D's MAC address when PC A tries to connect to PC D.



In FortiOS 7.4.0, Multiprotocol Border Gateway Protocol Ethernet Virtual Private Network (MP-BGP EVPN) support for VXLAN allows for learning MAC addresses in a way that is more suitable for large deployments than flood-and-learn.

MP-BGP EVPN is a standards-based control plane that supports the distribution of attached host MAC and IP addresses using MP-BGP, namely, using the EVPN address family and MAC addresses treated as routing entries in BGP. As a control plane that is separate from the data plane, MP-BGP EVPN avoids flood-and-learn in the network, and the wide use of BGP as an external gateway protocol on the internet proves its ability to scale well with large deployments. The following topology demonstrates how MP-BGP EVPN distributes route type 2 MAC/IP advertisement routes among

VTEPs in the VXLAN, and minimizes ARP broadcast traffic required for PC A to learn PC D's MAC address when PC A tries to connect to PC D.



MP-BGP EVPN supports the following features:

- Route type 2 (MAC/IP advertisement route) and route type 3 (inclusive multicast Ethernet tag route)
- Intra-subnet communication
- Single-homing use cases
- VLAN-based service, namely, there is only one broadcast domain per EVPN instance (EVI). This is due to the current VXLAN design that supports a single VNI for a VXLAN interface.
- EVPN running on IPv4 unicast VXLAN
- Egress replication for broadcast, unknown unicast, and multicast (BUM) traffic
- VXLAN MAC learning from traffic
- IP address local learning
- ARP suppression

For more information about MP-BGP EVPN, see RFC 7432. For more information about EVPN and VXLAN, see RFC 8365.

Basic MP-BGP EVPN configuration

The MP-BGP EVPN feature builds on the CLI commands used for configuring VXLAN using a VXLAN tunnel endpoint (VTEP). See General VXLAN configuration and topologies in the FortiOS Administration Guide for more details.

After configuring VXLAN using a VTEP, the following CLI commands are configured to enable MP-BGP EVPN on each VTEP.

To configure MP-BGP EVPN on each VTEP:

1. Configure the EVPN settings:

```
config system evpn
    edit <id>
        set rd {AA | AA:NN | A.B.C.D:NN}
        set import-rt <AA:NN>
        set export-rt <AA:NN>
        set ip-local-learning {enable | disable}
        set arp-suppression {enable | disable}
    next
end
```

   The `ip-local-learning` setting is used to enable/disable monitoring the local ARP table of the switch interface to learn the IP/MAC bindings, and advertise them to neighbors. This setting is disabled by default, but must be enabled when configuring MP-BGP EVPN.

   The `arp-suppression` setting is used to enable/disable using proxy ARP to perform suppression of ARP discovery using the flood-and-learn approach. This setting is disabled by default. When enabled, proxy ARP entries are added on the switch interface to suppress the ARP flooding of known IP/MAC bindings, which were learned by the MP-BGP EVPN control plane.

2. Configure the EVPN settings within the VXLAN settings:

```
config system vxlan
    edit <name>
        set interface <string>
        set vni <integer>
        set evpn-id <integer>
        set learn-from-traffic {enable | disable}
    next
end
```

   The `learn-from-traffic` setting is used to enable/disable learning of remote VNIs from VXLAN traffic. This setting is disabled by default, and should only be enabled when local and all remote peers are using same VNI value, and some of the peers do not have MP-BGP EVPN capability.

3. Configure the BGP settings:

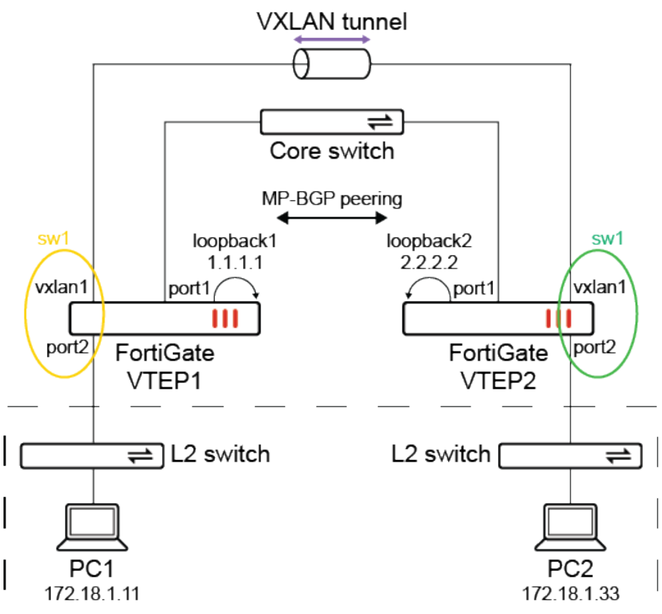
```
config router bgp
    set ibgp-multipath {enable | disable}
    set recursive-next-hop {enable | disable}
    set graceful-restart {enable | disable}
    config neighbor
        edit <WAN_IP_of_other_VTEP>
            set ebgp-enforce-multihop {enable | disable}
            set next-hop-self {enable | disable}
            set next-hop-self-vpnv4 {enable | disable}
            set soft-reconfiguration {enable | disable}
            set soft-reconfiguration-evpn {enable | disable}
            set remote-as <AS_number>
        next
    end
end
```

4. Configure the EVPN setting within the HA settings:

```
config system ha
    set evpn-ttl <integer>
end
```

Example

In this example, two FortiGates are configured as VXLAN tunnel endpoints (VTEPs). A VXLAN is configured to allow L2 connectivity between the networks behind each FortiGate. The VXLAN interface vxlan1 and port2 are placed on the same L2 network using a software switch (sw1). An L2 network is formed between PC1 and PC2. MP-BGP EVPN is used as the control plane to learn and distribute MAC address information within a single L2 domain identified using a specific VNI.



The VTEPs have the following MAC address tables:

Interface/endpoint	VTEP1	VTEP2
vxlan1	82:51:d1:44:bf:93	d2:21:00:c9:e6:98
port2	50:00:00:03:00:01	50:00:00:04:00:01
sw1	50:00:00:03:00:01	50:00:00:04:00:01

The MAC address of PC1 is 00:50:00:00:06:00. The MAC address of PC2 is 00:50:00:00:07:00.

This example assumes that the WAN interface and default route settings have already been configured on the VTEP 1 and VTEP 2 FortiGates. These configurations are omitted from the example. All peers are configured for MP-BGP EVPN.

To configure the VTEP1 FortiGate:

1. Configure the loopback interface:

```
config system interface
    edit "loopback1"
```

```
        set vdom "root"
        set ip 1.1.1.1 255.255.255.255
        set allowaccess ping https ssh http
        set type loopback
    next
end
```

2. Configure the EVPN settings:

```
config system evpn
    edit 100
        set rd "100:100"
        set import-rt "1:1"
        set export-rt "1:1"
        set ip-local-learning enable
        set arp-suppression enable
    next
end
```

3. Configure the local interface and EVPN settings within the VXLAN settings:

```
config system vxlan
    edit "vxlan1"
        set interface "loopback1"
        set vni 1000
        set evpn-id 100
    next
end
```

4. Configure the EVPN settings within the BGP settings:

```
config router bgp
    set as 65001
    set router-id 1.1.1.1
    set ibgp-multipath enable
    set recursive-next-hop enable
    set graceful-restart enable
    config neighbor
        edit "172.25.160.101"
            set ebgp-enforce-multihop enable
            set next-hop-self enable
            set next-hop-self-vpnv4 enable
            set soft-reconfiguration enable
            set soft-reconfiguration-evpn enable
            set remote-as 65001
        next
    end
    config network
        edit 1
            set prefix 1.1.1.1 255.255.255.255
        next
    end
end
```

172.27.16.237 is the WAN IP address of the VTEP2 FortiGate.

5. Configure the software switch:

```
config system switch-interface
    edit "sw1"
        set vdom "root"
        set member "port2" "vxlan1"
        set intra-switch-policy explicit
    next
end
```

6. Configure the software switch interface settings:

```
config system interface
    edit "sw1"
        set vdom "root"
        set ip 172.18.1.253 255.255.255.0
        set allowaccess ping
        set type switch
    next
end
```

7. Configure the firewall policies between the member interfaces in the software switch:

```
config firewall policy
     edit 1
         set srcintf "port2"
         set dstintf "vxlan1"
         set action accept
         set srcaddr "all"
         set dstaddr "all"
         set schedule "always"
         set service "ALL"
     next
     edit 2
         set srcintf "vxlan1"
         set dstintf "port2"
         set action accept
         set srcaddr "all"
         set dstaddr "all"
         set schedule "always"
         set service "ALL"
     next
end
```

To configure the VTEP2 FortiGate:

1. Configure the loopback interface:

```
config system interface
    edit "loopback2"
        set vdom "root"
        set ip 2.2.2.2 255.255.255.255
        set allowaccess ping https ssh http
        set type loopback
    next
end
```

2. Configure the EVPN settings:

```
config system evpn
    edit 100
        set rd "100:100"
        set import-rt "1:1"
        set export-rt "1:1"
        set ip-local-learning enable
        set arp-suppression enable
    next
end
```

3. Configure the local interface and EVPN settings within the VXLAN settings:

```
config system vxlan
    edit "vxlan1"
        set interface "loopback2"
        set vni 1000
        set evpn-id 100
    next
end
```

4. Configure the EVPN settings within the BGP settings:

```
config router bgp
    set as 65001
    set router-id 2.2.2.2
    set ibgp-multipath enable
    set recursive-next-hop enable
    set graceful-restart enable
    config neighbor
        edit "172.25.160.100"
            set ebgp-enforce-multihop enable
            set next-hop-self enable
            set next-hop-self-vpnv4 enable
            set soft-reconfiguration enable
            set soft-reconfiguration-evpn enable
            set remote-as 65001
        next
    end
    config network
        edit 1
            set prefix 2.2.2.2 255.255.255.255
        next
    end
end
```

172.27.16.236 is the WAN IP address of the VTEP1 FortiGate.

5. Configure the software switch:

```
config system switch-interface
    edit "sw1"
        set vdom "root"
        set member "port2" "vxlan1"
        set intra-switch-policy explicit
    next
end
```

6. Configure the software switch interface settings:

```
config system interface
    edit "sw1"
        set vdom "root"
        set ip 172.18.1.254 255.255.255.0
        set allowaccess ping
        set type switch
    next
end
```

7. Configure the firewall policies between the member interfaces in the software switch:

```
config firewall policy
     edit 1
         set srcintf "port2"
         set dstintf "vxlan1"
         set action accept
         set srcaddr "all"
         set dstaddr "all"
         set schedule "always"
         set service "ALL"
     next
     edit 2
         set srcintf "vxlan1"
         set dstintf "port2"
         set action accept
         set srcaddr "all"
         set dstaddr "all"
         set schedule "always"
         set service "ALL"
     next
end
```

To verify the MP-BGP EVPN status on the VTEP1 FortiGate:

1. From a host computer with IP address 172.18.1.11, perform the following.
   a. Check the ARP cache:

```
# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
172.18.1.253             ether   50:00:00:03:00:01   C                     ens3
```

   b. Ping the host computer with IP address 172.18.1.33:

```
# ping 172.18.1.33 -c 4
PING 172.18.1.33 (172.18.1.33) 56(84) bytes of data.
64 bytes from 172.18.1.33: icmp_seq=1 ttl=64 time=1325 ms
64 bytes from 172.18.1.33: icmp_seq=2 ttl=64 time=319 ms
64 bytes from 172.18.1.33: icmp_seq=3 ttl=64 time=3.96 ms
64 bytes from 172.18.1.33: icmp_seq=4 ttl=64 time=1.66 ms

--- 172.18.1.33 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 1.660/412.614/1325.209/542.530 ms
```

   c. Check the ARP cache again:

```
# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
```

```
172.18.1.33             ether   00:50:00:00:07:00   C                      ens3
172.18.1.253            ether   50:00:00:03:00:01   C                      ens3
```

2. On the VTEP1 FortiGate, run the switch and VXLAN debug commands.
   a. Verify the forwarding database for vxlan1:

```
# diagnose sys vxlan fdb list vxlan1
mac=00:00:00:00:00:00 state=0x0082 remote_ip=2.2.2.2 port=4789 vni=1000 ifindex0
mac=00:50:00:00:07:00 state=0x0082 remote_ip=2.2.2.2 port=4789 vni=1000 ifindex0

total fdb num: 2
```

   b. Verify the forwarding database statistics for vxlan1:

```
# diagnose sys vxlan fdb stat vxlan1
fdb_table_size=256 fdb_table_used=2 fdb_entry=2 fdb_max_depth=1 cleanup_idx=0 c2
```

   c. Verify the bridging information for sw1:

```
# diagnose netlink brctl name host sw1
show bridge control interface sw1 host.
fdb: hash size=32768, used=5, num=5, depth=1, gc_time=4, ageing_time=3, arp-sups
Bridge sw1 host table
port no device  devname mac addr                ttl     attributes
  2     15      vxlan1  00:00:00:00:00:00       28       Hit(28)
  2     15      vxlan1  00:50:00:00:07:00       18       Hit(18)
  2     15      vxlan1  82:51:d1:44:bf:93       0       Local Static
  1     4       port2   00:50:00:00:06:00       14       Hit(14)
  1     4       port2   50:00:00:03:00:01       0       Local Static
```

3. Run the BGP EVPN commands and observe the route type 2 (MAC/IP advertisement route) and route type 3 (inclusive multicast Ethernet tag route).
   a. Verify the BGP L2 VPN EVPN summary information:

```
# get router info bgp evpn summary

VRF 0 BGP router identifier 1.1.1.1, local AS number 65001
BGP table version is 2
1 BGP AS-PATH entries
0 BGP community entries

Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/Pd
172.25.160.101 4      65001       9       9        1    0    0 00:04:02        3

Total number of neighbors 1
```

   b. Verify the BGP L2 VPN EVPN network information:

```
# get router info bgp evpn network
   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
Route Distinguisher: 100:100 (Default for VRF 0)
*> [2][0][48][00:50:00:00:06:00][0]/72
                    1.1.1.1             0             100  32768        0 i <-/>
*> [2][0][48][00:50:00:00:06:00][32][172.18.1.11]/104
                    1.1.1.1             0             100  32768        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][0]/72
                    2.2.2.2             0             100      0        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][32][172.18.1.33]/104
```

```
                    2.2.2.2             0             100      0        0 i <-/>
*>  [3][0][32][1.1.1.1]/80
                    1.1.1.1             0             100  32768        0 i <-/>
*>i[3][0][32][2.2.2.2]/80
                    2.2.2.2             0             100      0        0 i <-/>

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
Route Distinguisher: 100:100 (received from VRF 0)
*>i[2][0][48][00:50:00:00:07:00][0]/72
                    2.2.2.2             0             100      0        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][32][172.18.1.33]/104
                    2.2.2.2             0             100      0        0 i <-/>
*>i[3][0][32][2.2.2.2]/80
                    2.2.2.2             0             100      0        0 i <-/>
```

c. Verify the BGP L2 VPN EVPN context:

```
# get router info bgp evpn context
L2VPN EVPN context for VRF 0
ID 100 vlan-based, RD is [100:100]
 Import RT: RT:1:1
 Export RT: RT:1:1
  Bridge domain 0 VNI 1000
   Encapsulation 8(VXLAN)
   Source interface loopback1
   Source address 1.1.1.1
```

d. Verify the neighbor EVPN routes:

```
# get router info bgp neighbors 172.25.160.101 routes evpn
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight RouteTag Path
Route Distinguisher: 100:100 (Default for VRF 0) (Default for VRF 0)
*>i[2][0][48][00:50:00:00:07:00][0]/72
                    2.2.2.2             0             100      0        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][32][172.18.1.33]/104
                    2.2.2.2             0             100      0        0 i <-/>
*>i[3][0][32][2.2.2.2]/80
                    2.2.2.2             0             100      0        0 i <-/>
Route Distinguisher: 100:100 (received from VRF 0) (received from VRF 0)
*>i[2][0][48][00:50:00:00:07:00][0]/72
                    2.2.2.2             0             100      0        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][32][172.18.1.33]/104
                    2.2.2.2             0             100      0        0 i <-/>
*>i[3][0][32][2.2.2.2]/80
                    2.2.2.2             0             100      0        0 i <-/>

Total number of prefixes 6
```

4. Run the following EVPN get commands.

a. Verify the EVPN instances:

```
# get l2vpn evpn instance
EVPN instance: 100
```

```
IP local learning enabled
ARP suppression enabled
HA primary
   Number of bridge domain: 1
   Bridge domain: TAGID 0 VNI 1000 ADDR 1.1.1.1 VXLAN vxlan1 SWITCH sw1
```

b. Verify the EVPN table:

```
# get l2vpn evpn table
EVPN instance 100
 Broadcast domain VNI 1000 TAGID 0

  EVPN instance 100
 Broadcast domain VNI 1000 TAGID 0

  EVPN MAC table:
  MAC               VNI      Remote Addr     Binded Address
  00:50:00:00:07:00 1000     2.2.2.2         172.18.1.33
                    1000     2.2.2.2         -

  EVPN IP table:
  Address         VNI      Remote Addr     MAC
  172.18.1.33     1000     2.2.2.2         00:50:00:00:07:00

  EVPN Local MAC table:
  "Inactive" means this MAC/IP pair will not be sent to peer.
  Flag code: S - Static F - FDB. Trailing * means HA
  MAC               Flag Status   Binded Address
  00:50:00:00:06:00      Active   172.18.1.11
                    F    Active   -

  EVPN Local IP table:
  Address         MAC
  172.18.1.11     00:50:00:00:06:00

  EVPN PEER table:
  VNI      Remote Addr     Binded Address
  1000     2.2.2.2         2.2.2.2
```

5. Run the proxy ARP diagnose command:

```
# diagnose ip parp list
Address            Hardware Addr     Interface
172.18.1.33        00:50:00:00:07:00 sw1
```

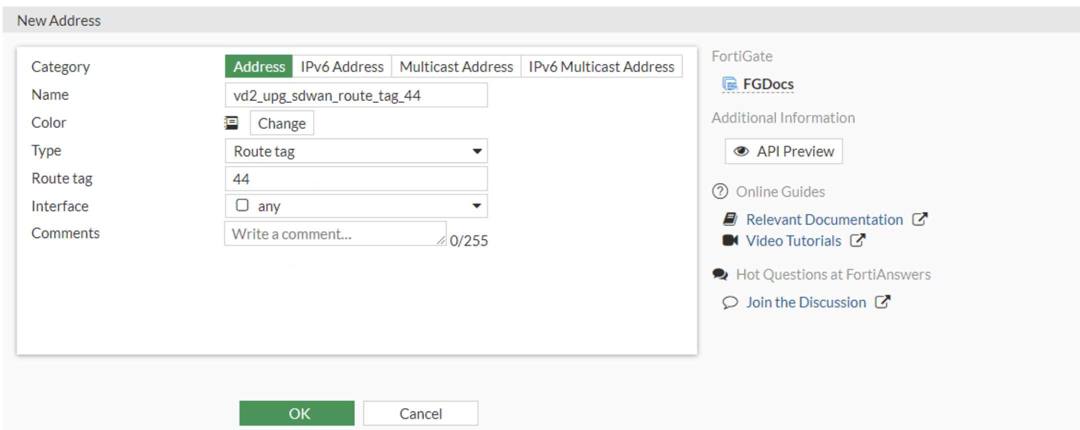
Add route tag address objects

A route tag (`route-tag`) firewall address object can include IPv4 or IPv6 addresses associated with a BGP route tag number, and is updated dynamically with BGP routing updates. The route tag firewall address object allows for a more dynamic and flexible configuration that does not require manual intervention to dynamic routing updates. This address object can be used wherever a firewall address can be used, such as in a firewall policy, a router policy, or an SD-WAN service rule.

The *Route tag* field has been removed from the *Priority Rule* configuration page (*Network > SD-WAN > SD-WAN Rules*). The `route-tag` option has been removed from the `config service settings` under `config system sdwan`.

To configure and apply a route tag address object in the GUI:

1. Configure the route tag address object:
   a. Go to *Policy & Objects* > *Addresses* and click *Create New* > *Address*.
   b. Enter a *Name*, such as *vd2_upg_sdwan_route_tag_44*.
   c. Set the *Type* to *Route tag*.
   d. Enter the *Route tag* number, such as *44*.



   e. Click *OK*.
2. Add the address to a firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy*.
   b. Edit an existing policy or create a new one.
   c. Set the *Destination* to *vd2_upg_sdwan_route_tag_44*.
   d. Configure the other settings as needed.
   e. Click *OK*.
3. Add the address to an SD-WAN service rule:
   a. Go to *Network* > *SD-WAN* and select the *SD-WAN Rules* tab.
   b. Edit an existing rule or create a new one.
   c. In the *Destination* section, set the *Address* to *vd2_upg_sdwan_route_tag_44*.
   d. Configure the other settings as needed.
   e. Click *OK*.

To configure and apply a route tag address object in the CLI:

1. Configure the route tag address object:

```
config firewall address
    edit "vd2_upg_sdwan_route_tag_44"
        set type route-tag
        set route-tag 44
```

```
    next
end
```

2. Add the address to a firewall policy:

```
config firewall policy
    edit 3
        set srcintf "any"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "vd2_upg_sdwan_route_tag_44"
        set schedule "always"
        set service "ALL"
    next
end
```

3. Add the address to an SD-WAN service rule:

```
config system sdwan
    config service
        edit 1
            set dst "vd2_upg_sdwan_route_tag_44"
            set priority-members 1
        next
    end
end
```

To verify the configuration:

1. After some traffic passes, verify that the route tag firewall address is associated with policy ID 3:

```
# diagnose firewall iprope list | grep -A 15 index=3
policy index=3 uuid_idx=754 action=accept
flag (8010008): redir master pol_stats
flag2 (4000): resolve_sso
flag3 (100000a0): link-local best-route no-vwp
schedule(always)
cos_fwd=255  cos_rev=255
group=00100004 av=00004e20 au=00000000 split=00000000
host=5 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 0 -> zone(1): 0
source(1): 0.0.0.0-255.255.255.255, uuid_idx=684,
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto
 route_tag(1):  44
```

2. Verify the list of firewall route tag addresses:

```
# diagnose firewall route_tag list
list route tag info(vf(vd2)):
route tag address, route_tag(30) vrf_num(1):
vrf id(0), num(2): 11.11.11.11-11.11.11.11 100.1.1.0-100.1.1.255

route tag address, route_tag(33) vrf_num(1):
vrf id(0), num(1): 33.1.1.0-33.1.1.255
```

```
route tag address, route_tag(40) vrf_num(1):
vrf id(0), num(2): 11.11.11.11-11.11.11.11 100.1.1.0-100.1.1.255

route tag address, route_tag(44) vrf_num(1):
vrf id(0), num(1): 33.1.1.0-33.1.1.255
```

Configuring a DHCP shared subnet

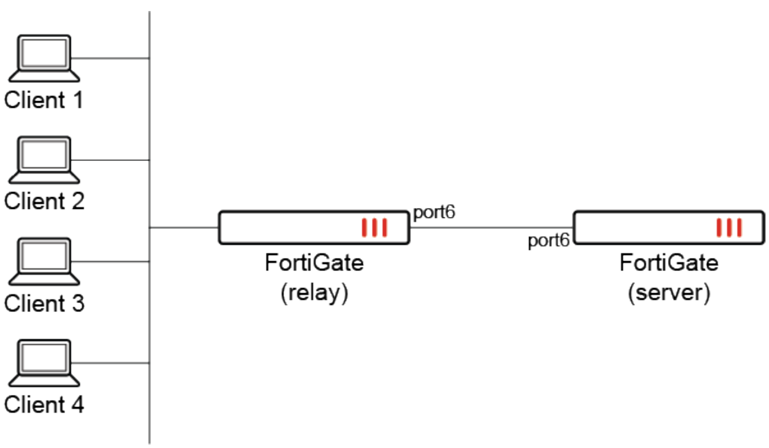
This information is also available in the FortiOS 7.4 Administration Guide:
- DHCP shared subnet

A FortiGate can act as a DHCP server and assign IP addresses from different subnets to clients on the same interface or VLAN based on the requests coming from the same DHCP relay agent. A FortiGate may have more than one server and pool associated with the relay agent, and it can assign IP addresses from the next server when the current one is exhausted. This way, the FortiGate can allocate IP addresses more efficiently and avoid wasting unused addresses in each subnet.

```
config system dhcp server
    edit <id>
        set shared-subnet {enable | disable}
        set relay-agent <ip_address>
    next
end
```

Example

In this example, there are two DHCP servers configured on the FortiGate. The first two clients (1 and 2) get their IP from the DHCP server 1. Once the DHCP server 1's IP pool is exhausted, subsequent clients (3 and 4) get their IP from DHCP server 2.



To configure a DHCP shared subnet:

1. Configure the DHCP servers:

```
config system dhcp server
    edit 1
        set default-gateway 10.18.0.10
        set netmask 255.255.255.0
        set interface "p2_vl3819"
        config ip-range
            edit 1
                set start-ip 10.18.0.110
                set end-ip 10.18.0.111
            next
        end
        set shared-subnet enable
        set relay-agent 10.18.0.10
        set dns-server1 8.8.8.8
    next
    edit 2
        set default-gateway 10.18.1.130
        set netmask 255.255.255.128
        set interface "p2_vl3819"
        config ip-range
            edit 1
                set start-ip 10.18.1.200
                set end-ip 10.18.1.201
            next
        end
        set shared-subnet enable
        set relay-agent 10.18.0.10
        set dns-server1 8.8.8.8
    next
end
```

2. Verify the DHCP lease list:

```
# execute dhcp lease-list
port6
  IP            MAC-Address       Hostname    VCI    SSID    AP     SERVER-ID    Expiry
  10.18.0.110  00:50:56:02:92:11                                    1            Fri Jan 13
15:37:35 2023
  10.18.0.111  00:50:56:02:92:12                                    1            Fri Jan 13
15:37:38 2023

Result: PASS
```

Clients 1 and 2 get their IP from the DHCP server 1.

When the IP pool is exhausted, the DHCP daemon assigns the IP from other pools that have the same relay agent.

3. Verify the DHCP lease list:

```
# execute dhcp lease-list
port6
  IP            MAC-Address       Hostname    VCI    SSID    AP     SERVER-ID    Expiry
  10.18.0.110  00:50:56:02:92:11                                    1            Fri Jan 13
15:37:35 2023
  10.18.0.111  00:50:56:02:92:12                                    1            Fri Jan 13
```

```
15:37:38 2023
  10.18.1.200  00:50:56:02:92:13                              2             Fri Jan 13
15:38:05 2023
  10.18.1.201  00:50:56:02:92:14                              2             Fri Jan 13
15:38:06 2023
```

Clients 3 and 4 get their IP from DHCP server 2, since the server 1 IP pool is exhausted.

Configuring DHCP smart relay on interfaces with a secondary IP

This information is also available in the FortiOS 7.4 Administration Guide:

- DHCP smart relay on interfaces with a secondary IP

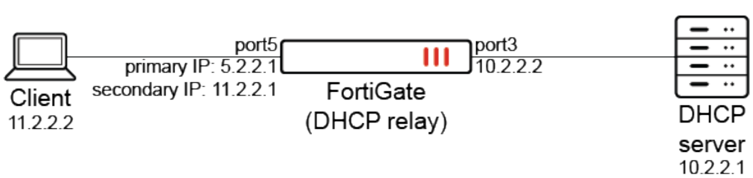
DHCP relays can be configured on interfaces with secondary IP addresses. The FortiGate will track the number of unanswered DHCP requests for a client on the interface's primary IP. After three unanswered DHCP requests, the FortiGate will forward DHCP requests to DHCP relays configured under the secondary IP using the secondary IP address as the source. After three unanswered DHCP requests, the FortiGate will return to using the primary IP and restart the process.

```
config system interface
    edit <name>
        set dhcp-smart-relay {enable | disable}
        config secondaryip
            edit <id>
                set secip-relay-ip <secondary_dhcp_relay_IP_1> <secondary_dhcp_relay_IP_2>
            next
        end
    next
end
```

DHCP relay targets under both the primary and secondary IP may be the same or unique. If smart relay is not configured, all requests are forwarded using the primary IP address on the interface.

Example

In this example, DHCP smart relay is configured on port5 with a DHCP relay IP address of 10.2.2.1.



To configure DHCP smart relay on interfaces with a secondary IP:

1. Configure DHCP relay on the interfaces:

```
config system interface
    edit "port3"
        set vdom "vdom1"
```

```
        set ip 10.2.2.2 255.255.255.0
        set allowaccess ping https ssh snmp http telnet
        set type physical
        set snmp-index 5
    next
    edit "port5"
        set vdom "vdom1"
        set dhcp-relay-service enable
        set dhcp-smart-relay enable
        set ip 5.2.2.1 255.255.255.0
        set allowaccess ping https ssh snmp http
        set type physical
        set snmp-index 7
        set secondary-IP enable
        set dhcp-relay-ip "10.2.2.1"
        config secondaryip
            edit 1
                set ip 11.2.2.1 255.255.255.0
                set secip-relay-ip "10.2.2.1"
                set allowaccess ping https ssh snmp http
            next
        end
    next
end
```

2. Verify the debug messages to check that the DHCP relay is working. After three unanswered DHCP requests, the request is forwarded to the secondary IP DHCP relay target:

```
# diagnose debug application dhcprelay -1
Debug messages will be on for 30 minutes.

(xid:7ea80e4b) received request message from 0.0.0.0:68 to 255.255.255.255 at port5
(xid:7ea80e4b) got a DHCPDISCOVER
(xid:7ea80e4b) Warning! can't get server id from client message
Insert option(82), len(7)
found route to 10.2.2.1 via 10.2.2.2 iif=11 oif=9/port3, mode=auto, ifname=
(xid:7ea80e4b) forwarding dhcp request from 5.2.2.1:67 to 10.2.2.1:67
(xid:7ea80e4b) received request message from 0.0.0.0:68 to 255.255.255.255 at port5
(xid:7ea80e4b) got a DHCPDISCOVER
(xid:7ea80e4b) Warning! can't get server id from client message
Insert option(82), len(7)
found route to 10.2.2.1 via 10.2.2.2 iif=11 oif=9/port3, mode=auto, ifname=
(xid:7ea80e4b) forwarding dhcp request from 5.2.2.1:67 to 10.2.2.1:67
(xid:7ea80e4b) received request message from 0.0.0.0:68 to 255.255.255.255 at port5
(xid:7ea80e4b) got a DHCPDISCOVER
(xid:7ea80e4b) Warning! can't get server id from client message
Insert option(82), len(7)
found route to 10.2.2.1 via 10.2.2.2 iif=11 oif=9/port3, mode=auto, ifname=
(xid:7ea80e4b) forwarding dhcp request from 11.2.2.1:67 to 10.2.2.1:67
(xid:7ea80e4b) received request message from 10.2.2.1:67 to 11.2.2.1 at port3
(xid:7ea80e4b) got a DHCPOFFER
(xid:7ea80e4b) from server 10.2.2.1
(xid:7ea80e4b) sending dhcp reply from 11.2.2.1:67 to 255.255.255.255:68
(xid:7ea80e4b) received request message from 0.0.0.0:68 to 255.255.255.255 at port5
(xid:7ea80e4b) got a DHCPREQUEST
Insert option(82), len(7)
```

```
found route to 10.2.2.1 via 10.2.2.2 iif=11 oif=9/port3, mode=auto, ifname=
(xid:7ea80e4b) forwarding dhcp request from 11.2.2.1:67 to 10.2.2.1:67
(xid:7ea80e4b) received request message from 10.2.2.1:67 to 11.2.2.1 at port3
(xid:7ea80e4b) got a DHCPACK
(xid:7ea80e4b) from server 10.2.2.1
(xid:7ea80e4b) sending dhcp reply from 11.2.2.1:67 to 255.255.255.255:68
```

Improve DVLAN QinQ performance for NP7 platforms over virtual wire pairs

This information is also available in the FortiOS 7.4 Administration Guide:
- DVLAN QinQ on NP7 platforms over virtual wire pairs

DVLAN 802.1ad and 802.1Q modes are supported on NP7 platforms over virtual wire pairs, which provides better performance and packet processing.

The default DVLAN mode is 802.1ad, but the DVLAN mode can be changed using `diagnose npu np7 dvlan-mode <dvlan_mode> {<npid> | all}`. The DVLAN mode can be applied to a specific NPID or all NPIDs. For example:

- `diagnose npu np7 dvlan-mode 802.1AD 0` will set NP0 to work in 802.1ad mode.
- `diagnose npu np7 dvlan-mode 802.1Q all` will set all NPUs to work in 802.1Q mode.

A reboot is required for custom DVLAN settings to take effect. To avoid any inconveniences or disruptions, changing the DVLAN settings should be done during a scheduled downtime or maintenance window.

The DVLAN mode should only be changed if you are solely using the virtual wire pair (VWP) and are seeking to enhance performance. Enabling this feature may impact VLAN interfaces within your network.

In the virtual wire pair settings, the `outer-vlan-id` can be set. This is the same value as the outer provider-tag (S-Tag).

To configure the outer VLAN ID:

```
config system virtual-wire-pair
    edit "dvlan-test"
        set member "port33" "port34"
        set wildcard-vlan enable
        set outer-vlan-id 1234
    next
end
```

Active SIM card switching available on FortiGates with cellular modem and dual SIM card support

This information is also available in the FortiOS 7.4 Administration Guide:
- Active SIM card switching

FortiGates with a cellular modem and dual SIM card can switch in real time from the active SIM card to the passive SIM card when any of the following issues arise with the active SIM card:

- Ping link monitor fails. The SIM switch time depends on the link monitor parameters set.
- An active SIM card cannot be detected. The SIM switch time is about 20 seconds after the SIM card is no longer detected.
- A modem disconnection is detected, and a specified interval has elapsed. The SIM switch time occurs after the specified interval.

SIM card switching events are captured in the FortiGate event log.

In most cases, SIM cards come with the wireless carrier's APN, which is automatically retrieved at the first connection of the LTE modem. For these cases, you can use SIM cards for different wireless carriers in SIM slot 1 and slot 2.

When one or both SIM cards require their APN settings to be configured on the FortiGate, then both SIM cards should be for the same wireless carrier because `config system lte-modem` currently only supports a single `set apn < apn >` setting.

The following command and options can be used to configure this feature:

```
config system lte-modem
    config sim-switch
        set by-sim-state {enable | disable}
        set by-connection-state {enable | disable}
        set by-link-monitor {enable | disable}
        set link-monitor <link-monitor-name>
        set sim-switch-log-alert-interval <interval>
        set sim-switch-log-alert-threshold <threshold>
        set modem-disconnection-time <integer>
    end
end
```

`by-sim-state {enable | disable}`	Enable switching based on active SIM card state: • `enable`: switch to the passive SIM card whenever FortiGate cannot detect the active SIM card, such as when the active SIM card is ejected. • `disable`: do not switch SIM cards based on state.
`by-connection-state {enable | disable}`	Enable switching based on the connection state of the active SIM card: • `enable`: switch to the passive SIM card whenever FortiGate detects a modem signal loss after the `modem-disconnection-time` expires. • `disable`: do not switch SIM cards based on the connection state.
`by-link-monitor {enable | disable}`	Enable switching when a configured link monitor fails: • `enable`: switch to the passive SIM card when a link monitor configured with `link-monitor-name` fails. • `disable`: do not switch SIM cards based on the failure of a configured link monitor.
`link-monitor <link-monitor-name>`	Specify the name of the link monitor to use with `by-link-monitor`.
`sim-switch-log-alert-interval <interval>`	Identify what number of constant SIM card switch events will trigger an event log after the threshold in `sim-switch-log-alert-threshold` is met.

`sim-switch-log-alert-threshold`	Specify how many minutes to wait before creating an event log when the number of SIM card switches defined in `sim-switch-log-alert-interval` is met.
`modem-disconnection-time <integer>`	Specify how many seconds to wait before switching over to the passive SIM card when `by-connection-state` is enabled and a modem signal loss is detected.

Example 1

In this example, automatic SIM card switching is disabled. When disabled, the SIM card only works in the default slot1, but you can manually switch the SIM card to slot2. Event logs include details about the SIM card switch.

To manually switch a SIM card:

1. Disable automatic SIM card switching:

```
config system lte-modem
    config sim-switch
        set by-sim-state disable
        set by-connection-state disable
        set by-link-monitor disable
        set sim-slot 1
    end
end
```

2. Manually switch the SIM card from slot1 to slot2, and run the following command:

```
# execute lte-modem sim-switch
```

The SIM card switch may take a few seconds. You can run `diagnose system lte-modem sim-info` to check the results.

The following log is generated after unplugging an active SIM card:

```
7: date=2023-05-02 time=10:41:05 eventtime=1683049264795418820 tz="-0700"
logid="0100046518" type="event" subtype="system" level="information" vd="root"
logdesc="LTE modem active SIM card switch event" msg="LTE modem active SIM card slot
changed to 2 by user."
```

Example 2

In this section, automatic SIM card switching is enabled and configured to switch based on SIM state, connection state, or link monitor state, and it includes example event logs for each scenario.

To enable automatic SIM card switching by SIM state:

1. Enable automatic SIM card switching by SIM state:

```
config system lte-modem
    config sim-switch
        set by-sim-state enable
    end
end
```

With this configuration, the second SIM card becomes active when the active SIM card is no longer detected, for example, if the active SIM card is ejected. The following event logs are generated:

```
5: date=2023-04-28 time=17:27:27 eventtime=1682728046989682780 tz="-0700"
logid="0100046513" type="event" subtype="system" level="information" vd="root"
logdesc="LTE modem data link connection event" msg="LTE modem data link changed from
QMI_WDS_CONNECTION_STATUS_DISCONNECTED to QMI_WDS_CONNECTION_STATUS_CONNECTED"

6: date=2023-04-28 time=17:27:17 eventtime=1682728036493684280 tz="-0700"
logid="0100046512" type="event" subtype="system" level="information" vd="root"
logdesc="LTE modem SIM card state event" msg="LTE modem SIM card change from QMI_UIM_
CARD_STATE_ABSENT to QMI_UIM_CARD_STATE_PRESENT"

7: date=2023-04-28 time=17:27:12 eventtime=1682728032589776580 tz="-0700"
logid="0100046513" type="event" subtype="system" level="information" vd="root"
logdesc="LTE modem data link connection event" msg="LTE modem data link changed from
QMI_WDS_CONNECTION_STATUS_CONNECTED to QMI_WDS_CONNECTION_STATUS_DISCONNECTED"

8: date=2023-04-28 time=17:27:11 eventtime=1682728031245682560 tz="-0700"
logid="0100046512" type="event" subtype="system" level="information" vd="root"
logdesc="LTE modem SIM card state event" msg="LTE modem SIM card change from QMI_UIM_
CARD_STATE_PRESENT to QMI_UIM_CARD_STATE_ABSENT"
```

To enable automatic SIM card switching by connection state:

1. Enable automatic SIM card switching by connection state:

```
config system lte-modem
    config sim-switch
        set by-connection-state enable
        set modem-disconnection-time 30
        set sim-switch-log-alert-interval 15
        set sim-switch-log-alert-threshold 5
    end
end
```

With this configuration, the second SIM card becomes active when the modem cannot establish a connection with the carrier through the active SIM card. For example, a FortiGate is in a room with poor signal quality. With this configuration, the SIM card switch is triggered after the modem is detected as disconnected for 30 seconds, and the following event log is generated:

```
56: date=2023-05-01 time=11:14:56 eventtime=1682964896356933480 tz="-0700"
logid="0100046519" type="event" subtype="system" level="notice" vd="root" logdesc="LTE
modem active SIM card switched: modem disconnection detected" msg="LTE modem active SIM
card slot changed to 2, due to modem connection down."

66: date=2023-05-01 time=11:14:13 eventtime=1682964852964869400 tz="-0700"
logid="0100046519" type="event" subtype="system" level="notice" vd="root" logdesc="LTE
modem active SIM card switched: modem disconnection detected" msg="LTE modem active SIM
card slot changed to 1, due to modem connection down."
```

When poor signal quality causes SIM cards to frequently switch back and forth, and the flapping rate occurs more than five times within the configured 15 minute time period, an event log is triggered to record the flapping severity:

```
65: date=2023-05-01 time=11:14:13 eventtime=1682964853083194400 tz="-0700"
logid="0100046521" type="event" subtype="system" level="warning" vd="root" logdesc="LTE
modem active SIM card slot flipped back and forth in short time" msg="LTE modem switched
SIM slot 8 times in last 15 minutes, which is greater than 5 times threshold."
```

To enable automatic SIM card switching based on link monitor:

1. Enable automatic SIM card switching by link monitor, and specify the link monitor:

```
config system lte-modem
    config sim-switch
        set by-link-monitor enable
        set link-monitor "modem"
        set sim-switch-log-alert-interval 15
        set sim-switch-log-alert-threshold 5
    end
    config system link-monitor
    edit "modem"
        set srcintf "wwan"
        set server "8.8.8.8"
        set interval 1000
        set probe-timeout 100
        set failtime 3
        set recoverytime 8
    next
end
```

   With this configuration, the second SIM card becomes active when the link monitor detects the active SIM card exceeds the SLA.

2. Check the link monitor status. In this example, the link monitor status is dead:

```
# diagnose system link-monitor status modem

Link Monitor: modem, Status: dead, Server num(1), cfg_version=7 HA state: local(dead),
shared(dead)
Flags=0x9 init log_downgateway, Create time: Fri Apr 28 16:34:56 2023
Source interface: wwan (19)
VRF: 0
Interval: 1000 ms
Service-detect: disable
Diffservcode: 000000
Class-ID: 0
  Peer: 8.8.8.8(8.8.8.8)
        Source IP(10.192.195.164)
        Route: 10.192.195.164->8.8.8.8/32, gwy(10.192.195.165)
        protocol: ping, state: dead
                Packet lost: 11.667%
                MOS: 4.353
                Number of out-of-sequence packets: 0
                Recovery times(5/8) Fail Times(1/3)
                Packet sent: 60, received: 56, Sequence(sent/rcvd/exp): 61/61/62
```

   The following event log is generated when the link-monitor status is dead:

```
15: date=2023-04-28 time=16:31:38 eventtime=1682724697936494139 tz="-0700"
logid="0100046520" type="event" subtype="system" level="notice" vd="root" logdesc="LTE
modem active SIM card switched: link monitor probe failure detected" msg="LTE modem
active SIM card slot changed to 2, due to link monitor probe failures."

19: date=2023-04-28 time=16:31:13 eventtime=1682724673152506599 tz="-0700"
logid="0100022932" type="event" subtype="system" level="warning" vd="root" logdesc="Link
```

```
monitor status warning" name="modem" interface="wwan" probeproto="ping" msg="Link
Monitor changed state from alive to dead, protocol: ping."
```

LAG interface status signaled to peer when available links fall below min-link

This information is also available in the FortiOS 7.4 Administration Guide:
- LAG interface status signaled to peer device

FortiGate can signal LAG (link aggregate group) interface status to the peer device. If the number of available links in the LAG on the FortiGate falls below the configured minimum number of links (`min-links`), the LAG interface goes down on both the FortiGate and the peer device.

When the minimum number of links is satisfied again, the LAG interface automatically resumes operation on both the FortiGate and the peer device. While the LAG interface is down, interface members are in the Link Aggregation Control Protocol (LACP) MUX state of *Waiting*.

Example

In this example, the LAG interface is configured on FGT_A and peered with FGT_B.

To verify the configuration:

1. On FGT_A, check the minimum number of links for the LAG interface named test_agg1.

   In the following example, `set min-links 1` indicates that a minimum of one alive interface member is required to keep the LAG interface up.

   ```
   # show
   config system interface
       edit "test_agg1"
           set vdom "vdom1"
           set ip 11.1.1.1 255.255.255.0
           set allowaccess ping https
           set type aggregate
           set member "port7" "port8" "port9"
           set device-identification enable
           set lldp-transmission enable
           set role lan
           set snmp-index 41
           set min-links 1
       next
   end
   ```

2. Change the status of port9 to down.

   ```
   Config system interface
      edit port9
         set status down
   end
   ```

3. On FGT_A, test the LAG interface named test_agg1.

The status is up for test_agg1 interface because two interface members (port7 and port8) are up, and only one interface member (port9) is down.

```
# diagnose netlink aggregate name test_agg1
LACP flags: (A|P)(S|F)(A|I)(I|O)(E|D)(E|D)
(A|P) - LACP mode is Active or Passive
(S|F) - LACP speed is Slow or Fast
(A|I) - Aggregatable or Individual
(I|O) - Port In sync or Out of sync
(E|D) - Frame collection is Enabled or Disabled
(E|D) - Frame distribution is Enabled or Disabled

status: up
npu: y
flush: n
asic helper: y
oid: 72
ports: 3
link-up-delay: 50ms
min-links: 1
ha: master
distribution algorithm: L4
LACP mode: active
LACP speed: slow
LACP HA: enable
aggregator ID: 1
actor key: 17
actor MAC address: d4:76:a0:01:e0:44
partner key: 17
partner MAC address: d4:76:a0:01:e8:1e

member: port7
  index: 0
  link status: up
  link failure count: 1
  permanent MAC addr: d4:76:a0:01:e0:44
  LACP state: established
  actor state: ASAIEE
  actor port number/key/priority: 1 17 255
  partner state: ASAIEE
  partner port number/key/priority: 1 17 255
  partner system: 1 d4:76:a0:01:e8:1e
  aggregator ID: 1
  speed/duplex: 1000 1
  RX state: CURRENT 6
  MUX state: COLLECTING_DISTRIBUTING 4

member: port8
  index: 1
  link status: up
  link failure count: 2
  permanent MAC addr: d4:76:a0:01:e0:45
  LACP state: established
  actor state: ASAIEE
  actor port number/key/priority: 2 17 255
  partner state: ASAIEE
```

```
  partner port number/key/priority: 2 17 255
  partner system: 1 d4:76:a0:01:e8:1e
  aggregator ID: 1
  speed/duplex: 1000 1
  RX state: CURRENT 6
  MUX state: COLLECTING_DISTRIBUTING 4

member: port9
  index: 2
  link status: down
  link failure count: 0
  permanent MAC addr: d4:76:a0:01:e0:46
```

4. On FGT_A, change the minimum number of links to 3.

```
config system interface
    edit "test_agg1"
        set vdom "vdom1"
        set ip 11.1.1.1 255.255.255.0
        set allowaccess ping https
        set type aggregate
        set member "port7" "port8" "port9"
        set device-identification enable
        set lldp-transmission enable
        set role lan
        set snmp-index 41
        set min-links 3
    next
end
```

5. On FGT_A, check the LAG interface named test_agg1:

   The status is down for test_agg1 interface because only two of the three required interface members are up. Interface members port7 and port8 are up, but interface member port9 is down.

```
# diagnose netlink aggregate name agg1
LACP flags: (A|P)(S|F)(A|I)(I|O)(E|D)(E|D)
(A|P) - LACP mode is Active or Passive
(S|F) - LACP speed is Slow or Fast
(A|I) - Aggregatable or Individual
(I|O) - Port In sync or Out of sync
(E|D) - Frame collection is Enabled or Disabled
(E|D) - Frame distribution is Enabled or Disabled

status: down
npu: y
flush: n
asic helper: y
oid: 230
ports: 3
link-up-delay: 50ms
min-links: 3
ha: master
distribution algorithm: L4
LACP mode: active
LACP speed: slow
LACP HA: enable
aggregator ID: 1
```

```
actor key: 17
actor MAC address: e8:1c:ba:b3:d0:df
partner key: 17
partner MAC address: e8:1c:ba:df:a0:ba

member: port7
  index: 0
  link status: up
  link failure count: 1
  permanent MAC addr: e8:1c:ba:b3:d0:df
  LACP state: negotiating
  actor state: ASAODD
  actor port number/key/priority: 1 17 255
  partner state: ASAIDD
  partner port number/key/priority: 1 17 255
  partner system: 61440 e8:1c:ba:df:a0:ba
  aggregator ID: 1
  speed/duplex: 1000 1
  RX state: CURRENT 6
  MUX state: WAITING 2

member: port8
  index: 1
  link status: up
  link failure count: 1
  permanent MAC addr: e8:1c:ba:b3:d0:e0
  LACP state: negotiating
  actor state: ASAODD
  actor port number/key/priority: 2 17 255
  partner state: ASAIDD
  partner port number/key/priority: 65 17 255
  partner system: 61440 e8:1c:ba:df:a0:ba
  aggregator ID: 1
  speed/duplex: 1000 1
  RX state: CURRENT 6
  MUX state: WAITING 2

member: port9
  index: 2
  link status: down
  link failure count: 0
  permanent MAC addr: e8:1c:ba:b3:d0:ed
```

6. On the peer FortiGate (FGT_B), check the LAG interface status.

   The status is down for test_agg2 interface due to FortiGate's ability to signal LAG interface status to the peer device. While interface members port7 and port8 are up, interface member port9 is down.

```
# diagnose netlink aggregate name test-agg2
LACP flags: (A|P)(S|F)(A|I)(I|O)(E|D)(E|D)
(A|P) - LACP mode is Active or Passive
(S|F) - LACP speed is Slow or Fast
(A|I) - Aggregatable or Individual
(I|O) - Port In sync or Out of sync
(E|D) - Frame collection is Enabled or Disabled
(E|D) - Frame distribution is Enabled or Disabled

status: down
```

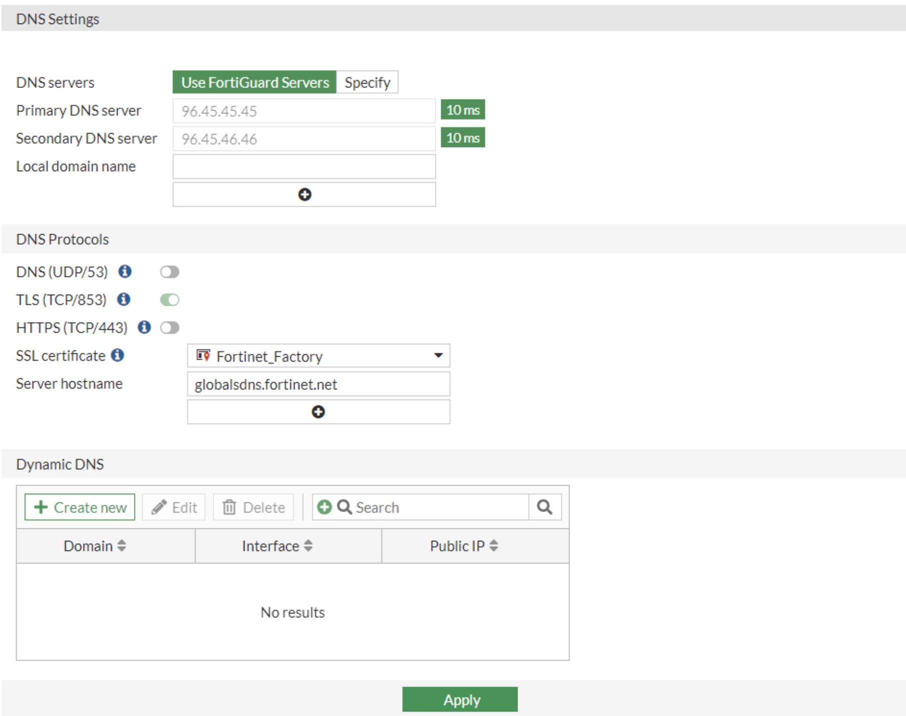
```
npu: y
flush: n
asic helper: y
oid: 72
ports: 3
link-up-delay: 50ms
min-links: 1
ha: master
distribution algorithm: L4
LACP mode: active
LACP speed: slow
LACP HA: enable
aggregator ID: 1
actor key: 17
actor MAC address: d4:76:a0:01:e8:1e
partner key: 17
partner MAC address: d4:76:a0:01:e0:44

member: port7
  index: 0
  link status: up
  link failure count: 1
  permanent MAC addr: d4:76:a0:01:e8:1e
  LACP state: negotiating
  actor state: ASAIDD
  actor port number/key/priority: 1 17 255
  partner state: ASAODD
  partner port number/key/priority: 1 17 255
  partner system: 44237 d4:76:a0:01:e0:44
  aggregator ID: 1
  speed/duplex: 1000 1
  RX state: CURRENT 6
  MUX state: ATTACHED 3

member: port8
  index: 1
  link status: up
  link failure count: 1
  permanent MAC addr: d4:76:a0:01:e8:1f
  LACP state: negotiating
  actor state: ASAIDD
  actor port number/key/priority: 2 17 255
  partner state: ASAODD
  partner port number/key/priority: 2 17 255
  partner system: 44237 d4:76:a0:01:e0:44
  aggregator ID: 1
  speed/duplex: 1000 1
  RX state: CURRENT 6
  MUX state: ATTACHED 3

member: port9
  index: 2
  link status: down
  link failure count: 0
  permanent MAC addr: d4:76:a0:01:e8:20
```

Configuring multiple DDNS entries in the GUI

Multiple DDNS interfaces can be configured in the FortiOS GUI. The visibility of DDNS entries in the GUI is no longer tied to the requirement of using the FortiGuard DNS server. The number of DDNS entries that can be configured is restricted by table size, with limits of 16, 32, and 64 entries for entry-level, mid-range, and high-end FortiGate models respectively.

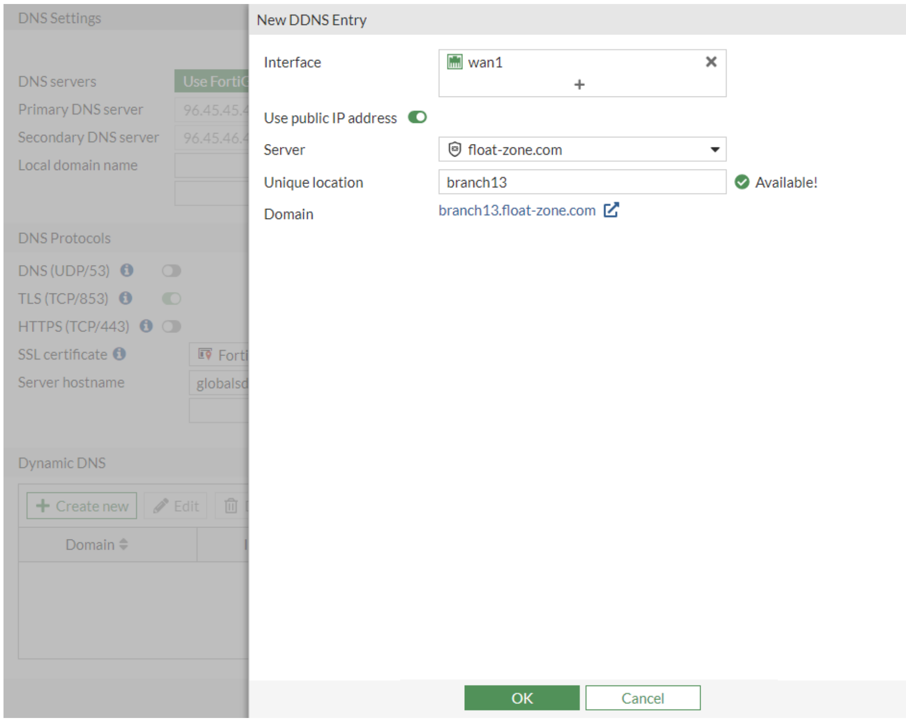
To configure multiple DDNS entries:

1. Go to *Network > DNS*.
2. In the *Dynamic DNS* table, click *Create new*.



The *New DDNS Entry* pane opens.

3. Configure the DDNS entry settings:
   a. Select the *Interface* with the dynamic connection.
   b. Select the *Server* that you have an account with.
   c. Enter the *Unique Location*.



   d. Click *OK*.
4. Click *Create new* and repeat step 3 to add more entries.
5. Click *Apply*.

Support DHCP client mode for inter-VDOM links - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:
- DHCP client mode for inter-VDOM links

The inter-VDOM link is capable of acquiring an IP address from the DHCP server, which allows for more seamless network integration.

Example

The following example is based on the configuration in FortiGate LAN extension, and assumes that the FortiGate connector, FortiGate access controller, interfaces, VDOMs, DHCP server, and firewall policies have already been configured.

In this example, the lan-ext VDOM was created on the FortiGate connector and is a `lan-extension` type. This configuration allows the VDOM to function as a FortiExtender in LAN extension mode. However, this configuration results in the loss of FortiGate security features on that VDOM. For users who wish to use the FortiGate security features locally on the FortiGate connector, another VDOM (such as the root VDOM) can be used. Once the DHCP server is enabled on the FortiGate controller, an inter-VDOM link belonging to another VDOM (in this case, the root VDOM) can receive an IP address by DHCP from the FortiGate controller.

To configure the inter-VDOM link:

1. Add the VDOM link with an Ethernet type:

```
config system vdom-link
    edit "lan_ext"
        set type ethernet
    next
end
```

2. Configure the VDOM link interfaces:

```
config system interface
    edit "lan_ext0"
        set vdom "lan-ext"
        set role lan
    next
    edit "lan_ext1"
        set vdom "root"
        set mode dhcp
    next
end
```

3. Verify that the lan_ext1 interface obtained an IP address from FortiGate access controller:

```
# diagnose ip address list | grep lan_ext1
IP=9.9.9.100->9.9.9.100/255.255.255.0 index=27 devname=lan_ext1
```

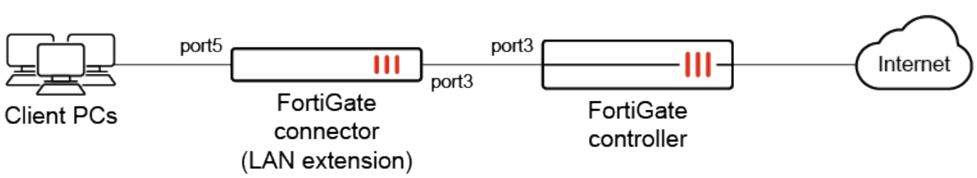
Configuring FortiGate LAN extension the GUI - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:
- Example GUI configuration

The FortiOS GUI supports configuring the FortiGate controller and connector for the FortiGate LAN extension feature.

Example

In this example, an FG-301E is the FortiGate controller, and CAPWAP access is allowed on port3. An FG-201F is the FortiGate connector with WAN port3 connected to the FortiGate controller, and LAN port5 is connected to the client PCs.



To configure the FortiGate LAN extension:

1. On the FortiGate controller, enable the FortiExtender setting. For high-end models (1000 series and higher) and VM models, enter:

```
config system global
    set fortiextender enable
end
```

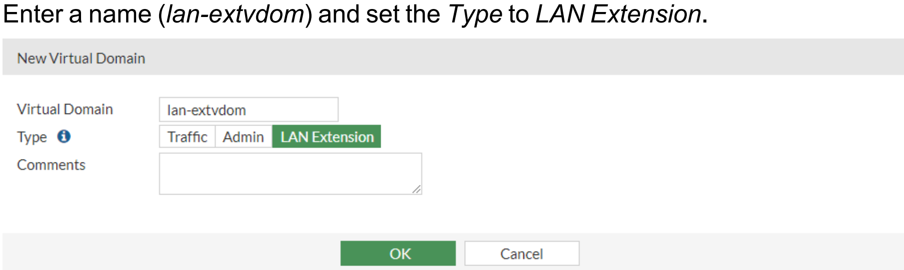
   This command is configured by default on entry-level and mid-range models (900 series and lower).

2. On the FortiGate controller, configure the port3 settings:
   a. Go to *Network > Interfaces* and edit *port3*.
   b. Set the *Addressing mode* to *IPAM*.
   c. In this example, IPAM is not enabled yet. Click *Enable IPAM*. The *IPAM Settings* pane opens.
   d. Set the *Status* to *Enabled*, enable *FortiExtender LAN extensions*, then click *OK*.
   e. In the *Administrative Access > IPv4* section, select *Security Fabric Connection* to enable CAPWAP on the interface.
   f. Enable *DHCP Server*.
   g. Click *OK*.
3. On the FortiGate connector, enable VDOMs:
   a. Go to *System > Settings*.
   b. In the *System Operation Settings* sections, enable *Virtual Domains*.
   c. Click *OK*. You will be logged out of the device when VDOM mode is enabled.
4. On the FortiGate connector, enable the FortiExtender setting. For high-end models (1000 series and higher) and VM models, enter:

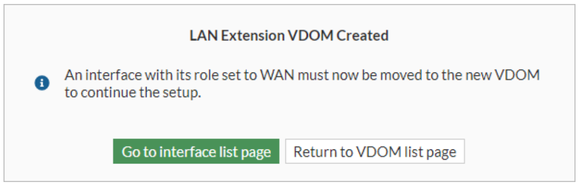
```
config system global
    set fortiextender enable
end
```

   This command is configured by default on entry-level and mid-range models (900 series and lower).

5. On the FortiGate connector, configure the LAN extension VDOM:
   a. Go to *System > VDOM* and click *Create New*.
   b. Enter a name (*lan-extvdom*) and set the *Type* to *LAN Extension*.

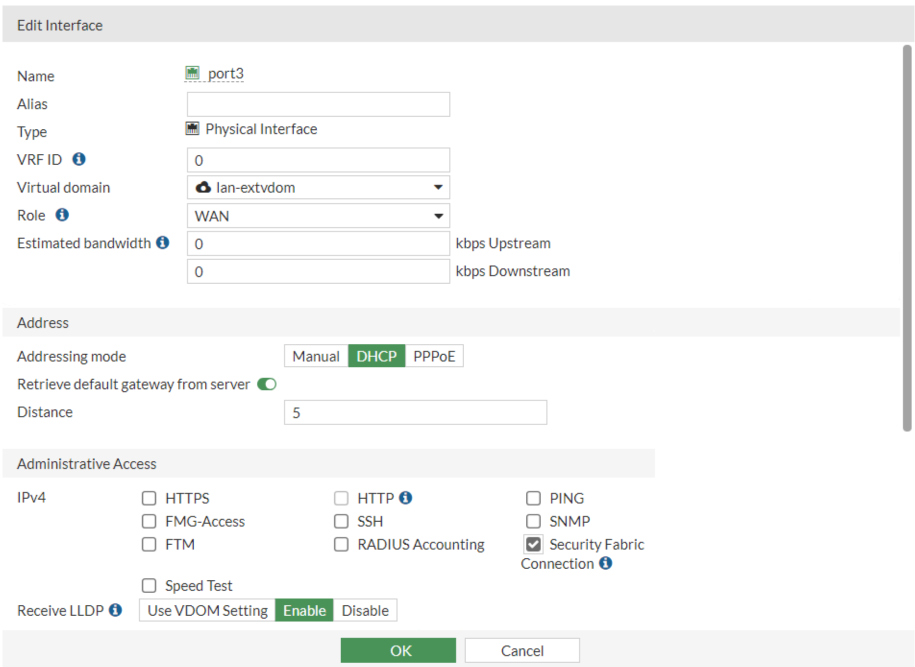


   c. Click *OK*. The *LAN Extension VDOM Created* prompt appears.



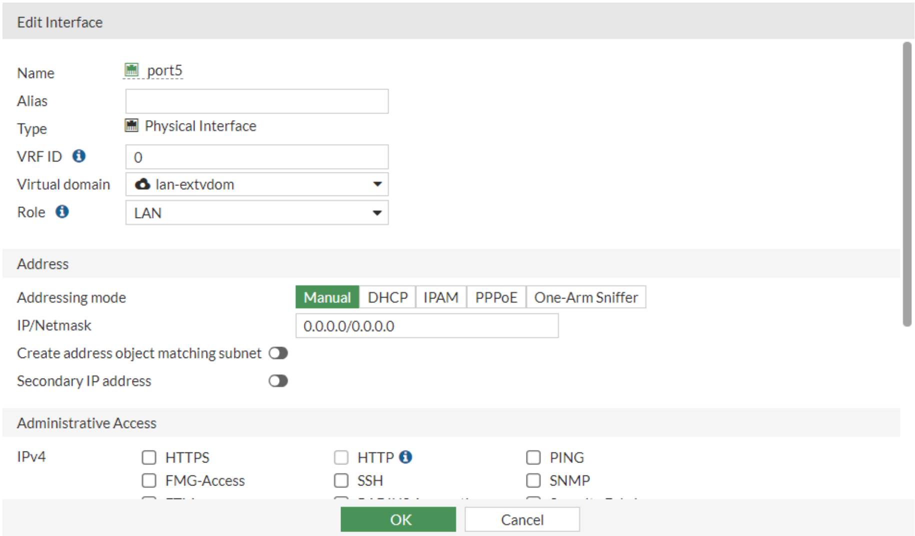
d. Click *Go to interface list page* to assign a role (LAN or WAN) and the LAN extension VDOM.

6. On the FortiGate connector, edit port3:
   a. Set the *Role* to *WAN*.
   b. Set the *Virtual domain* to *lan-extvdom*.



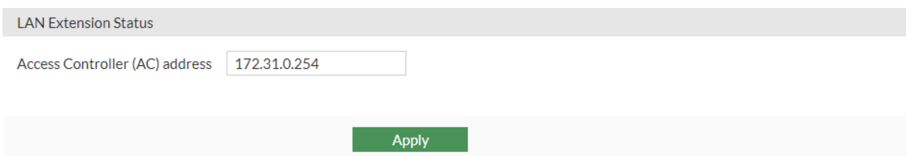
   c. Click *OK*.

7. On the FortiGate connector, edit port5:
   a. Set the *Role* to *LAN*.
   b. Set the *Virtual domain* to *lan-extvdom*.



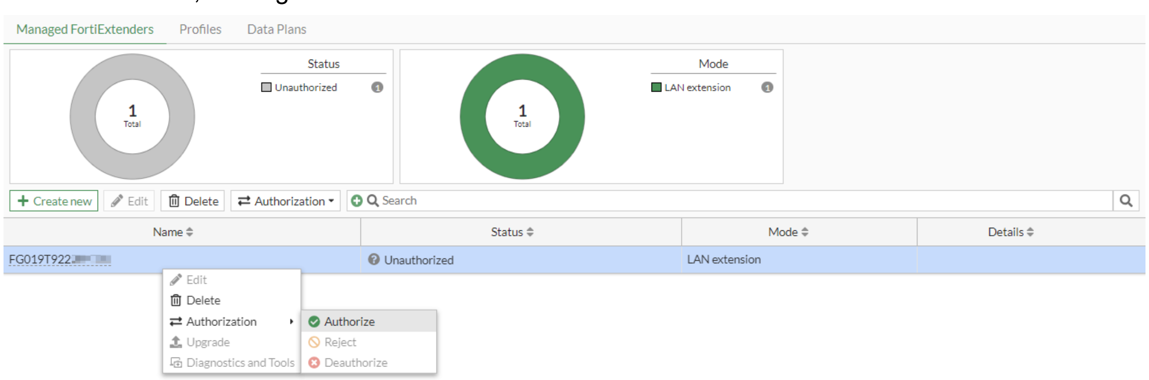
c. Click *OK*.

8. On the FortiGate connector, select the LAN extension VDOM, and enter the IP address of the FortiGate controller:
   a. Go to *Network > LAN Extension*.
   b. Set the *Access Controller (AC) address* to *172.31.0.254*.

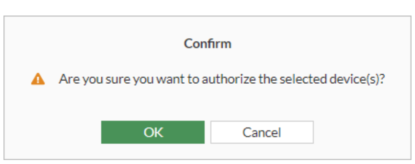


   c. Click *Apply*.

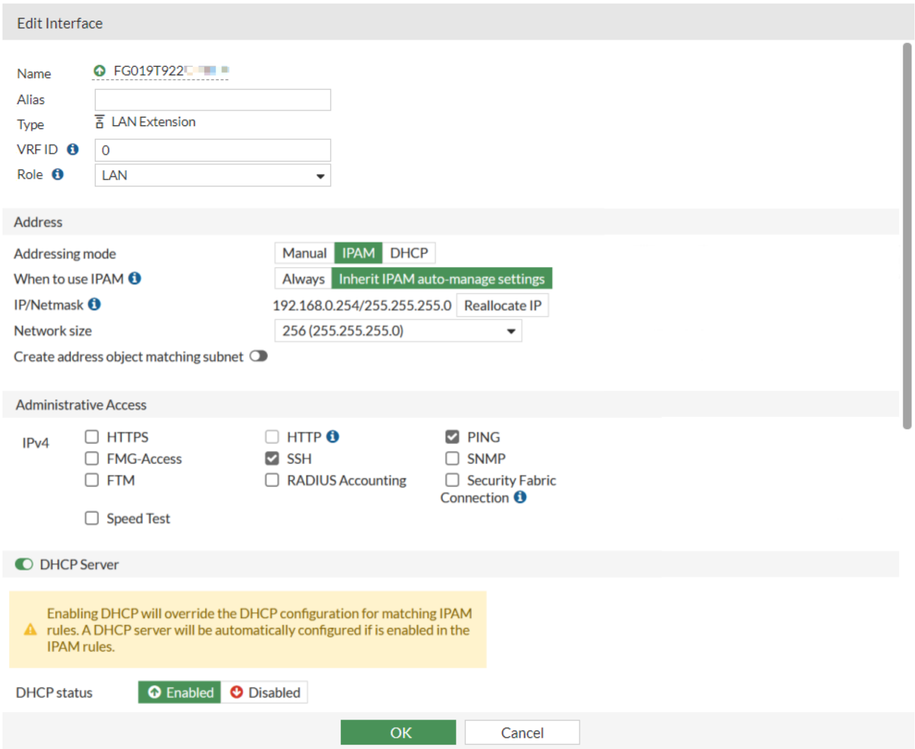
9. On the FortiGate controller, enable the FortiExtender feature visibility in the GUI, and authorize the FortiGate connector:
   a. Go to *System > Feature Visibility*. In the *Additional Features* section, enable *FortiExtender* and click *Apply*.
   b. Go to *Network > FortiExtenders* and select the *Managed FortiExtenders* tab.
   c. Select the device, then right-click and select *Authorization > Authorize*.



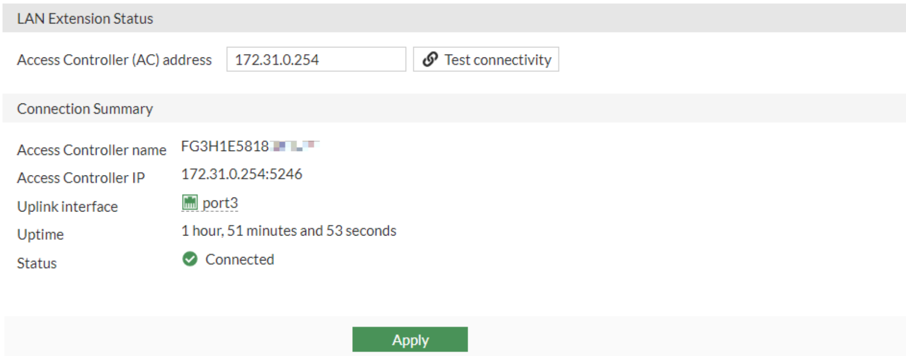
   d. Click *OK* to authorize the device.



10. On the FortiGate controller, configure the LAN extension interface:
    a. Go to *Network > Interfaces* and edit the LAN extension interface.
    b. Set the *Addressing mode* to *IPAM* and set *When to use IPAM* to *Inherit IPAM auto-manage settings* (default).
    c. Enable *DHCP Server*, and configure the settings as needed (see DHCP servers and relays for more information).



    d. Click *OK*.
11. On the FortiGate controller, configure the default gateway:
    a. Go to *Network > Static Routes* and edit the default gateway settings to specify the correct internet gateway address and WAN interface.
    b. Set the *Gateway Address* to *172.16.200.254*.
    c. Set the *Interface* to *mgmt*.
    d. Click *OK*.
12. On the FortiGate controller, configure the firewall policy to allow traffic to pass:
    a. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
    b. Set the *Incoming Interface* to the LAN extension interface.
    c. Configure the other settings as needed.
    d. Click *OK*.
13. On the FortiGate connector, verify that the LAN extension is connected:
    a. Go to *Network > LAN Extension*.
    b. Verify that the *Status* is *Connected*.



Transparent conditional DNS forwarder - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Transparent conditional DNS forwarder

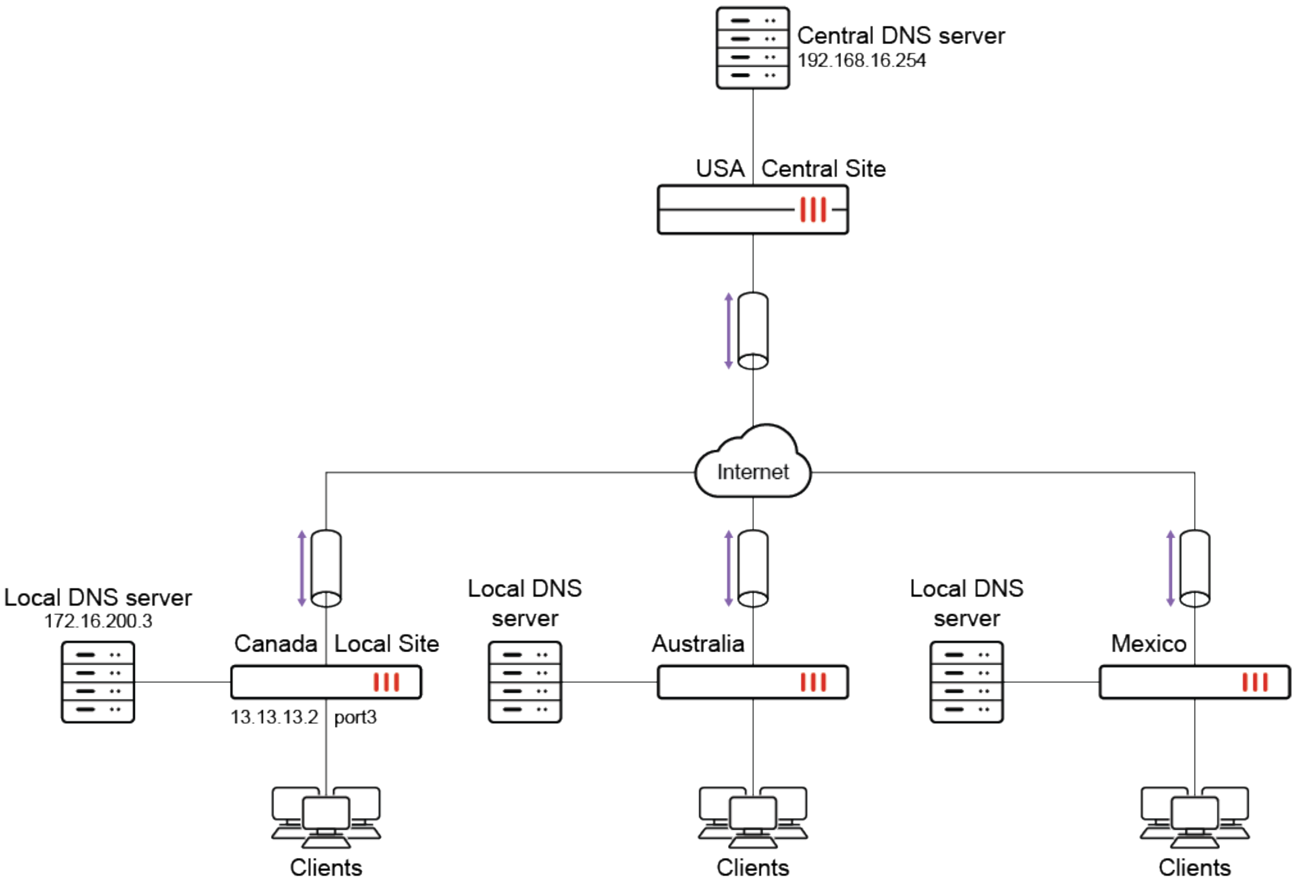
The transparent conditional DNS forwarder allows the FortiGate to intercept and reroute DNS queries for specific domains to a specific DNS server. For example, when a client’s DNS is located in a distant location, in order to resolve destination addresses (such as SaaS applications) to the closest application server, the FortiGate can intercept and reroute the requests to a local DNS to resolve.

This is done by parsing entries and creating a list of filters based on the domain names of zones. When a DNS request matches one of these filters, the DNS proxy will retrieve the zone's data. The DNS request will then be handled based on the zone's forwarder settings and whether a local answer is available. It may be forwarded to the original destination address, the forwarder address, or not forwarded at all if a local answer is available.

This provides greater control over DNS requests, especially when the administrator is not managing the DNS server configuration of the client devices. This can improve network efficiency and performance by resolving IPs local to the client's PCs rather than IPs local to the central DNS server.

Example

In this example, FortiGates at various locations are connected to a central site by VPN tunnels where the corporate DNS server is located. Typically, DNS queries from different sites are sent to the central DNS server and resolved to an IP local to the central site, which might cause latency and performance issues for certain destinations, such as SaaS applications.



The Local Site FortiGate is configured with the Microsoft domain and a local DNS entry. Traffic matching the Microsoft domain is either forwarded to the local DNS server or resolved by the FortiGate, which resolves it to an IP local to the Local Site, thus improving performance.

This example assumes the following have been configured:

- A successfully operational site-to-site VPN between the Local Site and the Central Site FortiGates (see Site-to-site VPN for more information).
- Appropriate routing and network interfaces.
- The client PCs are configured to use the Central DNS Server.

The transparent conditional DNS forwarder feature only works with a proxy-based firewall policy.

By default, DNS server options are not available in the GUI.

To enable DNS server options in the GUI:

1. Go to *System > Feature Visibility*.
2. In the *Additional Features* section, enable *DNS Database*.
3. Click *Apply*.

To configure the DNS zone and local DNS entries on the Local Site FortiGate in the GUI:

1. Go to *Network > DNS Servers*.
2. In the *DNS Database* table, click *Create New*.
3. Enter a *DNS Zone* name (*SaaS_applications*).
4. Enter a *Domain Name* (*microsoft.com*).
5. Disable the *Authoritative* setting.
6. In the *DNS Forwarder* field, click the + and enter the DNS Forwarder address (*172.16.200.3*).
7. Configure the DNS entry:
   a. In the *DNS Entries* table, click *Create New*.
   b. Set the *Type* to *Address (A)*.
   c. Enter a *Hostname* (*office*).
   d. Configure the remaining settings as needed. The options vary depending on the selected *Type*.
   e. Click *OK*.
   f. Optionally, add more DNS entries if needed.
8. In the CLI, configure the source IP:

```
config system dns-database
    edit "SaaS_applications"
        set source-ip 13.13.13.2
    next
end
```

If the DNS server is accessed over a VPN, it may be necessary to specify a source IP for the FortiGate to reach the DNS server. See How to let the FortiGate access internal DNS through site-to-site IPsec VPN for more information.

Site-to-site VPN is not a mandatory requirement for this feature to work and is only applicable to this example.

To configure the DNS zone and local DNS entries on the Local Site FortiGate in the CLI:

```
config system dns-database
    edit "SaaS_applications"
        set domain "microsoft.com"
        set authoritative disable
        set forwarder "172.16.200.3"
        set source-ip 13.13.13.2
        config dns-entry
            edit 1
                set hostname "office"
                set ip 172.16.200.55
            next
        end
```

```
    next
end
```

To add the DNS database to a DNS filter profile:

```
config dnsfilter profile
    edit "SaaS"
        set transparent-dns-database "SaaS_applications"
    next
end
```

Multiple DNS databases can be selected for `transparent-dns-database`.

After selecting a DNS database, users are not permitted to modify the domain name of the zone. Before making any changes to the domain name, remove the reference from the `dnsfilter profile`.

To apply the DNS filter profile in a firewall policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and edit the outbound policy towards the IPsec VPN tunnel.
2. Set the *Inspection Mode* to *Proxy-based*.
3. In the *Security Profiles* section, enable *DNS Filter* and select the profile created in the previous procedure (*SaaS*).
4. In the *Logging Options* section, enable *Log Allowed Traffic*.
5. Configure the remaining settings as needed.
6. Click *OK*.

To apply the DNS filter profile to the outbound policy towards the IPsec VPN tunnel in the CLI:

```
config firewall policy
    edit 1
        set name "outbound_VPN"
        ...
        set inspection-mode proxy
        set dnsfilter-profile "SaaS"
        set logtraffic enable
        ...
    next
end
```

To verify the configuration:

From one of the Windows client desktops, use the `nslookup` command to send various DNS queries.

1. Send a DNS query for a DNS entry configured locally on the Local Site FortiGate:

   ```
   C:\Users\demo>nslookup office.microsoft.com
   Server:  Unknown
   Address:  192.168.16.254
   Non-authoritative answer:
   Name:     osiprod-wus-pineapple-100.westus.cloudapp.azure.com
   Address:  172.16.200.55
   ```

   The query is resolved to the IP address configured on the Local Site FortiGate.
2. Send a DNS query for the domain configured on the Local Site FortiGate:

```
C:\Users\demo>nslookup teams.microsoft.com
Server:  Unknown
Address:  192.168.16.254
Non-authoritative answer:
Name:     s-0005.s-msedge.net
Address:  172.16.200.254
```

The query is resolved by the local DNS server.

3. Send a DNS query for a domain that is not configured on the Local Site FortiGate:

```
C:\Users\demo>nslookup facebook.com
Server:  Unknown
Address:  192.168.16.254
Non-authoritative answer:
Name:    facebook.com
Addresses:  157.240.249.35
```

The query is resolved by the central DNS server.

IPv6 support for conditional DNS forwarder

The configuration for IPv6 is similar to an IPv4 conditional DNS forwarder. When configuring the DNS forwarder address, the IPv6 address must be specified.

To configure a DNS forwarder:

```
config system dns-database
    edit <name>
        set source-ip6 <IPv6_address>
        set forwarder6 <IPv6_address>
    next
end
```

If the DNS server is accessed over a VPN, it may be necessary to specify a source IP for the FortiGate to reach the DNS server. See How to let the FortiGate access internal DNS through site-to-site IPsec VPN for more information.

IPAM enhancements - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Configure IPAM locally on the FortiGate

Interfaces with a LAN role, wireless network interfaces (`vap-switch` type), and FortiExtender LAN extension interfaces (`lan-extension` type) can now receive an IP address from an IPAM server without any additional configuration at the interface level. IPAM also detects and resolves any IP conflicts that may occur on the interfaces that it manages. See Interfaces in the FortiOS Administration Guide for more information.

This enables easier administration for widely used interfaces in the network and reduces complexity, which usually arise when there are a large number of interfaces to be managed in the network. By using IPAM, network administrators can easily keep track of the various interfaces in their network and ensure that they are properly configured and functioning as intended. This can save time and effort, and helps prevent issues that may arise from misconfigured or improperly managed interfaces.

To configure IPAM in the GUI:

1. Go to *Network > IPAM* and select the *IPAM Settings* tab.
2. Configure the following settings:

Status	Enable/disable integration with IP address management services (IPAM).
Auto-resolve conflicts	Enable/disable automatic conflict resolution.
Interfaces with LAN role	Enable/disable LAN interface address management by default.
FortiAP SSIDs	Enable/disable FortiAP SSID address management by default.
FortiExtender LAN extensions	Enable/disable FortiExtender LAN extension interface address management by default.

3. Click *OK*.

To configure IPAM in the CLI:

```
config system ipam
    set status {enable | disable}
    set automatic-conflict-resolution {enable | disable}
    set manage-lan-addresses {enable | disable}
    set manage-lan-extension-addresses {enable | disable}
    set manage-ssid-addresses {enable | disable}
end
```

When `automatic-conflict-resolution` is enabled, IPAM will periodically check and validate the addresses of all interfaces. In case of any conflicts, IPAM will automatically attempt to obtain a new address for the affected interface managed by IPAM, ensuring no address duplication.

When a `manage-` option is enabled, any interface that meets the specified criteria will automatically receive an IP address from IPAM. However, if this option is disabled, interfaces that meet the criteria will not be configured by IPAM. All `manage-` options are disabled by default. The central FortiIPAM configuration can be overridden at the interface level.

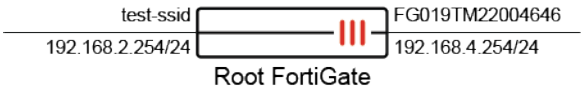
To override the central FortiIPAM configuration at the interface level:

```
config system interface
    edit <name>
        set ip-managed-by-fortiipam {enable | disable | inherit-global}
    next
end
```

The default setting is to inherit from the global configuration (`inherit-global`) through the relevant `manage-` option under `config system ipam`.

Example

In this example, the FortiGate serves as the Security Fabric root and has two interfaces: test-ssid (`vap-switch` type) and FG019TM22004646 (`lan-extension` type). Currently, neither interface has an IP address assigned to it.



To configure IPAM on the root FortiGate:

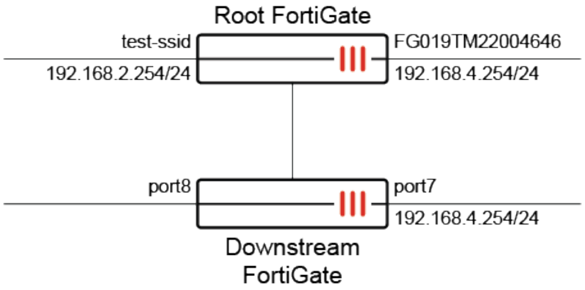
1. Go to *Network > IPAM* and select the *IPAM Settings* tab.
2. Enable the *Status*, *Auto-resolve conflicts*, *Interfaces with LAN role*, *FortiAP SSIDs*, and *FortiExtender LAN extensions* settings.

   IPAM is disabled by default, so all these options are disabled by default. Each option must be activated individually to function, and they do not depend on one another.

3. Click *OK*.
   After enabling IPAM on the root FortiGate with the specified settings, FortiGates that are part of the Security Fabric and have an interface set to either the LAN role, `vap-switch` type, or `lan-extension` type will automatically receive an IP assignment from the IPAM server without requiring any additional configuration at the interface level.
4. Verify the list of IPAM entries:

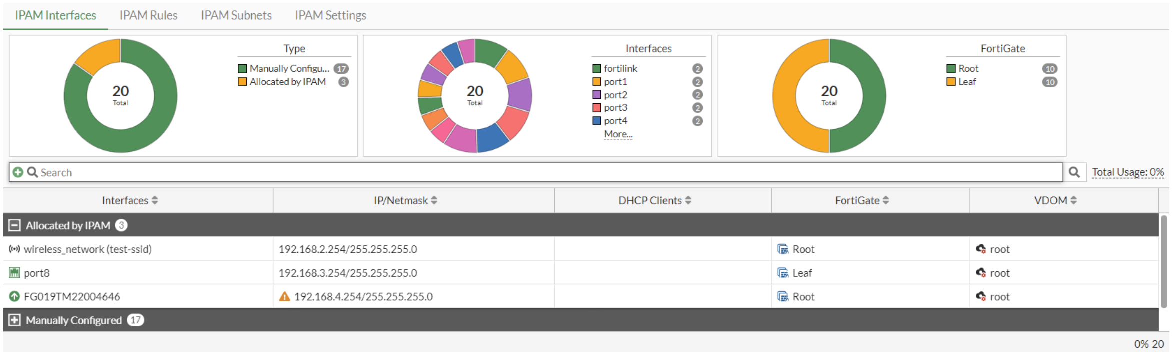
```
# diagnose sys ipam list entries
Entries: (sn, vdom, interface, subnet/mask, conflict)

IPAM Entries:
  FGVM08TM22004645 root FG019TM22004646 192.168.4.254/24
  FGVM08TM22004645 root test-ssid 192.168.2.254/24
```

When a downstream FortiGate joins the Security Fabric, the port7 interface is configured with a static IP (192.168.4.254/24), and port8 is set to a LAN role with no IP address assigned. The IPAM server assigns an IP to port8 of the downstream FortiGate since its role was set to LAN. It is observed that the FG019TM22004646 interface of the root FortiGate conflicts with port7 of the downstream FortiGate.



To verify the IP address conflict resolution:

1. On the root FortiGate, go to *Network > IPAM* and select the *IPAM Interfaces* tab.

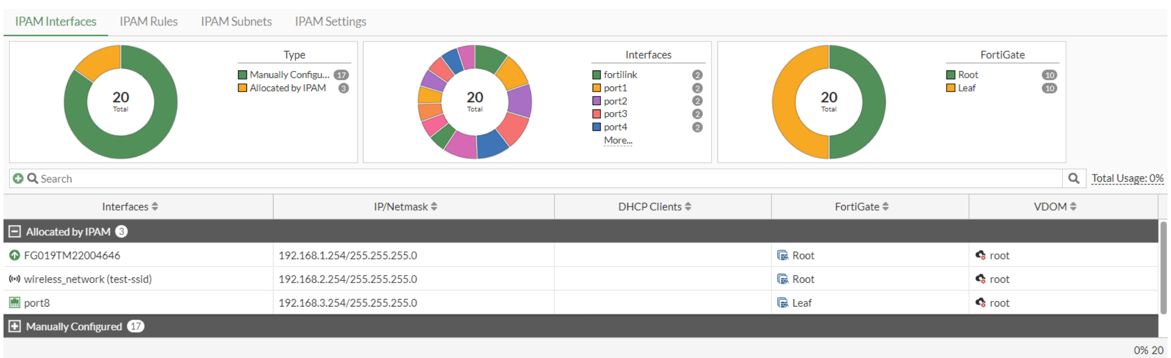


There is a conflict marker (warning icon) beside the IP address of *FG019TM22004646* due to a conflict between the IPAM-assigned interface FG019TM22004646 of the root FortiGate and the manually configured interface of the downstream FortiGate.

   a. Verify the list of IPAM entries in the CLI:

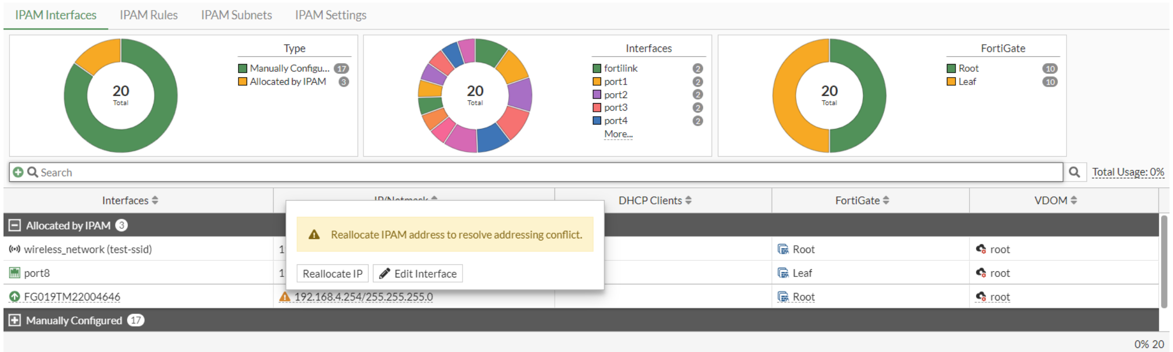
```
# diagnose sys ipam list entries
Entries: (sn, vdom, interface, subnet/mask, conflict)

  IPAM Entries:
  FGVM08TM22004645 root test-ssid 192.168.2.254/24
  FGVM08TM22004647 root port8 192.168.3.254/24
  FGVM08TM22004645 root FG019TM22004646 192.168.4.254/24 C
```

2. After some time, since *Auto-resolve conflicts* is enabled in the IPAM settings, the conflict is resolved automatically.



FG019TM22004646 has been assigned a new IP address of 192.168.1.254/24.

If *Auto-resolve conflicts* is disabled in the IPAM settings, mouse over the conflict marker and select *Reallocate IP* to manually reallocate the IP address.



a. Verify the list of IPAM entries in the CLI:

```
# diagnose sys ipam list entries
Entries: (sn, vdom, interface, subnet/mask, conflict)

  IPAM Entries:
  FGVM08TM22004645 root FG019TM22004646 192.168.1.254/24
  FGVM08TM22004645 root test-ssid 192.168.2.254/24
  FGVM08TM22004647 root port8 192.168.3.254/24
```

DNS over QUIC and DNS over HTTP3 for transparent and local-in DNS modes - 7.4.1

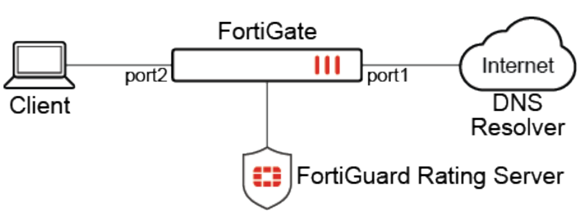
This information is also available in the FortiOS 7.4 Administration Guide:

- DNS over QUIC and DNS over HTTP3 for transparent and local-in DNS modes

DNS over QUIC (DoQ) and DNS over HTTP3 (DoH3) are supported in proxy mode inspection for transparent and local-in explicit modes. With DoQ and DoH3, connections can be established faster than with DNS over TLS (DoT) or DNS over HTTPS (DoH). The FortiGate can also handle the QUIC/TLS handshake and perform deep inspection for HTTP3 and QUIC traffic. This allows for faster and more secure DNS resolution, with improved privacy and reduced latency.

In transparent mode, the FortiGate is acting as a proxy, forwarding DNS queries, and not as a DNS server. In local-in DNS mode, the FortiGate acts as the DNS server and a DNS filter profile is applied in the system DNS server.

The firewall policy must be in proxy mode.



DoQ transparent and local-in query can be achieved using tools or applications in Linux, such as the q tiny command line DNS client from Natesales.

DoH3 transparent and local-in query can be achieved in Linux using q or Curl. In Windows, change the client network DNS server to the FortiGate and treat the FortiGate as a HTTP3 DNS server listening for DoH3 connections.

To configure DoQ in transparent mode:

1. Enable QUIC in the ssl-ssh-profile:

```
config firewall ssl-ssh-profile
    edit "protocols"
        config dot
            set status deep-inspection
            set quic enable
        end
    next
end
```

2. Configure a DNS filter profile:

```
config dnsfilter profile
    edit "dnsfilter_fgd"
        config ftgd-dns
            config filters
                edit 1
                    set category 30
                    set action block
                next
            end
        end
    next
end
```

3. Apply the profiles to a proxy firewall policy:

```
config firewall policy
    edit 1
        set name "dnsfilter"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "protocols"
        set dnsfilter-profile "dnsfilter_fgd"
        set logtraffic all
        set nat enable
    next
end
```

4. Test the configuration:
   On the client, use q to query a FortiGuard category30 domain with the Adguard DNS server over QUIC. The default redirect block IP address should be returned:

```
pc03:~# q www.sfu.ca @quic://dns.adguard.com --tls-no-verify
2023/08/18 18:53:44 failed to sufficiently increase receive buffer size (was: 208 kiB,
wanted: 2048 kiB, got: 416 kiB). See https://github.com/quic-go/quic-go/wiki/UDP-
Receive-Buffer-Size for details.
```

```
www.sfu.ca. 1m0s A 208.91.112.55
www.sfu.ca. 1m0s AAAA 2620:101:9000:53::55
```

To configure DoQ in local-in mode:

1. In the FortiGate DNS server configuration, enable DoQ for a port with the previously configured DNS filter profile applied:

```
config system dns-server
    edit "port2"
        set dnsfilter-profile "dnsfilter_fgd"
        set doq enable
    next
end
```

2. Test the configuration:

   On the client, use q to query a FortiGuard category30 domain with the FortiGate interface over QUIC. The default redirect block IP address should be returned:

```
pc03:~# q www.mcgill.ca @quic://10.1.100.150 --tls-no-verify
2023/08/18 20:05:53 failed to sufficiently increase receive buffer size (was: 208 kiB,
wanted: 2048 kiB, got: 416 kiB). See https://github.com/quic-go/quic-go/wiki/UDP-
Receive-Buffer-Size for details.
www.mcgill.ca. 1m0s A 208.91.112.55
www.mcgill.ca. 1m0s AAAA 2620:101:9000:53::55
```

To configure DoH3 in transparent mode:

1. Enable QUIC in the ssl-ssh-profile:

```
config firewall ssl-ssh-profile
    edit "protocols"
        config https
            set ports 443 8443
            set status deep-inspection
            set quic enable
        end
    next
end
```

2. Configure a DNS filter profile:

```
config dnsfilter profile
    edit "dnsfilter_fgd"
        config ftgd-dns
            config filters
                edit 1
                    set category 30
                    set action block
                next
            end
        end
    next
end
```

3. Apply the profiles to a proxy firewall policy:

```
config firewall policy
    edit 1
        set name "dnsfilter"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "protocols"
        set dnsfilter-profile "dnsfilter_fgd"
        set logtraffic all
        set nat enable
    next
end
```

4. Test the configuration:

   On the client with HTTP3 support, use q or Curl to query a FortiGuard category30 domain with the Adguard DNS server or Cloudflare DNS server over QUIC. The default redirect block IP address should be returned:

```
pc03:~# q www.mcgill.ca --http3 @https://dns.adguard.com --tls-no-verify
2023/08/18 21:04:02 failed to sufficiently increase receive buffer size (was: 208 kiB,
wanted: 2048 kiB, got: 416 kiB). See https://github.com/quic-go/quic-go/wiki/UDP-
Receive-Buffer-Size for details.
www.mcgill.ca. 1m0s A 208.91.112.55
www.mcgill.ca. 1m0s AAAA 2620:101:9000:53::55
```

```
pc03:~# curl -H 'accept: application/dns-message' -v -k --http3 'https://1.1.1.1/dns-
query?dns=q80BAAABAAAAAAAAA3d3dwN1YmMCY2EAAAEAAQ' | hexdump
*   Trying 1.1.1.1:443...
* Connect socket 5 over QUIC to 1.1.1.1:443
* Sent QUIC client Initial, ALPN: h3,h3-29,h3-28,h3-27
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0*
Connected to 1.1.1.1 (1.1.1.1) port 443 (#0)
* h3 [:method: GET]
* h3 [:path: /dns-query?dns=q80BAAABAAAAAAAAA3d3dwN1YmMCY2EAAAEAAQ]
* h3 [:scheme: https]
* h3 [:authority: 1.1.1.1]
* h3 [user-agent: curl/7.80.0-DEV]
* h3 [accept: application/dns-message]
* Using HTTP/3 Stream ID: 0 (easy handle 0x558fdd1c2220)
> GET /dns-query?dns=q80BAAABAAAAAAAAA3d3dwN1YmMCY2EAAAEAAQ HTTP/3
> Host: 1.1.1.1
> user-agent: curl/7.80.0-DEV
> accept: application/dns-message
>
< HTTP/3 200
< content-type: application/dns-message
< content-length: 44
<
```

```
{ [44 bytes data]
100    44  100    44    0     0   1305      0 --:--:-- --:--:-- --:--:--  1375
* Connection #0 to host 1.1.1.1 left intact
0000000 cdab 0081 0100 0100 0000 0000 7703 7777
0000010 7503 6362 6302 0061 0100 0100 0cc0 0100
0000020 0100 0000 3c00 0400 5bd0 3770
000002c
```

To configure DoH3 in local-in mode:

1. In the FortiGate DNS server configuration, enable DoH3 for a port with the previously configured DNS filter profile applied:

```
config system dns-server
    edit "port2"
        set dnsfilter-profile "dnsfilter_fgd"
        set doh3 enable
    next
end
```

2. Test the configuration:

   On the client with HTTP3 support, use q or Curl to query a FortiGuard category30 domain with the FortiGate interface over HTTP3. The default redirect block IP address should be returned:

```
pc03:~# q www.mcgill.ca --http3 @https://10.1.100.150 --tls-no-verify
2023/08/18 20:37:55 failed to sufficiently increase receive buffer size (was: 208 kiB,
wanted: 2048 kiB, got: 416 kiB). See https://github.com/quic-go/quic-go/wiki/UDP-
Receive-Buffer-Size for details.
www.mcgill.ca. 1m0s A 208.91.112.55
www.mcgill.ca. 1m0s AAAA 2620:101:9000:53::55
```

```
pc03:~# curl -H 'accept: application/dns-message' -v -k --http3
'https://10.1.100.150/dns-query?dns=q80BAAABAAAAAAAAA3d3dwN1YmMCY2EAAAEAAQ' | hexdump
*   Trying 10.1.100.150:443...
* Connect socket 5 over QUIC to 10.1.100.150:443
* Sent QUIC client Initial, ALPN: h3,h3-29,h3-28,h3-27
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0*
Connected to 10.1.100.150 (10.1.100.150) port 443 (#0)
* h3 [:method: GET]
* h3 [:path: /dns-query?dns=q80BAAABAAAAAAAAA3d3dwN1YmMCY2EAAAEAAQ]
* h3 [:scheme: https]
* h3 [:authority: 10.1.100.150]
* h3 [user-agent: curl/7.80.0-DEV]
* h3 [accept: application/dns-message]
* Using HTTP/3 Stream ID: 0 (easy handle 0x55ced8274250)
> GET /dns-query?dns=q80BAAABAAAAAAAAA3d3dwN1YmMCY2EAAAEAAQ HTTP/3
> Host: 10.1.100.150
> user-agent: curl/7.80.0-DEV
> accept: application/dns-message
>
< HTTP/3 200
< content-type: application/dns-message
< content-length: 44
<
{ [44 bytes data]
```

```
100    44  100    44    0     0   1893      0 --:--:-- --:--:-- --:--:--  2000
* Connection #0 to host 10.1.100.150 left intact
0000000 cdab 0081 0100 0100 0000 0000 7703 7777
0000010 7503 6362 6302 0061 0100 0100 0cc0 0100
0000020 0100 0000 3c00 0400 5bd0 3770
000002c
```

Interfaces in non-management VDOMs as the source IP address of the DNS conditional forwarding server - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Interfaces in non-management VDOMs as the source IP address of the DNS conditional forwarding server

Interfaces that are in non-management VDOMs can be the source IP address of the DNS conditional forwarding server.

- When `vdom-dns` is enabled in a VDOM, only the IP addresses of interfaces in that VDOM can be configured as the `source-ip`.
- When `vdom-dns` is disabled (default), only the IP address of interfaces in the management VDOM can be configured as the `source-ip`.

In this example:

- `vdom1` is a non-management VDOM
- `port8` is assigned to `vdom1` and has IP address 13.13.13.13
- `port1` is assigned to the management VDOM (root) and has IP address 172.16.200.1

To configure the interfaces:

```
config global
    config system interface
        edit "port8"
            set vdom "vdom1"
            set ip 13.13.13.13 255.255.255.0
        next
        edit "port1"
            set vdom "root"
            set ip 172.16.200.1 255.255.255.0
        next
    end
end
```

To test configuring a source IP address when `vdom-dns` is disabled:

```
config vdom
    edit vdom1
        config system vdom-dns
            set vdom-dns disable
        end
    next
end
```

- `port8` cannot be used as the source IP address in a DNS database because it is assigned to `vdom1`, and not to a management VDOM:

```
config vdom
    edit vdom1
        config system dns-database
            edit "1"
                set source-ip 13.13.13.13
13.13.13.13 does not match any interface ip in vdom root.
node_check_object fail! for source-ip 13.13.13.13
```

- `port1` can be used as the source IP address in a DNS database because it is assigned to the management VDOM:

```
config vdom
    edit vdom1
        config system dns-database
            edit "1"
                set source-ip 172.16.200.1
            next
        end
    next
end
```

To test configuring a source IP address when `vdom-dns` is enabled:

```
config vdom
    edit vdom1
        config system vdom-dns
            set vdom-dns enable
        end
    next
end
```

- `port8` can be used as the source IP address in a DNS database because it is assigned to the `vdom1`:

```
config vdom
    edit vdom1
        config system dns-database
            edit "1"
                set source-ip 13.13.13.13
            next
        end
    next
end
```

- `port1` cannot be used as the source IP address in a DNS database because it is assigned to the management VDOM, and not to `vdom1`:

```
config vdom
    edit vdom1
        config system dns-database
            edit "1"
                set source-ip 172.16.200.1
172.16.200.1 does not match any interface ip in vdom vdom1.
node_check_object fail! for source-ip 172.16.200.1
```

FortiGate 3G4G: improved dual SIM card switching capabilities - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Active SIM card switching

Real time switching from active to passive SIM cards is improved on FortiGates with a cellular modem and dual SIM cards. Now SIM cards can switch when LTE modem traffic exceeds a specified data plan limit for a specified billing period. The SIM card switches shortly after the data plan limit is exceeded.

The following commands and options can be used to configure this feature:

```
config system lte-modem
    set data-usage-tracking enable
    config sim-switch
        set by-data-plan enable
    end
    config data-plan
        edit <id>
            set target-sim-slot {SIM-slot-1 | SIM-slot-2}
            set data-limit <data plan limit in MB to trigger SIM card switch>
            set data-limit-alert <percentage 1 to 99 to trigger log entry>
            set billing-period {monthly | weekly | daily}
            set billing-date <1 to 31 when billing-period is monthly>
            set billing-weekday <Sunday to Saturday when billing-period is weekly>
            set billing-hour <0 to 23 when billing-period is daily>
            set overage disable
            set iccid <19 to 20 digits to specify ICCID of SIM card>
            set delay-switch-time <delay SIM card switch to a specified UTC time in format
HH:MM>
        next
    end
end
```

`data-usage-tracking {enable | disable}`	Enable tracking of data usage for the LTE modem: • `enable`: track data usage. • `disable`: do not track data usage. Must be enabled to configure SIM card switching based on data plan overage.
`by-data-plan {enable | disable}`	Enable switching of SIM cards on the LTE modem based on data plan limits: • `enable`: allow SIM card switching when `data-limit` is exceeded. • `disable`: do not switch SIM cards when `data-limit` is exceeded.
`target-sim-slot {sim-slot-1 | sim-slot-2}`	Specify which SIM slot to configure.
`data-limit <integer>`	Specify the data limit in MB for the SIM slot (0 - 100000, use 0 for unlimited data).
`data-limit-alert <integer>`	Specify at what percentage of used `data-limit` to trigger a log entry (1 to 99).
`billing-period {month | week | day}`	Specify the billing period.

`billing-date <integer>`	When `billing-period` is set to `monthly`, specify what day of the month the bill is issued (1 to 31).
`billing-weekday {sunday | monday | tuesday | wednesday | thursday | friday | saturday}`	When `billing-period` is set to `weekly` specify what day of the week the bill is issued.
`billing-hour <integer>`	When `billing-period` is set to `daily` specify what hour of the day the bill is issued (0 to 23).
`overage {enable | disable}`	Disable data usage from exceeding the configured data limit: • `enable`: allow data usage to exceed the amount specified in `data-limit`. • `disable`: do not allow data usage to exceed the amount specified in `data-limit`. When disabled, SIM cards are switched before the data limit is exceeded. Must be disabled to allow SIM card switching.
`iccid <string>`	Specify the Integrated Circuit Card Identification Number (ICCID) for the SIM card in 19 to 20 digits.
`delay-switch-time <integer:integer>`	Delay SIM card switch to a specified UTC time in format HH:MM.

Example

In this example, data tracking and SIM card switching by data plan are enabled for the LTE modem. Each SIM card for the LTE modem is configured with a data plan.

When traffic causes data usage to surpass the configured data limit for one SIM card, the LTE modem disconnects, and the wwan interface loses its IP address and gateway. The idle SIM card becomes active, as long as it has available data to be used. After the SIM card switch completes, the LTE modem reconnects, and the wwan interface gains its IP address and gateway again.

To configure SIM card switching by data plan overage:

1. Enable data tracking for the LTE modem:
```
config system lte-modem
    set data-usage-tracking enable
end
```
2. Enable SIM card switching by data plan for the LTE modem:
```
config system lte-modem
    config sim-switch
        set by-data-plan enable
    end
end
```
3. Configure a data plan for each SIM card on the LTE modem:
   In this example, SIM-slot-1 is configured with a data limit of 50 MB for a monthly bill issued on the 10th day of the month.
   SIM-slot-2 is configured is configured with a data limit of 60 MB for a monthly bill issued on the first day of the month.
   Data overage is disabled for both SIM card slots to allow the SIM cards to switch when the data limits are exceeded.

```
config system lte-modem
    config data-plan
        edit "1"
            set target-sim-slot SIM-slot-1
            set data-limit 50
            set billing-period monthly
            set overage disable
            set billing-date 10
        next
        edit "2"
            set target-sim-slot SIM-slot-2
            set data-limit 60
            set billing-period monthly
            set overage disable
            set billing-date 1
        next
    end
end
```

When the specified `data-limit` is exceeded while `overage` is disabled, the SIM card switch is triggered.

When `overage` is enabled, the specified `data-limit` can be exceeded, and a SIM card switch is not triggered.

Data usage is reset after the billing period passes.

4. Monitor data usage against the data limit:

```
# diagnose sys lte-modem data-usage
Estimated LTE Modem data usage in this billing cycle:
Active data plan:              1
Active SIM slot:               slot-1
Plan data limit:               60(MB)
Plan overage status:           disable
sim-switch.by-data-plan:       enable
Usage:                         67(MB)
Usage percentage:              111.67%
Current time:                  2023-07-20 16:16:38
Plan refresh time:             2023-08-05 01:00:00
=============================================
Idle data plan:                2
Idle SIM slot:                 slot-2
Idle Plan data limit:          100(MB)
Idle Plan overage status:      disable
Idle Plan Usage:               78(MB)
Idle Plan Usage percentage:    78.00%
Idle Plan refresh time:        2023-08-10 01:00:00
```

5. After the SIM card switch completes, view the active SIM card:

```
# diagnose sys lte-modem sim-info
LTE Modem SIM card information:
Active Slot: Slot 2
SIM state: QMI_UIM_CARD_STATE_PRESENT
ICCID: 89302370323035043340
IMSI: 302370605258650
```

```
Country: Canada
Network: Fido
SIM PIN status: Verified
```

Cellular interface of FortiGate-40F-3G4G supports IPv6 - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Cellular interface support for IPv6

The cellular interface of FG-40F-3G4G devices supports IPv6.

```
config system lte-modem
    set pdptype {IPv4 | IPv6 | IPv4v6}
end
```

`pdptype`	Specify the Packet Data Protocol (PDP) for the cellular interface: • `IPv4`: use only IPv4. • `IPv6`: use only IPv6. • `IPv4v6`: use both IPv4 and IPv6 (default).

Example

In this example, PDP type is set to IPv4v6 in the wireless profile.

To use IPv4v6:

1. On FortiGate-40F-3G4G, use the `execute lte-modem wireless-profile` command to create or modify a wireless profile with `pdptype` set to `IPv4v6`. See the 3G4G LTE Modem Operator's Manual for details.
2. List all profiles.

   In the following example, `PDP_Type` is set to `3` to indicate support for both IPv4 and IPv6.

```
# execute lte-modem wireless-profile list
ID    Type   Name                 APN                   PDP_Type  Authen  Username
 1    0                              ota.bell.ca               3                  0
 2    0         Bell                 ota.bell.ca               3                  0


Profile Type:
    0 ==> QMI_WDS_PROFILE_TYPE_3GPP

Profile PDP type:
    0 ==> QMI_WDS_PDP_TYPE_IPV4
    1 ==> QMI_WDS_PDP_TYPE_PPP
    2 ==> QMI_WDS_PDP_TYPE_IPV6
    3 ==> QMI_WDS_PDP_TYPE_IPV4_OR_IPV6
Authentication:
    0 ==> QMI_WDS_AUTHENTICATION_NONE
    1 ==> QMI_WDS_AUTHENTICATION_PAP
```

```
2 ==> QMI_WDS_AUTHENTICATION_CHAP
3 ==> QMI_WDS_AUTHENTICATION_PAP|QMI_WDS_AUTHENTICATION_CHAP
```

3. Apply the correct profile.

   In the following example, profile 2 is selected. The `apn` setting must also match the `apn` setting in the selected profile.

```
config sys lte-modem
    set pdptype ipv4v6
    set force-wireless-profile 2
    set apn ota.bell.ca
end
```

4. Wait for the profile to take effect, and then check the data session information:

```
# diagose sys lte-modem data-session-info
LTE Modem data session information:
Interface name:         wwan
IPV4 connection:        QMI_WDS_CONNECTION_STATUS_CONNECTED
IPV6 connection:        QMI_WDS_CONNECTION_STATUS_CONNECTED
Profile ID:             2
Data profile name:      Bell
Profile type:           QMI_WDS_PROFILE_TYPE_3GPP
PDP context type:       QMI_WDS_PDP_TYPE_IPV4_OR_IPV6
APN name:               ota.bell.ca
-------------------------------------
IP:                     10.34.139.21
IP gateway:             10.34.139.22
IP netmask:             255.255.255.252
Primary DNS:            161.216.153.1
Secondary DNS:          161.216.157.1
MTU:                    1500
Link protocol:          QMI_WDA_LINK_LAYER_PROTOCOL_RAW_IP
-------------------------------------
IPv6:                   2605:b100:93b:cf64:bd33:e6ba:b2ef:5e58
IPv6 prefix len:        64
IPv6 gateway:           2605:b100:93b:cf64:60c8:e41d:be4b:eaf5
IPv6 GW prefix len:     64
IPv6 PRI DNS:           2605:b100:880:9::1
IPv6 SEC DNS:           2605:b100:680:9::1
MTU:                    1500
Link protocol:          QMI_WDA_LINK_LAYER_PROTOCOL_RAW_IP
Auto connect:           QMI_WDS_AUTOCONNECT_DISABLED
Network type:           Unknown WDS Bearer Tech
Network type(last):     Unknown WDS Bearer Tech
```

5. Verify IPv4.

   In the following example, an IPv4 address is assigned to the wwan interface, and an IPv4 route is automatically added.

```
# diagnose ip address list
IP=192.168.2.111->192.168.2.111/255.255.255.0 index=5 devname=wan
IP=127.0.0.1->127.0.0.1/255.0.0.0 index=13 devname=root
IP=169.254.1.1->169.254.1.1/255.255.255.0 index=17 devname=fortilink
IP=192.168.1.99->192.168.1.99/255.255.255.0 index=18 devname=lan
IP=127.0.0.1->127.0.0.1/255.0.0.0 index=19 devname=vsys_ha
IP=127.0.0.1->127.0.0.1/255.0.0.0 index=21 devname=vsys_fgfm
IP=10.34.139.21->10.34.139.21/255.255.255.255 index=23 devname=wwan
```

```
FortiGate-40F-3G4G # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       V - BGP VPNv4
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 10.34.139.22, wwan, [1/0]
C       10.34.139.21/32 is directly connected, wwan
```

6. Verify IPv6.

   In the following example, an IPv6 address is assigned to the wwan interface, and an IPv6 route is automatically added.

```
# diagnose ipv6  address list
dev=13 devname=root flag=P scope=254 prefix=128 addr=::1 preferred=4294967295
valid=4294967295 cstamp=2861 tstamp=2861
dev=19 devname=vsys_ha flag=P scope=254 prefix=128 addr=::1 preferred=4294967295
valid=4294967295 cstamp=5231 tstamp=5231
dev=21 devname=vsys_fgfm flag=P scope=254 prefix=128 addr=::1 preferred=4294967295
valid=4294967295 cstamp=5875 tstamp=5875
dev=23 devname=wwan flag=P scope=0 prefix=64
addr=2605:b100:93b:cf64:bd33:e6ba:b2ef:5e58 preferred=4294967295 valid=4294967295
cstamp=102181 tstamp=102181
dev=23 devname=wwan flag=P scope=253 prefix=64 addr=fe80::8049:4eff:fefc:ea5e
preferred=4294967295 valid=4294967295 cstamp=102181 tstamp=102181

FortiGate-40F-3G4G # get router info6 routing-table database
IPv6 Routing Table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, B - BGP, V - BGP VPNv6
       > - selected route, * - FIB route, p - stale info
Timers: Uptime

Routing table for VRF=0
S    *> ::/0 [10/0] via 2605:b100:93b:cf64:60c8:e41d:be4b:eaf5, wwan, 00:09:20,
[1024/0]
C    *> ::1/128 via ::, root, 00:24:50
C    *> 2605:b100:93b:cf64::/64 via ::, wwan, 00:09:20
```

Connectivity Fault Management supported for network troubleshooting - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:
- Connectivity Fault Management

Some FortiGate hardware models support Connectivity Fault Management (CFM) technology. With CFM, administrators can easily diagnose and resolve issues in Ethernet networks. CFM provides tools for monitoring, testing, and verifying the connectivity and performance of network segments.

The following platforms support CFM:

FortiGate	FG-40F, FG-40F-3G4G, FG-60E, FG-60F, FG-61E, FG-61F, FG-80E, FG-80E-POE, FG-81E, FG-81E-POE, FG-81F, FG-90E-POE, FG-100F, FG-101F, FG-200E, FG-1100E
FortiWiFi	FWF-40F, FWF-60E, FWF-60F, FWF-61E, FWF-61F

Use the `config ethernet-oam cfm` command to configure the CFM protocol.

```
config ethernet-oam cfm
    edit <domain-id>
        set domain-name <string>
        set domain-level <integer>
        config service
            edit <service-id>
                set service-name <string>
                set interface "<string>"
                set mepid <integer>
                set message-interval <integer>
                set cos <integer>
                set sender-id Hostname {none | Hostname}
            next
        end
    next
end
```

`<domain-id>`	Specify the domain ID for the Ethernet layer operation, administration, and management (OAM) protocol. A unique domain ID is used to communicate with other peers under the same domain ID and domain level.
`domain-level <integer>`	Specify the OAM maintenance level (0 to 7, with 0 being the smallest and 7 being the largest). A unique domain level is used to communicate with other devices under the same domain ID and domain level.
`domain-name <string>`	Specify the OAM domain name or maintenance domain identifier (MDID). Other peer devices recognize the domain name. All devices in the same domain with the same service level can communicate with each other.

A domain can provide multiple services. Each service uses a special service ID. The following items describe a service:

`<service-id>`	Specify the ID for the service.
`service-name <string>`	Specify the name of the service.

`interface <string>`	Specify the name of the VLAN interface where the service is enabled. The service is associated with a particular VLAN network port and can't be accessed by other network ports.
`mepid <integer>`	Specify the unique ID of the maintenance association endpoints (MEP) (1 - 8191). The service is associated with a unique MEP ID and can't respond to other service requests of a different MEP ID.
`message interval <integer>`	Specify the continuity-check message frequency interval in milliseconds. Determines how long to send a continuity-check message to determine whether the service is alive.
`cos <integer>`	Specify the class of service (COS) bit for continuity-check messages (0 to 7). CoS is an optional, special bit in the packet of continuity-check messages.
`sender-id {none | hostname}`	Specify the type, length, value (TLV) sender ID: • `none`: indicates no sender ID. • `hostname`: uses the Fortinet production name of the device as the sender ID, for example, FortiGate-80F. The sender ID is an optional column that includes a hostname in the packet of continuity-check messages.

The following diagnose commands can be used with this feature:

`diagnose ethernet-oam cfmpeer`	Locate peers configured with `config ethernet-oam cfm` that are using the CFM Continuity Check Protocol (CCP) protocol to connect to the CCP daemon (CCD).
`diagnose debug application cfmd {enable | disable}`	Enable or disable debugging messages of the CFM protocol. • `enable`: enable debugging messages for the CFM protocol. Messages appear on the console. • `disable`: disable debugging messages.

The following execute commands can be used with this feature:

`execute ethernet ping`	Check if an interface has a peer with mac address and level available under CFM support.
`execute ethernet traceroute`	Check the Ethernet traceroute with the peer FortiGate. The traceroute is instructed to achieve a peer through an interface with mac_address and level available under CFM support.

Example

In this example, an interface (vlan101) connects FortiGate 81F to FortiGate 101F. CFM is configured for the interface (vlan101) on the FortiGate 81F. All steps are performed on the FortiGate 101F.

Because this feature is based on IEEE 802.1Q, an IP address is not needed to connect the interface.

To configure and use CFM :

1. Configure CFM for the interface named `vlan101`:

```
config ethernet-oam cfm
    edit 1
        set domain-name cfm-test
        set domain-level 1
        config service
            edit 1
                set service-name vlan-101
                set interface "vlan101"
                set mepid 101
                set message-interval 10000
                set cos 7
                set sender-id Hostname
            next
        end
    next
end
```

2. On the FortiGate 101F, show the peers connecting to the device:

```
# diagnose ethernet-oam cfmpeer
wait for the responses from CCD daemons ...

========                MEPs (pid 11251)               ========
======== domain_name: cfm-test service_name: vlan-101 mepid: 101 ========
1  MAC = e0:23:ff:9b:07:0a, state = UP, mdlevel = 1, domain_name = cfm-test, service_
name = vlan-101, mepid = 81, TLV_port_status = PsUP, TLV_interface_status = isUp
========                     END                    ========
```

3. On FortiGate 101F, check whether the interface has a peer under CFM support:

```
# execute ethernet ping vlan101 1 5 e0:23:ff:9b:07:0a
Sending CFM LBM to e0:23:ff:9b:07:0a
64 bytes from e0:23:ff:9b:07:0a, sequence 422603820, 1 ms
64 bytes from e0:23:ff:9b:07:0a, sequence 422603821, 1 ms
64 bytes from e0:23:ff:9b:07:0a, sequence 422603822, 1 ms
64 bytes from e0:23:ff:9b:07:0a, sequence 422603823, 1 ms
64 bytes from e0:23:ff:9b:07:0a, sequence 422603824, 1 ms
```

4. Execute the Ethernet traceroute:

```
# execute ethernet traceroute vlan101 1 e0:23:ff:9b:07:0a
Sending CFM LTM probe to e0:23:ff:9b:07:0a
ethtrace_main: flags = 0, usefdbonly = 0
ttl 1: LTM with id 984984516
cfm_matchltr - 384
cfm_matchltr - 404
        reply from e0:23:ff:9b:07:0a, id=984984516, ttl=0, RlyHit
```

Support LTE / BLE airplane mode for FGR-70F-3G4G - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Airplane mode and LTE/BLE

Airplane mode is supported on FGR-70F-3G4G models to enable/disable radio frequency signals for the internal LTE modem and Bluetooth Low Energy (BLE) module:

```
config system global
    set airplane-mode {disable | enable}
end
```

By default airplane mode is disabled, and LTE and BLE radio frequency signals are transmitted. Airplane mode can be enabled with a CLI command followed by a reboot of the FortiGate. Once airplane mode is enabled, LTE and BLE radio frequency signals remain silent during normal operation of the FortiGate.

A specific BIOS version is required to ensure radio frequency signals remain silent for LTE and BLE modules when FortiGate is rebooted.

`set airplane-mode {disable | enable}`	Enable or disable airplane mode on FGR-70F-3G4G models: • `disable`: disable airplane mode, which means radio frequency signals of the internal LTE modem and BLE module are enabled and transmitted. • `enable`: enable airplane mode, which means radio frequency signals of the internal LTE modem and BLE module are turned off.

Example

To disable airplane mode:

1. Disable airplane mode:

```
config system global
    ...
    set airplane-mode disable
    ...
end
```

   Radio frequency signals of the LTE modem and BLE module are turned on.

2. Use the following commands to verify the settings:

`execute usb-device list`	Check the status of the LTE modem.
`diagnose test application lted 5`	Check the signal strength of the LTE modem.
`diagnose sys lte-modem modem-details`	Get detailed information about the LTE modem.

`diagnose sys lte-modem data-session-info`	Get session information about the LTE modem.
`diagnose bluetooth test_ bt_conn`	Check the status of the BLE mode.
`diagnose bluetooth status`	Check the bluetooth status of the BLE mode.

To enable airplane mode:

1. Enable airplane mode:

```
config system global
    ...
    set airplane-mode enable
    ...
Enabling airplane mode will turn off LTE modem and Bluetooth RF signals.
Do you want to continue? (y/n)y
end
```

2. Reboot the FortiGate.

```
execute reboot
This operation will reboot the system !
Do you want to continue? (y/n)y
```

The LTE modem and BLE module are disabled, and radio frequency signals are turned off.

3. Show the configuration to confirm that airplane mode is enabled.

```
show full-configuration
config system global
    ...
    set airplane-mode enable
    ...
end
```

4. Check the USB device list (`execute usb-device list`) to confirm that the modem is not displayed in the list.
5. Check the signal strength to confirm that the modem is not found.

```
# diagnose test application lted 5
Modem device not currently connected! Please try again later...
```

6. Check the modem details to confirm that airplane mode is enabled, and the modem is not detected.

```
# diagnose sys lte-modem modem-details
LTE Modem detailed information:
system.global.airplane-mode:      On
Modem detected:   No
```

7. Check the modem session information to confirm that the modem is not detected.

```
# diagnose sys lte-modem data-session-info
LTE Modem data session information:
Modem not detected!
```

8. Run a Bluetooth test to confirm that airplane mode is on and Bluetooth testing is not allowed.

```
# diagnose bluetooth test_bt_conn
It's on airplane mode now. Bluetooth testing is not allowed.
```

9. Check Bluetooth status to confirm that the BLE module is disabled.

```
# diagnose bluetooth status
Bluetooth Status: RESET BOOTLOADER
Connect State(0):  BLE_MODE_DISABLED
```

BGP incorporates the advanced security measures of TCP Authentication Option (TCP-AO) - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- TCP Authentication Option advanced security measures

Border Gateway Protocol (BGP) incorporates the advanced security measures of TCP Authentication Option (TCP-AO), which supports stronger algorithms, such as AES-128 CMAC and HMAC-SHA1. This integration bolsters the security of and enhances the reliability of BGP connections and contributes to the overall security of the internet.

CLI changes include:

- Added `cmac-aes128` option in the router key-chain:

```
config router key-chain
    edit <name>
        config key
            edit <id>
                set algorithm cmac-aes128
            next
        end
    next
end
```

- Added `auth-options` for BGP `neighbor` and `neighbor-group`:

```
config router bgp
    config neighbor|neighbor-group
        edit <string>
            set auth-options <string>
        end
    next
end
```

- Added debug command for `tcp-auth-options`:

```
diagnose sys tcp-auth-options
```

Example

In this example, the router BGP neighbor is configured to use the AES-128 CMAC algorithm.

To configure the router BGP to use the AES-128 CMAC algorithm:

1. Configure the router key-chain to use the AES-128 CMAC algorithm:

```
config router key-chain
    edit "11"
        config key
            edit "1"
                set accept-lifetime 01:01:01 01 01 2021 2147483646
                set send-lifetime 01:01:01 01 01 2021 2147483646
                set key-string **********
                set algorithm cmac-aes128
            next
        end
    next
end
```

2. Apply the key-chain to the BGP neighbor or neighbor group:
   In this example, the key-chain is applied to the BGP neighbor with IP address 2.2.2.2.

```
config router bgp
    set as 65412
    config neighbor
        edit "2.2.2.2"
            set auth-options "11"
        next
    end
end
```

3. Verify that the router BGP is using the algorithm.
   The command output shows that BGP neighbor 2.2.2.2 is using the AES-128 CMAC algorithm.

```
# diagnose sys tcp-auth-options

VFID=0 send-id=1 recv-id=1 flags=0x784 keylen=6
alg=2(aes128) addr=2.2.2.2
send-begin: Fri Jan  1 01:01:01 2021
send-end: Wed Jan 19 04:15:07 2089
recv-begin: Fri Jan  1 01:01:01 2021
recv-end: Wed Jan 19 04:15:07 2089
```

Allow multiple sFlow collectors - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:
- Example 1: multiple sFlow collectors in a non-VDOM environment
- Example 2: multiple sFlow collectors in a multi-VDOM environment

FortiOS can be configured with a maximum of three sFlow collectors. This also applies to multi-VDOM environments where a maximum of three sFlow collectors can be used globally and/or on a per-VDOMs basis. This feature enables up to a maximum of three unique parallel sFlow streams or transmissions per sFlow sample to three different sFlow collectors. The sFlow collector configuration can only be configured in the CLI.

```
config system {sflow | vdom-sflow}
    config collectors
```

```
        edit <id>
            set collector-ip <IPv4_address>
            set collector-port <port>
            set source-ip <IPv4_address>
            set interface-select-method {auto | sdwan | specify}
            set interface <interface>
        next
    end
end
```

`collector-ip <IPv4_address>`	Enter the IP address of the sFlow collector that sFlow agents added to interfaces in this VDOM send sFlow datagrams to (default = 0.0.0.0).
`collector-port <port>`	Enter the UDP port number used for sending sFlow datagrams; only configure if required by the sFlow collector or network configuration (0 - 65535, default = 6343).
`source-ip <IPv4_address>`	Enter the source IP address for the sFlow agent.
`interface-select-method {auto | sdwan | specify}`	Specify how to select the outgoing interface to reach the server. • `auto`: set the outgoing interface automatically. • `sdwan`: set the outgoing interface by SD-WAN or policy routing rules. • `specify`: set the outgoing interface manually.
`interface <interface>`	Enter the outgoing interface to reach the server.

Example 1: multiple sFlow collectors in a non-VDOM environment

In this example, three sFlow collectors are configured in a non-VDOM environment with sFlow sampling on the wan1 interface.

To configure multiple sFlow collectors:

1. Configure the sFlow collectors:

```
config system sflow
    config collectors
        edit 1
            set collector-ip 10.1.1.1
            set collector-port 6344
            set source-ip 0.0.0.0
            set interface-select-method auto
        next
        edit 2
            set collector-ip 10.1.1.2
            set collector-port 6345
            set source-ip 0.0.0.0
            set interface-select-method auto
        next
        edit 3
            set collector-ip 10.1.1.3
            set collector-port 6346
            set source-ip 0.0.0.0
            set interface-select-method auto
```

```
        next
    end
end
```

2. Configure sFlow sampling on wan1:

```
config system interface
    edit wan1
        set sflow-sampler enable
        set sample-rate 2000
        set polling-interval 20
        set sample-direction both
    next
end
```

3. Verify the sFlow diagnostics.
   a. Verify the sFlow configuration status:

```
# diagnose test application sflowd 1

global collector:10.1.1.1:[6344]
 global source ip: 0.0.0.0:[1399]

global collector:10.1.1.2:[6345]
 global source ip: 0.0.0.0:[1399]

global collector:10.1.1.3:[6346]
 global source ip: 0.0.0.0:[1399]
vdom: root, index=0, vdom sflow collector is disabled(use global sflow config),
primary (management vdom)
  intf:wan1 sample_rate:2000 polling_interval:20 sample_direction:both
```

   b. Verify the sampled sFlow traffic packet capture:

```
# diagnose sniffer packet any 'port 1399' 4 0 l
interfaces=[any]
filters=[port 6344 or port 6345 or port 6346]
2023-11-14 15:44:41.658799 wan1 out 172.16.151.157.1399 -> 10.1.1.1.6344: udp 144
2023-11-14 15:44:41.658829 wan1 out 172.16.151.157.1399 -> 10.1.1.2.6345: udp 144
2023-11-14 15:44:41.658848 wan1 out 172.16.151.157.1399 -> 10.1.1.3.6346: udp 144
```

The outgoing interface that is used to send the sampled sFlow traffic to the sFlow collector is decided by the routing table lookup.

Example 2: multiple sFlow collectors in a multi-VDOM environment

In this example, three sFlow collectors are configured in a multi-VDOM environment globally and per VDOM. sFlow sampling is on the wan1 and dmz interfaces.

To configure multiple sFlow collectors:

1. Configure the global sFlow collectors:

```
config system sflow
    config collectors
        edit 1
            set collector-ip 10.1.1.1
            set collector-port 6344
            set source-ip 0.0.0.0
            set interface-select-method auto
        next
        edit 2
            set collector-ip 10.1.1.2
            set collector-port 6345
            set source-ip 0.0.0.0
            set interface-select-method auto
        next
        edit 3
            set collector-ip 10.1.1.3
            set collector-port 6346
            set source-ip 0.0.0.0
            set interface-select-method auto
        next
    end
end
```

2. Configure the per-VDOM sFlow collectors:

```
config vdom
    edit testvdom
        config system vdom-sflow
            set vdom-sflow enable
            config collectors
                edit 1
                    set collector-ip 10.1.1.4
                    set collector-port 6347
                    set source-ip 0.0.0.0
                    set interface-select-method auto
                next
                edit 2
                    set collector-ip 10.1.1.5
                    set collector-port 6348
                    set source-ip 0.0.0.0
                    set interface-select-method auto
                next
                edit 3
                    set collector-ip 10.1.1.6
                    set collector-port 6349
                    set source-ip 0.0.0.0
                    set interface-select-method auto
                next
            end
        end
    next
end
```

3. Configure sFlow sampling on wan1 and dmz:

```
config system interface
    edit wan1
        set vdom "root"
        set sflow-sampler enable
        set sample-rate 2000
        set polling-interval 20
        set sample-direction both
    next
    edit dmz
        set vdom "testvdom"
        set sflow-sampler enable
        set sample-rate 2000
        set polling-interval 20
        set sample-direction both
    next
end
```

4. Verify the sFlow diagnostics.
   a. Verify the sFlow configuration status:

```
# diagnose test application sflowd  1

global collector:10.1.1.1:[6344]
 global source ip: 0.0.0.0:[1399]

global collector:10.1.1.2:[6345]
 global source ip: 0.0.0.0:[1399]

global collector:10.1.1.3:[6346]
 global source ip: 0.0.0.0:[1399]
vdom: root, index=0, vdom sflow collector is disabled(use global sflow config),
primary (management vdom)
  intf:wan1 sample_rate:2000 polling_interval:20 sample_direction:both
vdom: testvdom, index=1, vdom sflow collector is enabled, primary
  collector:10.1.1.4:[6347] src:192.168.1.1:[1399]
  collector:10.1.1.5:[6348] src:192.168.1.1:[1399]
  collector:10.1.1.6:[6349] src:192.168.1.1:[1399]
  intf:dmz sample_rate:2000 polling_interval:20 sample_direction:both
```

   b. Verify the sampled sFlow traffic packet capture:

```
# sudo root diagnose sniffer packet any 'port 1399' 4 0 l
interfaces=[any]
filters=[port 1399]
2023-11-14 16:50:11.118807 wan1 out 172.16.151.157.1399 -> 10.1.1.1.6344: udp 144
2023-11-14 16:50:11.118838 wan1 out 172.16.151.157.1399 -> 10.1.1.2.6345: udp 144
2023-11-14 16:50:11.118865 wan1 out 172.16.151.157.1399 -> 10.1.1.3.6346: udp 144
2023-11-14 16:50:20.198784 dmz out 192.168.1.1.1399 -> 10.1.1.4.6347: udp 144
2023-11-14 16:50:20.198813 dmz out 192.168.1.1.1399 -> 10.1.1.5.6348: udp 144
2023-11-14 16:50:20.198832 dmz out 192.168.1.1.1399 -> 10.1.1.6.6349: udp 144
```

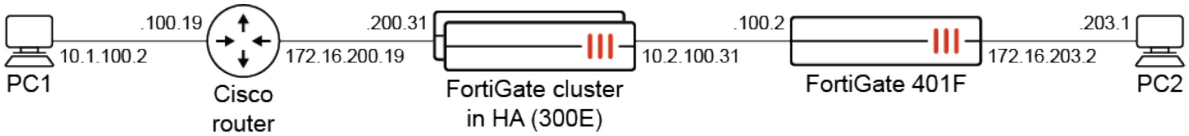
The outgoing interface that is used to send the sampled sFlow traffic to the sFlow collector is decided by the routing table lookup.

Support BGP graceful restart helper-only mode - 7.4.2

This feature ensures that during a FortiGate HA failover, the neighboring router that only supports BGP graceful restart helper mode retains its routes. This is crucial as it prevents any loss of traffic packets, and provides a more reliable and seamless network experience for the customer.

Example

In this example, a cluster of FortiGate 300Es are in HA and form BGP neighbors with a Cisco router and FortiGate 401F. The Cisco router is configured with graceful restart helper-only mode and will retain its routes during an HA failover. The FortiGate 401F is configured with graceful restart mode and will retain its routes during an HA failover. If PC1 keeps pinging PC2 during the HA failover, there will be no traffic loss.



To configure BGP graceful restart:

1. Configure BGP graceful restart on the HA cluster:

```
config router bgp
    set graceful-restart enable
    config neighbor
        edit "172.16.200.19"
            set capability-graceful-restart enable
        next
    end
end
```

2. Configure BGP graceful restart on the FortiGate 401F:

```
config router bgp
    set graceful-restart enable
    config neighbor
        edit "10.2.100.31"
            set capability-graceful-restart enable
        next
    end
end
```

3. After an HA failover, verify the status of the Cisco router neighbor:

```
# get router info bgp neighbors 172.16.200.19
VRF 0 neighbor table:
BGP neighbor is 172.16.200.19, remote AS 20, local AS 20, internal link
  BGP version 4, remote router ID 5.5.5.5
  BGP state = Established, up for 00:38:05
  Last read 00:00:07, hold time is 180, keepalive interval is 60 seconds
  Configured hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received (old and new)
    Graceful restart helper
```

```
    Address family IPv4 Unicast: advertised and received
    Address family VPNv4 Unicast: advertised
    Address family IPv6 Unicast: advertised
    Address family VPNv6 Unicast: advertised
    Address family L2VPN EVPN: advertised
  Received 47 messages, 0 notifications, 0 in queue
  Sent 51 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  NLRI treated as withdraw: 0
  Minimum time between advertisement runs is 30 seconds

 For address family: IPv4 Unicast
  BGP table version 4, neighbor version 4
  Index 1, Offset 0, Mask 0x2
  AF-dependant capabilities:
    Graceful restart: advertised, helper
  Inbound soft reconfiguration allowed
  NEXT_HOP is always this router
  Community attribute sent to this neighbor (both)
  1 accepted prefixes, 1 prefixes in rib
  12 announced prefixes

 For address family: VPNv4 Unicast
  BGP table version 1, neighbor version 1
  Index 1, Offset 0, Mask 0x2
  Community attribute sent to this neighbor (both)
  0 accepted prefixes, 0 prefixes in rib
  0 announced prefixes

 For address family: IPv6 Unicast
  BGP table version 4, neighbor version 4
  Index 1, Offset 0, Mask 0x2
  Community attribute sent to this neighbor (both)
  0 accepted prefixes, 0 prefixes in rib
  0 announced prefixes

 For address family: VPNv6 Unicast
  BGP table version 1, neighbor version 1
  Index 1, Offset 0, Mask 0x2
  Community attribute sent to this neighbor (both)
  0 accepted prefixes, 0 prefixes in rib
  0 announced prefixes

 For address family: L2VPN EVPN
  BGP table version 1, neighbor version 1
  Index 1, Offset 0, Mask 0x2
  Community attribute sent to this neighbor (both)
  0 accepted prefixes, 0 prefixes in rib
  0 announced prefixes

 Connections established 1; dropped 0
 Graceful-restart Status:
  Remote restart-time is 120 sec

Local host: 172.16.200.31, Local port: 2608
Foreign host: 172.16.200.19, Foreign port: 179
```

```
Egress interface: 9
Nexthop: 172.16.200.31
Nexthop interface: port1
Nexthop global: ::
Nexthop local: ::
BGP connection: non shared network
```

4. Verify the status of the FortiGate 401F neighbor:

```
# get router info bgp neighbors 10.2.100.2
VRF 0 neighbor table:
BGP neighbor is 10.2.100.2, remote AS 65412, local AS 20, external link
  BGP version 4, remote router ID 2.2.2.2
  BGP state = Established, up for 00:33:33
  Last read 00:00:57, hold time is 180, keepalive interval is 60 seconds
  Configured hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received (old and new)
    Address family IPv4 Unicast: advertised and received
    Address family VPNv4 Unicast: advertised and received
    Address family IPv6 Unicast: advertised and received
    Address family VPNv6 Unicast: advertised and received
    Address family L2VPN EVPN: advertised and received
  Received 44 messages, 0 notifications, 0 in queue
  Sent 42 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  NLRI treated as withdraw: 0
  Minimum time between advertisement runs is 30 seconds

 For address family: IPv4 Unicast
  BGP table version 2, neighbor version 2
  Index 2, Offset 0, Mask 0x4
  AF-dependant capabilities:
    Graceful restart: advertised, received, negotiated

  Inbound soft reconfiguration allowed
  Community attribute sent to this neighbor (both)
  12 accepted prefixes, 12 prefixes in rib
  1 announced prefixes

 For address family: VPNv4 Unicast
  BGP table version 1, neighbor version 1
  Index 2, Offset 0, Mask 0x4
  Community attribute sent to this neighbor (both)
  0 accepted prefixes, 0 prefixes in rib
  0 announced prefixes

 For address family: IPv6 Unicast
  BGP table version 2, neighbor version 2
  Index 2, Offset 0, Mask 0x4
  Community attribute sent to this neighbor (both)
  1 accepted prefixes, 1 prefixes in rib
  0 announced prefixes

 For address family: VPNv6 Unicast
  BGP table version 1, neighbor version 1
  Index 2, Offset 0, Mask 0x4
```

```
    Community attribute sent to this neighbor (both)
    0 accepted prefixes, 0 prefixes in rib
    0 announced prefixes

   For address family: L2VPN EVPN
    BGP table version 1, neighbor version 1
    Index 2, Offset 0, Mask 0x4
    Community attribute sent to this neighbor (both)
    0 accepted prefixes, 0 prefixes in rib
    0 announced prefixes

   Connections established 1; dropped 0
   Graceful-restart Status:
    Remote restart-time is 120 sec

  Local host: 10.2.100.31, Local port: 4438
  Foreign host: 10.2.100.2, Foreign port: 179
  Egress interface: 10
  Nexthop: 10.2.100.31
  Nexthop interface: port2
  Nexthop global: ::
  Nexthop local: ::
  BGP connection: non shared network
```

Support for LAN extension VDOM simplifications - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:
- FortiGate secure edge to FortiSASE

VDOM configuration for the FortiGate LAN extension has been simplified. When you configure the FortiGate LAN extension VDOM, FortiOS automatically configures a VDOM link between a traffic VDOM, which is by default the root VDOM, and the LAN extension VDOM.

After connecting to the FortiGate Controller, the following settings are automatically configured on the FortiGate Connector:

- VDOM link interface in the LAN extension VDOM is a part of the LAN extension software switch.
- VDOM link interface in the traffic VDOM is dynamically assigned an IP address obtained through the FortiGate Controller.

This feature supports the FortiGate secure edge for FortiSASE.

Example

This example demonstrates how to configure the FortiGate Connector to connect to FortiSASE as the FortiGate Controller.

To configure the FortiGate Connector using the CLI:

1. Enable multi-VDOM mode from the CLI:

```
config system global
    set vdom-mode multi-vdom
end
```

2. Verify that the FortiExtender setting is enabled in the global VDOM:

```
# config global
# show full system global | grep fortiextender -f
…
    set fortiextender enable
…
```

3. Create a new LAN extension VDOM with the LAN extension controller address as the FortiSASE domain name. See Connecting FortiGate to FortiSASE using GUI and CLI for details on how to find the FortiSASE domain name. In this example, the VDOM name is `ext`, and the FortiSASE domain name is `turbo-a1p0hv3p.edge.prod.fortisase.com`.

```
config vdom
    edit ext
        config system settings
            set vdom-type lan-extension
            set lan-extension-controller-addr turbo-
a1p0hv3p.edge.prod.fortisase.com
            set ike-port 4500
        end
    next
end
```

4. Move interfaces from the root VDOM to the new LAN extension VDOM, and set the appropriate WAN and LAN roles.
   - Before moving an interface to a new VDOM, delete all references, such as firewall policies or firewall objects. See Finding object dependencies.
   - If interfaces are already part of a hardware switch, remove them from the hardware switch to make them available for the new VDOM. See Hardware switch.

   In this example from the global VDOM, the `WAN1` and `internal1` interfaces are moved to the LAN extension VDOM named `ext`, and their roles are set appropriately as `wan` and `lan`.

```
config global
    config system interface
        edit WAN1
            set vdom "ext"
            set role wan
        next
        edit internal1
            set vdom "ext"
            set role lan
        next
    end
```

```
end
```

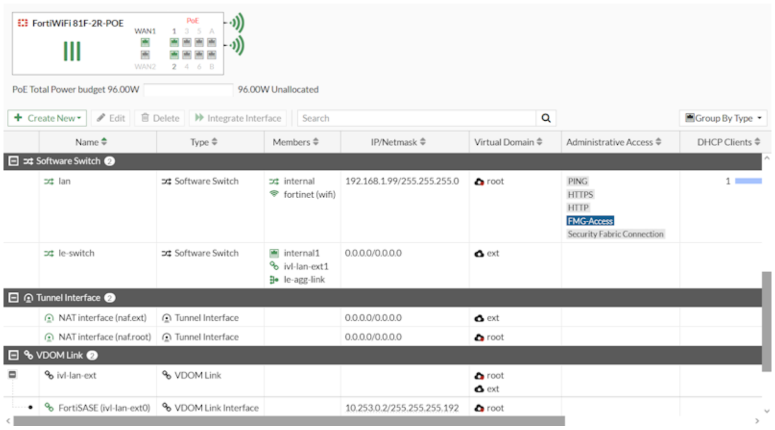
5. For the WAN interface within the LAN extension VDOM, edit the interface and ensure that Security Fabric connections are allowed:

```
config vdom
    edit ext
        config system interface
            edit WAN1
                set allowaccess ping fabric
            next
        end
    next
end
```

   This configuration assumes that the WAN and LAN interfaces are already configured with static IP addresses or configured to use DHCP accordingly.
6. (Optional) If your LAN extension VDOM is not configured as the management VDOM, and you require a custom DNS server to resolve the FortiGate Controller hostname, then you must configure the VDOM DNS settings within the VDOM:

```
config vdom
    edit ext
        config system vdom-dns
            set vdom-dns enable
            set primary 1.2.3.4
            set secondary 2.3.4.5
        end
    next
end
```

7. After the LAN extension VDOM connects to FortiSASE, observe from the global VDOM under *Network* > *Interfaces*:
   - A VDOM link *ivl-lan-ext* is created.
   - The VDOM link interface in the LAN extension VDOM (*ivl-lan-ext1*) is part of the *le-switch* LAN extension software switch. Network connectivity to the FortiGate Controller (that is, to FortiSASE) is achieved through the software switch.
   - The VDOM link interface in the traffic (root) VDOM (*ivl-lan-ext0*) has obtained an IP address dynamically from the FortiGate Controller.
     The traffic VDOM can be used to:
     - Apply application steering to the local internet connection or to FortiGate Controller network (FortiSASE) using SD-WAN.
     - Apply local security features for traffic egressing the local internet connection, such as antivirus, intrusion prevention security (IPS), application control, and web filtering, by creating a firewall policy with *ivl-lan-ext0* as the destination interface.



8. Create a firewall policy with *ivl-lan-ext0* as the destination and *lan* as the source within the traffic VDOM to allow local traffic from the FortiGate Connector to access the internet through the FortiGate Controller (FortiSASE):

```
config firewall policy
    edit 1
        set name "traffic-VDOM-to-FortiSASE"
        set srcintf "lan"
        set dstintf "ivl-lan-ext0"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

Allow multiple Netflow collectors - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:
- Allow multiple Netflow collectors

FortiOS can be configured with a maximum of six NetFlow collectors. This also applies to multi-VDOM environments where a maximum of six NetFlow collectors can be used globally or on a per-VDOMs basis. This feature enables up to a maximum of six unique parallel NetFlow streams or transmissions per NetFlow sample to six different NetFlow collectors. The NetFlow collector configuration can only be configured in the CLI.

```
config system {netflow | vdom-netflow}
    config collectors
        edit <id>
            set collector-ip <IP address>
            set collector-port <port>
            set source-ip <IP address>
            set interface-select-method {auto | sdwan | specify}
            set interface <interface>
        next
```

```
    end
end
```

`collector-ip`	Enter the IPv4 or IPv6 address of the NetFlow collector that NetFlow agents added to interfaces in this VDOM send NetFlow datagrams to.
`collector-port`	Enter the UDP port number used for sending NetFlow datagrams; only configure if it is required by the NetFlow collector or network configuration (0 - 65535, default = 6343).
`source-ip`	Enter the source IPv4 or IPv6 address for the NetFlow agent.
`interface-select-method`	Specify how to select the outgoing interface to reach the server. • `auto`: Set the outgoing interface automatically. • `sdwan`: Set the outgoing interface by SD-WAN or policy routing rules. • `specify`: Set the outgoing interface manually.
`interface <interface>`	Enter the outgoing interface to reach the server.

If the `interface-select-method` is set to `auto`, the outgoing interface that is used to send the sampled NetFlow traffic to the NetFlow collector is decided by the routing table lookup.

Example 1: Multiple NetFlow collectors in a non-VDOM environment

In this example, six NetFlow collectors are configured in a non-VDOM environment with NetFlow sampling on the port1 interface.

To configure multiple NetFlow collectors:

1. Configure the NetFlow collectors:

```
config system netflow
   config collectors
   set active-flow-timeout 60
   set template-tx-timeout 60
        edit 1
            set collector-ip 172.16.200.155
            set collector-port 2055
            set source-ip 172.16.200.6
            set interface-select-method specify
            set interface "port1"
        next
        edit 2
            set collector-ip 10.1.100.59
            set collector-port 2056
            set source-ip 10.1.100.6
            set interface-select-method specify
            set interface "port2"
        next
        edit 3
            set collector-ip 172.18.60.80
```

```
            set collector-port 2057
            set interface-select-method specify
            set interface "port1"
        next
        edit 4
            set collector-ip "172.18.60.1"
            set collector-port 2058
        next
        edit 5
            set collector-ip "172.18.60.3"
            set collector-port 2059
        next
        edit 6
            set collector-ip "172.18.60.4"
            set collector-port 2060
        next
    end
end
```

2. Configure NetFlow sampling on port1:

```
config system interface
    edit port1
        set netflow-sampler both
    next
end
```

3. Verify the NetFlow diagnostics.
   a. Verify the NetFlow configuration status:

```
# diagnose test application sflowd 3

===== Netflow Vdom Configuration =====
Global collector(s) active-timeout(seconds):60 inactive-timeout(seconds):15
    Collector id:1: 172.16.200.155[2055] source IP:172.16.200.6
    Collector id:2: 10.1.100.59[2056] source IP:10.1.100.6
    Collector id:3: 172.18.60.80[2057] source IP:
    Collector id:4: 172.18.60.1[2058] source IP:
    Collector id:5: 172.18.60.3[2059] source IP:
    Collector id:6: 172.18.60.4[2060] source IP:
____ vdom: root, index=0, is master, collector: disabled (use global config) (mgmt
vdom)
   |_ coll_ip:172.16.200.155:2056,src_ip:172.16.200.6
   |_ coll_ip:10.1.100.59:2057,src_ip:10.1.100.6
   |_ coll_ip:172.18.60.80:2058,src_ip:172.16.200.6
   |_ coll_ip:172.18.60.1:2058,src_ip:172.16.200.6
   |_ coll_ip:172.18.60.3:2059,src_ip:172.16.200.6
   |_ coll_ip:172.18.60.4:2060,src_ip:172.16.200.6
   |_ seq_num:13 pkts/time to next template: 16/29
   |_ exported: Bytes:2533746, Packets:3911, Sessions:70 Flows:70
   |_ active_intf: 1
   |____ interface:port1 sample_direction:both device_index:9 snmp_index:3
```

   b. Verify the sampled NetFlow traffic packet capture:

```
# diagnose sniffer packet any 'udp and port 2056 or 2057 or 2058' 4
```

```
filters=[udp and port 2056 or 2057 or 2058]
5.717060 port1 out 172.16.200.6.2472 -> 172.16.200.155.2055: udp 60
5.717068 port2 out 10.1.100.6.2472 -> 10.1.100.59.2056: udp 60
5.717075 port1 out 172.16.200.6.2472 -> 172.18.60.80.2057: udp 60
5.717078 port1 out 172.16.200.6.2472 -> 172.18.60.1.2058: udp 60
5.717081 port1 out 172.16.200.6.2472 -> 172.18.60.3.2059: udp 60
5.717085 port1 out 172.16.200.6.2472 -> 172.18.60.4.2060: udp 60
```

Example 2: Multiple NetFlow collectors in a multi-VDOM environment

In this example, six NetFlow collectors are configured in a multi-VDOM environment globally and per VDOM. NetFlow sampling is on the port1 and port4 interfaces.

Please note it is not mandatory to set up per-VDOM NetFlow collectors in a multi-vdom environment. However, if you don't enable per-VDOM collectors, the settings of the global NetFlow Collector will be used instead.

To configure multiple NetFlow collectors:

1. Configure the global NetFlow collectors:

```
config system netflow
   config collectors
   set active-flow-timeout 60
   set template-tx-timeout 60
        edit 1
            set collector-ip 172.16.200.155
            set collector-port 2055
            set source-ip 172.16.200.6
            set interface-select-method specify
            set interface "port1"
        next
        edit 2
            set collector-ip 10.1.100.59
            set collector-port 2056
            set source-ip 10.1.100.6
            set interface-select-method specify
            set interface "port2"
        next
        edit 3
            set collector-ip 172.18.60.80
            set collector-port 2057
            set interface-select-method specify
            set interface "port1"
        next
        edit 4
            set collector-ip "172.18.60.1"
            set collector-port 2058
        next
        edit 5
            set collector-ip "172.18.60.3"
            set collector-port 2059
        next
```

```
        edit 6
            set collector-ip "172.18.60.4"
            set collector-port 2060
        next
    end
end
```

2. Configure the per-VDOM NetFlow collectors:

```
config system vdom-netflow
    set vdom-netflow enable
    config collectors
        edit 1
            set collector-ip "172.10.100.101"
            set collector-port 2059
        next
        edit 2
            set collector-ip "172.10.100.102"
            set collector-port 2060
        next
        edit 3
            set collector-ip "172.10.100.103"
            set collector-port 2061
        next
        edit 4
            set collector-ip "172.10.100.104"
            set collector-port 2062
        next
        edit 5
            set collector-ip "172.10.100.105"
            set collector-port 2063
        next
        edit 6
            set collector-ip "172.10.100.106"
            set collector-port 2064
        next
    end
end
```

3. Configure NetFlow sampling on port1 and port4:

```
config system interface
    edit port1
        set netflow-sampler both
    next
    edit port4
        set netflow-sampler both
    next
end
```

In a multi-VDOM environment, ensure the interface selected for NetFlow sampling is in the same VDOM as the per-VDOM NetFlow collector. For global NetFlow collectors, the interface selected for NetFlow sampling should be in the management VDOM.

4. Verify the NetFlow diagnostics.

a. Verify the NetFlow configuration status:

```
# diagnose test application sflowd 3

===== Netflow Vdom Configuration =====
Global collector(s) active-timeout(seconds):60 inactive-timeout(seconds):15
    Collector id:1: 172.16.200.155[2055] source IP:172.16.200.6
    Collector id:2: 10.1.100.59[2056] source IP:10.1.100.6
    Collector id:3: 172.18.60.80[2057] source IP:
    Collector id:4: 172.18.60.1[2058] source IP:
    Collector id:5: 172.18.60.3[2059] source IP:
    Collector id:6: 172.18.60.4[2060] source IP:
____ vdom: root, index=0, is master, collector: disabled (use global config) (mgmt
vdom)
   |_ coll_ip:172.16.200.155:2056,src_ip:172.16.200.6
   |_ coll_ip:10.1.100.59:2057,src_ip:10.1.100.6
   |_ coll_ip:172.18.60.80:2058,src_ip:172.16.200.6
   |_ coll_ip:172.18.60.1:2058,src_ip:172.16.200.6
   |_ coll_ip:172.18.60.3:2059,src_ip:172.16.200.6
   |_ coll_ip:172.18.60.4:2060,src_ip:172.16.200.6
   |_ seq_num:13 pkts/time to next template: 16/29
   |_ exported: Bytes:2533746, Packets:3911, Sessions:70 Flows:70
   |_ active_intf: 1
   |____ interface:port1 sample_direction:both device_index:9 snmp_index:3
____ vdom: vdom1, index=1, is master, collector: enabled
   |_ coll_ip:172.10.100.101:2059,src_ip:20.1.100.111
   |_ coll_ip:172.10.100.102:2060,src_ip:20.1.100.111
   |_ coll_ip:172.10.100.103:2061,src_ip:20.1.100.111
   |_ coll_ip:172.10.100.104:2062,src_ip:20.1.100.111
   |_ coll_ip:172.10.100.105:2063,src_ip:20.1.100.111
   |_ coll_ip:172.10.100.106:2064,src_ip:20.1.100.111
   |_ seq_num:27 pkts/time to next template: 15/18
   |_ exported: Bytes:5040, Packets:60, Sessions:6 Flows:6
   |_ active_intf: 1
   |____ interface:port4 sample_direction:both device_index:12 snmp_index:6
```

b. Verify the sampled NetFlow traffic packet capture:

```
# diagnose sniffer packet any 'udp and port 2059 or 2060 or 2061 or 2062 or 2063 or
2064' 4

filters=[udp and port 2059 or 2060 or 2061 or 2062 or 2063 or 2064]
7.005812 port4 out 20.1.100.111.2472 -> 172.10.100.101.2059: udp 60
7.005821 port4 out 20.1.100.111.2472 -> 172.10.100.102.2060: udp 60
7.005826 port4 out 20.1.100.111.2472 -> 172.10.100.103.2061: udp 60
7.005830 port4 out 20.1.100.111.2472 -> 172.10.100.104.2062: udp 60
7.005834 port4 out 20.1.100.111.2472 -> 172.10.100.105.2063: udp 60
7.005838 port4 out 20.1.100.111.2472 -> 172.10.100.106.2064: udp 60
```

Enhance port-level control for STP and 802.1x authentication - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- Hardware switch

FortiOS now provides the capability to enable or disable Spanning Tree Protocol (STP) and 802.1x authentication on a per-port basis, granting administrators precise management over what ports necessitate STP and 802.1x.

After ports are added to a virtual switch with STP enabled, a new option is available to enable or disable STP for each member port:

```
config system interface
    edit <port>
        set stp-edge {enable | disable}
    next
end
```

set stp-edge {enable | disable}	The stp-edge option is visible when <port> is a member of config system switch-interface with a corresponding config system interface entry that has set stp enable. Specify whether the port supports STP: • enable: Enable as an STP edge port. The port does not send any STP BPDUs and ignores any STP BPDUs sent to it. • disable: Disable as an STP edge port. The port can send and receive STP.

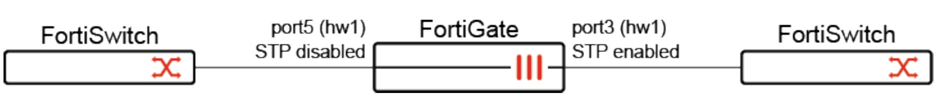
After ports are added to a virtual switch with 802.1x authentication enabled, a new option is available to enable or disable 802.1x authentication for each member port:

```
config system interface
    edit <port>
        set security-8021x-member-mode {enable | disable}
    next
end
```

set security-8021x-member-mode {enable | disable}	The security-8021x-member-mode option is visible when <port> is a member of config system switch-interface with a corresponding config system interface entry that has set security-mode 802.1X. Specify whether the port uses 802.1x authentication: • enable: Enable 802.1x authentication for the port. • disable: Disable 802.1x authentication for the port.

Example 1

In this example, FortiGate is connected to two switches, and a virtual switch named hw1 is configured with two port members: port3 and port5. STP is enabled for port3 and disabled for port5. Any STP sent to port5 is silently ignored. Port3 remains enabled for STP.



To configure STP for individual ports:

1. Configure a virtual switch to use port3 and port5:

```
config system virtual-switch
    edit "hw1"
```

```
        set physical-switch "sw0"
        config port
            edit "port3"
            next
            edit "port5"
            next
        end
    next
end
```

2. Enable STP for the virtual switch:

```
config system interface
    edit "hw1"
        set vdom "vdom1"
        set ip 6.6.6.1 255.255.255.0
        set allowaccess ping https ssh
        set type hard-switch
        set stp enable
        set device-identification enable
        set lldp-transmission enable
        set role lan
        set snmp-index 55
        set ip-managed-by-fortiipam disable
    next
end
```

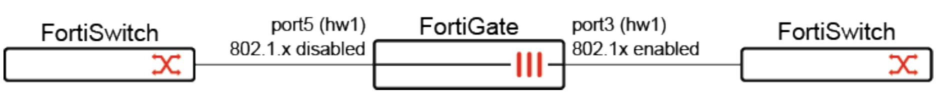
3. Disable STP on port5 by enabling it as an STP edge port:

```
config system interface
    edit "port5"
        set vdom "vdom1"
        set type physical
        set stp-edge enable
        set snmp-index 9
    next
end
```

Port5 is enabled as an edge port with STP disabled. Port3 remains enabled for STP.

Example 2

In this example, FortiGate is connected to two switches, and a virtual switch named hw1 is configured with two port members: port3 and port5. 802.1x authentication is enabled for port3 and disabled for port5.



To configure 802.1x authentication for individual ports:

1. Configure a virtual switch to use port3 and port5:

```
config system virtual-switch
    edit "hw1"
        set physical-switch "sw0"
        config port
```

```
            edit "port3"
            next
            edit "port5"
            next
        end
    next
end
```

2. Enable 802.1x authentication for the virtual switch:

```
config system interface
   edit "hw1"
        set vdom "vdom1"
        set ip 6.6.6.1 255.255.255.0
        set allowaccess ping https ssh
        set type hard-switch
        set security-mode 802.1X
        set security-groups "group_radius"
        set device-identification enable
        set lldp-transmission enable
        set role lan
        set snmp-index 55
        set ip-managed-by-fortiipam disable
    next
end
```

3. Disable 802.1x authentication on port5:

```
config system interface
    edit "port5"
        set vdom "vdom1"
        set type physical
        set security-8021x-member-mode disable
        set snmp-index 9
    next
end
```

802.1x authentication is disabled on port5 and remains enabled on port3.

IPv6

This section includes information about IPv6 related new features:

- BGP conditional advertisements for IPv6 prefix when IPv4 prefix conditions are met and vice-versa on page 127

BGP conditional advertisements for IPv6 prefix when IPv4 prefix conditions are met and vice-versa

This information is also available in the FortiOS 7.4 Administration Guide:

- BGP conditional advertisements for IPv6 prefix when IPv4 prefix conditions are met and vice-versa

BGP conditional advertisement allows the router to advertise a route only when certain conditions are met. Multiple conditions can be used together, with conditional route map entries treated as an AND operator. The FortiGate supports conditional advertisement of IPv4 and IPv6 route maps with `edit <advertise-routemap>` under `config conditional-advertise`, and supports configuring IPv4 and IPv6 route maps as conditions with the `condition-routemap` setting.

The FortiGate can cross-check conditions involving IPv4 and IPv6 route maps and perform conditional advertisements accordingly when those conditions are met. The global option, `cross-family-conditional-adv` in the BGP configuration settings allows this cross-checking to occur.

```
config router bgp
    set cross-family-conditional-adv {enable | disable}
    config conditional-advertise
        edit <advertise-routemap>
            set advertise-routemap <string>
            set condition-routemap <name1>, <name2>, ...
            set condition-type {exist | non-exist}
        next
    end
end
```

By default, the `cross-family-conditional-adv` setting is disabled. When disabled, the FortiGate will only check conditional route maps against the routing information base (RIB) of the IP address family (IPv4 or IPv6) that corresponds to the IP address family of the route map to be advertised conditionally.

For example, for an IPv6 conditional advertisement, if IPv4 conditional route maps have been configured, then the FortiGate will not meet any of these conditions because IPv4 routes will not exist in the IPv6 RIB. The same behavior applies for an IPv4 conditional advertisement, namely, that the FortiGate will not meet any configured IPv6 conditions since these routes will not exist in the IPv4 RIB. If routes do not match a conditional route map, then the condition is considered non-existent.

IPv4 and IPv6 BGP conditional advertisements using advertising and conditional route maps of the same IP address family are already supported in previous versions of FortiOS.

DS-Lite example

In this example, the FortiGate acts as a Dual-Stack Lite (DS-Lite) address family transition router (AFTR) where the customer equipment (CE) network via Router1 uses IPv6 and where Router2 is the internet gateway using IPv4.

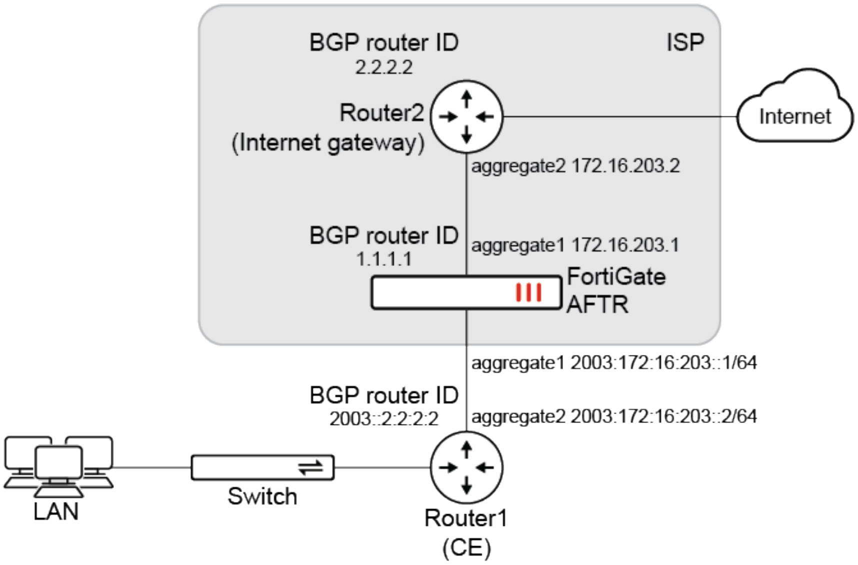
The administrator of the AFTR has the following requirements:

- The FortiGate needs to announce IPv4 pools for NAT translation towards the internet gateway only if the IPv6 B4 prefix exists in the routing table.
- The FortiGate needs to advertise the DS-Lite termination IPv6 address towards the CE network only if the IPv4 default route exists on the FortiGate.

The prefixes defined in IPv4 route map `2814` and IPv6 route map `map-281` both exist, so the FortiGate advertises the route map prefix in route-map `2224` (172.22.2.0/255.255.255.0) to its BGP neighbor 2.2.2.2.

For IPv6 neighbor 2003::2:2:2:2, the prefixes defined in IPv4 route map `2874` and IPv6 route map `map-38` both do not exist, and the `condition-type` is set to `non-exist`, so the FortiGate advertises the route map prefix in route map `map-222` (2003:172:22:1::/64) to its BGP neighbor 2003::2:2:2:2.

When the global `cross-family-conditional-adv` enabled, this is the only time the FortiGate will cross-check the address family; otherwise, it only checks the corresponding conditional map and treats the cross-family addresses as non-existent.



To configure the BGP settings with address family cross-checking:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    set ibgp-multipath enable
    set network-import-check disable
    set cluster-id 1.1.1.1
    set graceful-restart enable
    set cross-family-conditional-adv enable
    config neighbor
        edit "3.3.3.3"
            set activate6 disable
            set capability-graceful-restart enable
            set soft-reconfiguration enable
            set prefix-list-out "local-out"
            set remote-as 65412
            set route-map-out "as-prepend"
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
            set route-reflector-client enable
        next
        edit "2.2.2.2"
            set advertisement-interval 5
            set activate6 disable
            set capability-graceful-restart enable
            set soft-reconfiguration enable
            set remote-as 65412
            set keep-alive-timer 34
            set holdtime-timer 90
            set update-source "loopback1"
```

```
            config conditional-advertise
                edit "2224"
                    set condition-routemap "2814" "map-281"
                next
            end
            set route-reflector-client enable
        next
        edit "2003::2:2:2:2"
            set advertisement-interval 5
            set activate disable
            set capability-graceful-restart6 enable
            set soft-reconfiguration enable
            set soft-reconfiguration6 enable
            set remote-as 65412
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
            config conditional-advertise6
                edit "map-222"
                    set condition-routemap "map-38" "2874"
                    set condition-type non-exist
                next
            end
            set route-reflector-client6 enable
        next
        edit "2003::3:3:3:3"
            set advertisement-interval 5
            set activate disable
            set capability-graceful-restart6 enable
            set soft-reconfiguration6 enable
            set remote-as 65412
            set route-map-in6 "community-del777"
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
        next
    end
    config network
        edit 1
            set prefix 172.27.1.0 255.255.255.0
        next
        edit 2
            set prefix 172.27.2.0 255.255.255.0
        next
        edit 3
            set prefix 172.22.2.0 255.255.255.0
        next
    end
    config network6
        edit 1
            set prefix6 2003:172:22:1::/64
        next
    end
end
```

To verify the BGP status and the BGP routing table for IPv4:

```
# get router info bgp summary
VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 2
6 BGP AS-PATH entries
2 BGP community entries

Neighbor   V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2    4      65412     100     148        2    0    0 00:42:22        3
3.3.3.3    4      65412      99      99        2    0    0 00:42:05        6
6.6.6.6    4         20       0       0        0    0    0    never Idle (Admin)
10.100.1.1 4         20     100     107        2    0    0 00:43:43        2
10.100.1.5 4         20      53      57        2    0    0 00:43:42        0

Total number of neighbors 5

Condition route map:
  2814, state 1, use 3
  map-281, state 1, use 3
```

To verify the BGP status and the BGP routing table for IPv6:

```
# get router info6 bgp summary
VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 3
6 BGP AS-PATH entries
2 BGP community entries

Neighbor      V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
6.6.6.6       4         20       0       0        0    0    0    never Idle (Admin)
10.100.1.1    4         20     100     108        3    0    0 00:43:51        0
10.100.1.5    4         20      53      57        3    0    0 00:43:50        0
2003::2:2:2:2 4      65412      98     118        3    0    0 00:42:25        1
2003::3:3:3:3 4      65412     102     100        2    0    0 00:42:20        3

Total number of neighbors 5

Condition route map:
  map-38, state 0, use 3
  2874, state 0, use 3
```

To verify the BGP routing table for IPv4 and confirm the conditional advertisement occurred:

```
# get router info routing-table bgp
Routing table for VRF=0
B       172.22.2.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:00:03,
[1/0]
B       172.27.1.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:37:30,
[1/0]
B       172.27.2.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:37:30,
[1/0]
B       172.27.5.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:37:30,
[1/0]
B       172.27.6.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:37:30,
[1/0]
```

```
B       172.27.7.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:37:30,
[1/0]
B       172.27.8.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:37:30,
[1/0]
B       172.29.1.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:37:30,
[1/0]
B       172.29.2.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:37:30,
[1/0]
```

To verify the BGP routing table for IPv6 and confirm the conditional advertisement occurred:

```
# get router info6 routing-table bgp
Routing table for VRF=0
B       2003:172:22:1::/64 [200/0] via 2003::1:1:1:1 (recursive via 2003:172:16:203::1,
agg2), 00:00:01, [1024/0]
B       2003:172:28:1::/64 [200/0] via 2003::3:3:3:3 (recursive via fe80::a5b:eff:feeb:ca45,
port1), 00:37:59, [1024/0]
B       2003:172:28:2::/64 [200/0] via 2003::3:3:3:3 (recursive via fe80::a5b:eff:feeb:ca45,
port1), 00:37:59, [1024/0]
```

Behavior when address family cross-checking is disabled

Using a similar BGP configuration with `cross-family-conditional-adv` disabled, note the following behavior based on the condition type.

When the condition type is set to exist:

```
config router bgp
       set cross-family-conditional-adv disable
       config neighbor
        edit "2.2.2.2"
            config conditional-advertise
                edit "222v4"
                    set condition-routemap "4-281" "6-281"
                    set condition-type exist
                next
            end
        next
    end
end
```

The FortiGate will only check the IPv4 RIB table to see if there is a matching IP address for each route map. Any IPv6 address under the route map will not get checked in the corresponding IPv6 RIB table, and the condition result will be non-existent. The `222v4` route map will not advertise to its neighbor because the result is non-existent, while the condition type is existent.

When the condition type is set to non-exist:

```
config router bgp
       set cross-family-conditional-adv disable
       config neighbor
        edit "2003::2:2:2:2"
            config conditional-advertise6
                edit "v6-222"
                    set condition-routemap "v6-238" "v4-287"
```

```
                    set condition-type non-exist
                next
            end
        next
    end
end
```

If the `v6-238` IPv6 prefix does not exist in the IPv6 RIB table, then the FortiGate will only check `v4-287` in the IPv6 RIB table. The FortiGate will not find it because it is an IPv4 address. Since the condition type is also `non-exist`, route `v6-222` will be advertised to its neighbor.

Explicit and transparent proxy

This section includes information about explicit and transparent proxy related new features:

Changing the FTP mode from active to passive for explicit proxy

This information is also available in the FortiOS 7.4 Administration Guide:

- Changing the FTP mode from active to passive for explicit proxy

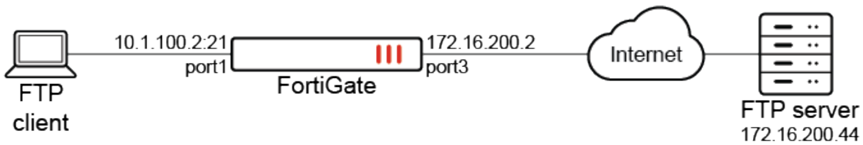
An explicit FTP proxy can convert an active FTP connection initiated by an FTP client to a passive FTP connection between the explicit FTP proxy and FTP server.

```
config ftp-proxy explicit
    set server-data-mode {client | passive}
end
```

`server-data-mode {client | passive}`	Set the data selection mode on the FTP server side: • `client`: use the same transmission mode for client and server data sessions (default). • `passive`: use passive mode for server data sessions.

Example

In this example, a client that only supports active mode FTP connects to a remote FTP server through the explicit FTP proxy to download a text file (test1.txt). The explicit FTP proxy converts the active FTP connection to a passive connection between the explicit FTP proxy and the FTP server.



To configure passive mode for FTP server data sessions:

1. Configure the web proxy:

```
config ftp-proxy explicit
    set status enable
    set incoming-port 21
    set server-data-mode passive
end
```

2. Enable the explicit FTP proxy on port1:

```
config system interface
    edit "port1"
        set ip 10.1.100.2 255.255.255.0
        set explicit-ftp-proxy enable
    next
end
```

3. Configure the firewall policy:

```
config firewall proxy-policy
    edit 1
        set proxy ftp
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
    next
end
```

4. Get the client to download the text file from the FTP server (NcFTP is used in this example):

```
ncftpget -E -r 0 -d stdout -u pc4user1@172.16.200.44 -p 123456 10.1.100.2 ./
/home/pc4user1/test1.txt
...
Cmd: PORT 10,1,100,11,151,115
200: PORT command successful. Consider using PASV.
Cmd: RETR /home/pc4user1/test1.txt
```

5. In the FTP server logs, verify that the explicit FTP proxy converted the active FTP connection to a passive connection:

```
...
2023-01-28 01:56:39,909 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching PRE_CMD command 'PASV' to mod_exec
2023-01-28 01:56:39,909 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching PRE_CMD command 'PASV' to mod_rewrite
2023-01-28 01:56:39,909 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching PRE_CMD command 'PASV' to mod_tls
2023-01-28 01:56:39,909 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching PRE_CMD command 'PASV' to mod_core
```

```
2023-01-28 01:56:39,909 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching PRE_CMD command 'PASV' to mod_core
2023-01-28 01:56:39,909 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching CMD command 'PASV' to mod_core
2023-01-28 01:56:39,909 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): in dir_check_full(): path = '/home/pc4user1', fullpath =
'/home/pc4user1'
2023-01-28 01:56:39,909 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): Entering Passive Mode (172,16,200,44,175,61).
2023-01-28 01:56:39,910 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching POST_CMD command 'PASV' to mod_exec
2023-01-28 01:56:39,910 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching LOG_CMD command 'PASV' to mod_log
2023-01-28 01:56:39,911 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching PRE_CMD command 'RETR /home/pc4user1/test1.txt' to mod_exec
```

Configuring a secure explicit proxy

This information is also available in the FortiOS 7.4 Administration Guide:

- Secure explicit proxy

Secure explicit web proxy with HTTPS connections is supported between web clients and the FortiGate.

```
config web-proxy explicit
    set secure-web-proxy {disable | enable | secure}
    set secure-web-proxy-cert <certificate1> <certificate2> ...
    set ssl-dh-bits {768 | 1024 | 1536 | 2048}
end
```

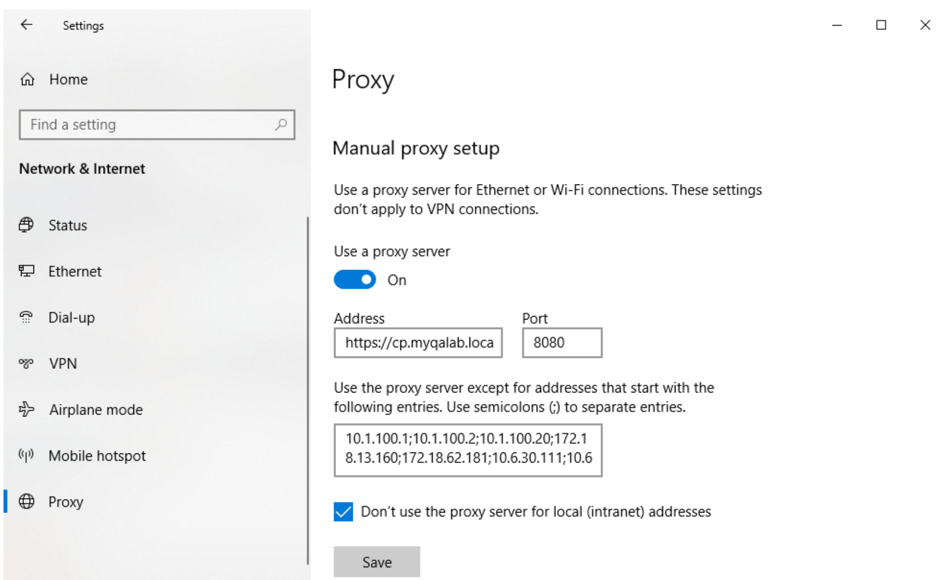
secure-web-proxy {disable | enable | secure}	Enable/disable/require the secure web proxy for HTTP and HTTPS session. • disable: disable secure web proxy (default) • enable: enable secure web proxy access, allowing both HTTPS and HTTP connections to the explicit proxy • secure: require secure web proxy access, allowing only HTTPS connections to the explicit proxy
secure-web-proxy-cert <certificate1> <certificate2> ...	Enter the names of the server certificates in the local certificate store of the FortiGate used to establish a TLS connection between the user's browser and the FortiGate. Multiple server certificates can be configured, and different certificate types can be used. The SNI tries to match the right server certificate for the connection. If the SNI cannot not match with the certificates' CN or SAN, the first server certificate will be offered.
ssl-dh-bits {768 | 1024 | 1536 | 2048}	Set the bit size of Diffie-Hellman (DH) prime used in the DHE-RSA negotiation. • 768: use 768-bit Diffie-Hellman prime • 1024: use 1024-bit Diffie-Hellman prime • 1536: use 1536-bit Diffie-Hellman prime • 2048: use 2048-bit Diffie-Hellman prime (default)

Example

In this example, a Windows PC user configures an HTTPS URL (https://cp.myqalab.local) as the proxy address for the explicit web proxy. When the user opens a browser (such as Edge or Chrome), the browser will use the HTTPS URL to connect to the explicit web proxy and send any HTTP requests to the proxy over HTTPS. The certificate (server_cert) contains the explicit web proxy's name (cp.myqalab.local) as its CN, so the browser will accept this certificate for the TLS connection.

To configure the Windows proxy settings:

1. On the Windows PC, go to *Settings > Network & Internet > Proxy*.
2. In the *Manual proxy setup* section configure the following:
   a. Enable *Use a proxy server*.
   b. Set the *Address* to *https://cp.myqalab.local*.
   c. Set the *Port* to *8080*.
   d. If needed, enter any addresses to exempt in the text box (use a semicolon to separate entries).
   e. Enable *Don't use the proxy server for local (intranet) addresses*.



3. Click *Save*.

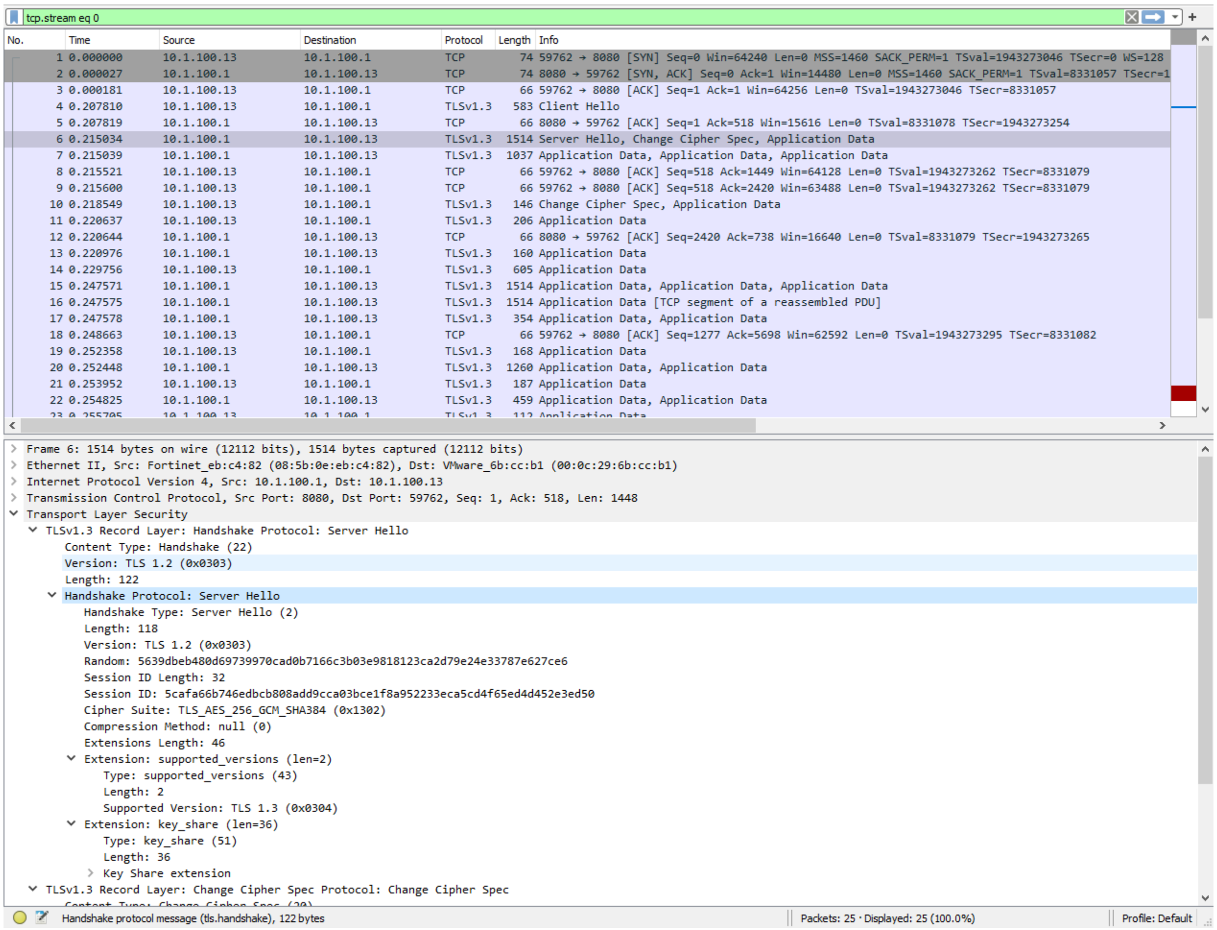
To configure the secure explicit web proxy:

```
config web-proxy explicit
    set status enable
    set secure-web-proxy enable
    set ftp-over-http enable
    set socks enable
```

```
    set http-incoming-port 8080
    set secure-web-proxy-cert "server_cert"
    set socks-incoming-port 1080
    set ipv6-status enable
    set unknown-http-version best-effort
    set pac-file-server-status enable
    set pac-file-data "function FindProxyForURL(url, host) {
   // testtest
   return \"PROXY 10.1.100.1:8080\";
}
"
    set pac-file-through-https enable
end
```

To verify the TLS connection:

1. Perform a packet capture of HTTPS traffic between the web client and the web server. Wireshark is used in this example.
2. Locate the exchange between the web client (10.1.100.13) and the explicit web proxy (10.1.100.1:8080):



After the client initiates the TLS connection to the explicit web proxy with a client hello packet, the web proxy is able to respond appropriately with a server hello packet to establish a TLS connection first before any HTTP messages are exchanged, and all HTTP messages will be protected by the TLS connection.

Explicit proxy logging enhancements

This information is also available in the FortiOS 7.4 Administration Guide:

- Explicit proxy logging

Explicit proxy traffic logging has been enhanced to improve troubleshooting the HTTP proxy status for each HTTP transaction with the following:

- Support monitoring HTTP header requests and responses in the UTM web filter log. This requires an SSL deep inspection profile to be configured in the corresponding firewall policy.
- Support logging the explicit web proxy forward server name using `set log-forward-server`, which is disabled by default.

```
config web-proxy global
    set log-forward-server {enable | disable}
end
```

- Support logging TCP connection failures in the traffic log when a client initiates a TCP connection to a remote host through the FortiGate, and the remote host is unreachable.

Basic configuration

The following FortiGate configuration is used in the three explicit proxy traffic logging use cases in this topic.

To configure the FortiGate:

1. Configure the web proxy profile:

```
config web-proxy profile
    edit "header"
        config headers
            edit 1
                set name "test_request_header"
                set action monitor-request
            next
            edit 2
                set name "ETag"
                set action monitor-response
            next
        end
    next
end
```

2. Enable forward server name logging in traffic:

```
config web-proxy global
    set proxy-fqdn "100D.qa"
    set log-forward-server enable
end
```

3. Configure the web filter banned word table to block any HTTP response containing the text, `works`:

```
config webfilter content
    edit 1
        set name "default"
        config entries
            edit "works"
                set status enable
                set action block
            next
        end
    next
end
```

4. Configure the web filter profile:

```
config webfilter profile
    edit "header"
        set feature-set proxy
        config web
            set bword-table 1
        end
        config ftgd-wf
            unset options
        end
        set log-all-url enable
        set extended-log enable
        set web-extended-all-action-log enable
    next
end
```

5. Configure the web proxy forwarding server:

```
config web-proxy forward-server
    edit "fgt-b"
        set ip 172.16.200.20
    next
end
```

6. Configure the firewall policy:

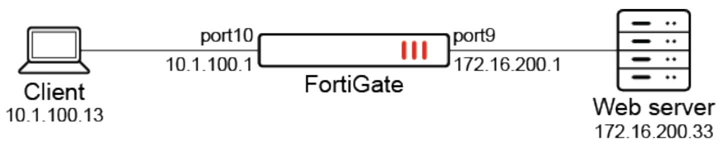
```
config firewall policy
    edit 1
        set srcintf "port10"
        set dstintf "port9"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set webproxy-profile "header"
        set webproxy-forward-server "fgt-b"
        set ssl-ssh-profile "deep-inspection"
        set webfilter-profile "header"
        set logtraffic all
        set nat enable
    next
end
```

A firewall policy is used in this basic configuration example and the specific examples that follow. This feature also works for the explicit web proxy or transparent web proxy with proxy policies, and the configurations are similar:

- Example 1: apply the `web-proxy profile` and `webfilter profile` to the proxy policy.
- Example 2: apply the `webproxy-forward-server`.

Example 1: monitoring HTTP header requests

In this example, the user wants to monitor some HTTP headers in HTTP messages forwarded through a FortiGate proxy (either transparent or explicit proxy with a firewall policy in proxy mode or a proxy policy). When the monitored headers are detected, they will be logged in the UTM web filter log.



In the web proxy profile configuration, the following HTTP headers are monitored:

- `test_request_header`: this is a user-customized HTTP header.
- `ETag`: this is a HTTP header returned by the web server's 200 OK response.

Based on the web filter profile configuration, the monitored headers in the web proxy profile will only be logged when the HTTP response received by the FortiGate triggers a block action by the banned word table. The `log-all-url`, `extended-log`, and `web-extended-all-action-log` settings in the web filter profile must be enabled.

The following settings are required in the firewall policy:

- `set inspection-mode proxy`
- `set webproxy-profile "header"`
- `set ssl-ssh-profile "deep-inspection"`
- `set webfilter-profile "header"`
- `set logtraffic all`

To verify the configuration:

1. Send a HTTP request from the client:

   ```
   curl -kv https://172.16.200.33 -H "test_request_header: aaaaa"
   ```

   This command sends a HTTP request with the header `test_request_header: aaaaa` through the FortiGate. Since the response from the web server contains the word `works`, the response will be blocked by the web filter profile (`header`). During this process, two logs will be generated.
2. On the FortiGate, check the traffic logs:

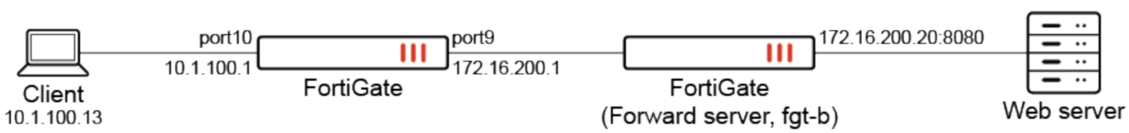
   ```
   # execute log filter category 3
   1: date=2023-04-19 time=19:01:19 eventtime=1681956079146481995 tz="-0700"
   logid="0314012288" type="utm" subtype="webfilter" eventtype="content" level="warning"
   vd="vdom1" policyid=1 poluuid="4d8dc396-46e3-51ea-7f3f-ee328a5bd07b" policytype="policy"
   sessionid=40980 srcip=10.1.100.13 srcport=54512 srccountry="Reserved" srcintf="port10"
   srcintfrole="undefined" srcuuid="6ce0b8ca-30ae-51ea-a388-ceacbb4fb045"
   ```

```
dstip=172.16.200.33 dstport=443 dstcountry="Reserved" dstintf="port9"
dstintfrole="undefined" dstuuid="6ce0b8ca-30ae-51ea-a388-ceacbb4fb045" proto=6
httpmethod="GET" service="HTTPS" hostname="172.16.200.33" agent="curl/7.61.1"
profile="header" reqtype="direct" url="https://172.16.200.33/" sentbyte=0 rcvdbyte=0
direction="incoming" action="blocked" banword="works" msg="URL was blocked because it
contained banned word(s)." rawdata="[REQ] test_request_header:=aaaaa||[RESP] Content-
Type=text/html|ETag=\"34-5b23b9d3b67f4\""

2: date=2023-04-19 time=19:01:19 eventtime=1681956079144896978 tz="-0700"
logid="0319013317" type="utm" subtype="webfilter" eventtype="urlmonitor" level="notice"
vd="vdom1" policyid=1 poluuid="4d8dc396-46e3-51ea-7f3f-ee328a5bd07b" policytype="policy"
sessionid=40980 srcip=10.1.100.13 srcport=54512 srccountry="Reserved" srcintf="port10"
srcintfrole="undefined" srcuuid="6ce0b8ca-30ae-51ea-a388-ceacbb4fb045"
dstip=172.16.200.33 dstport=443 dstcountry="Reserved" dstintf="port9"
dstintfrole="undefined" dstuuid="6ce0b8ca-30ae-51ea-a388-ceacbb4fb045" proto=6
httpmethod="GET" service="HTTPS" hostname="172.16.200.33" agent="curl/7.61.1"
profile="header" action="passthrough" reqtype="direct" url="https://172.16.200.33/"
sentbyte=724 rcvdbyte=2769 direction="outgoing" msg="URL has been visited"
ratemethod="ip" cat=255 rawdata="[REQ] test_request_header:=aaaaa"
```

Log 1 is for the blocked HTTP response that contains both monitored headers, `test_request_header` and `ETag`, and their values, `aaaaa` and `34-5b23b9d3b67f4`, respectively. Log 2 is for the HTTP request passing through the FortiGate proxy that contains `test_request_header` and its `aaaaa` value in the `rawdata` field.

Example 2: logging the explicit web proxy forward server name

In this example, the user wants to see the name of the web proxy forward server in the traffic log when the traffic is forwarded by a web proxy forward server.



In the global web proxy settings, `log-forward-server` must be enabled.

The following settings are required in the firewall policy:

- `set inspection-mode proxy`
- `set webproxy-forward-server "fgt-b"`
- `set logtraffic all`

When a HTTP request is sent through the FortiGate proxy, the request will be forwarded by the FortiGate to the upstream proxy (fgt-b), and the forward server's name will be logged in the traffic log.

To verify the configuration:

1. Send a HTTP request from the client:

```
curl -kv https://www.google.com
```

2. On the FortiGate, check the traffic logs:

```
# execute log filter category 3
1: date=2023-04-19 time=19:51:33 eventtime=1681959093510003961 tz="-0700"
logid="0000000013" type="traffic" subtype="forward" level="notice" vd="vdom1"
```

```
srcip=10.1.100.13 srcport=49762 srcintf="port10" srcintfrole="undefined"
dstip=142.250.217.100 dstport=443 dstintf="port9" dstintfrole="undefined"
srccountry="Reserved" dstcountry="United States" sessionid=43292 proto=6 action="client-
rst" policyid=1 policytype="policy" poluuid="4d8dc396-46e3-51ea-7f3f-ee328a5bd07b"
service="HTTPS" trandisp="snat" transip=172.16.200.1 transport=49762 duration=120
sentbyte=0 rcvdbyte=37729 sentpkt=0 rcvdpkt=33 appcat="unscanned" wanin=3779 wanout=682
lanin=879 lanout=36005 fwdsrv="fgt-b" utmaction="block" countssl=1 utmref=65506-14
```

Example 3: logging TCP connection failures

In this example, a client initiates a TCP connection to a remote network node through the FortiGate. The connection fails because the IP address or port of the remote node is unreachable. A `Connection Failed` message appears in the logs. In the firewall policy configuration, the `inspection-mode` can be set to either `proxy` or `flow` mode.

Based on the basic FortiGate configuration used in examples 1 and 2, the forward server may need to be removed from the firewall policy if the forward server's TCP IP port is actually reachable. If the forward server proxy tries to set up back-to-back TCP connections with the downstream FortiGate and the remote server as in the case of deep-inspection, then when the client tries to connect to a remote node (even if the IP address or port is unreachable), the downstream FortiGate is able to establish a TCP connection with the upstream forward server, so there will be no `Connection Failed` message in the downstream FortiGate's log.

Currently, the `Connection Failed` message in the downstream FortiGate's log is visible for the case when there is an unreachable TCP port only when explicit web proxy with a proxy policy is configured. Therefore, the following example that makes use of a firewall policy demonstrates this log message is only supported for the unreachable IP address case.

To verify the configuration:

1. Send a HTTP request from the client to an unreachable IP:

```
curl -kv https://172.16.200.34
```

2. On the FortiGate, check the traffic logs:

```
# execute log filter category 3
1: date=2023-04-19 time=20:25:55 eventtime=1681961155100007061 tz="-0700"
logid="0000000013" type="traffic" subtype="forward" level="notice" vd="vdom1"
srcip=10.1.100.13 srcport=52452 srcintf="port10" srcintfrole="undefined"
dstip=172.16.200.34 dstport=443 dstintf="port9" dstintfrole="undefined"
srccountry="Reserved" dstcountry="Reserved" sessionid=44903 proto=6 action="server-rst"
policyid=1 policytype="policy" poluuid="4d8dc396-46e3-51ea-7f3f-ee328a5bd07b"
service="HTTPS" trandisp="snat" transip=172.16.200.1 transport=52452 duration=20
sentbyte=180 rcvdbyte=164 sentpkt=3 rcvdpkt=3 appcat="unscanned" wanin=0 wanout=0
lanin=0 lanout=0 crscore=5 craction=262144 crlevel="low" msg="Connection Failed"
```

Support the Happy Eyeballs algorithm for explicit proxy - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:
- Configuring fast fallback for explicit proxy

The "Happy Eyeballs" (also named fast fallback) algorithm, as outlined in RFC 8305, is now supported for explicit web proxy. This feature operates by attempting to connect to a web server that is available at multiple IPv4 and IPv6 addresses, either sequentially or simultaneously. As a result, the web server can be connected with reduced user-visible delay, which enhances the overall browsing experience.

```
config web-proxy fast-fallback
    edit <name>
        set status {enable | disable}
        set connection-mode {sequentially | simultaneously}
        set protocol {IPv4-first | IPv6-first | IPv4-only | IPv6-only}
        set connection-timeout <integer>
    next
end
```

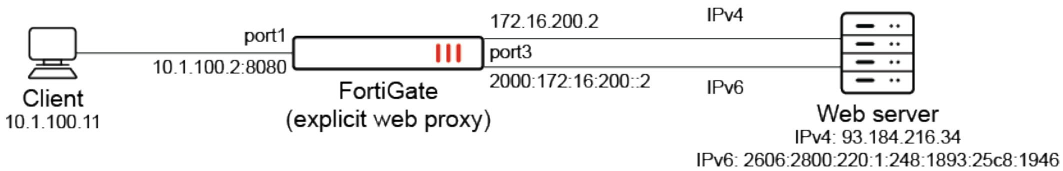
`status {enable | disable}`	Enable/disable the fast fallback entry (default = `enable`).
`connection-mode {sequentially | simultaneously}`	Set the connection mode for multiple destinations. • `sequentially`: connect the different destinations sequentially (default). • `simultaneously`: connect the different destinations simultaneously.
`protocol {IPv4-first | IPv6-first | IPv4-only | IPv6-only}`	Set the connection protocols for multiple destinations. • `IPv4-first`: connect IPv4 destinations first (default). • `IPv6-first`: connect IPv6 destinations first. • `IPv4-only`: connect IPv4 destinations only. • `IPv6-only`: connect IPv6 destinations only.
`connection-timeout <integer>`	Start another connection if a connection takes longer than the timeout value, in milliseconds (200 - 1800000, default = 200).

Based on the settings for `connection-mode` and `protocol`, the explicit web proxy will try connecting to the web server in different ways:

- If the `connection-mode` is set to `sequential` (default), then the explicit web proxy will try connecting to the web server by IPv4 first, or by IPv6 first depending on the `protocol` setting. If the connection attempt over IPv4 or IPv6 succeeds, then the connection is kept; but if the connection fails, then it falls back to try a connection over IPv6 or IPv4 instead.
- If the `connection-mode` is set to `simultaneously`, then the explicit web proxy will try connecting to the web server by IPv4 and IPv6 at the same time. If the connection over IPv4 is established first, then the connection is kept for the session and the IPv6 connection is discarded and vice-versa.
- If the user only wants to connect by IPv4 but not IPv6, or by IPv6 but not IPv4, then the `protocol` option can be set to `IPv4-only` or `IPv6-only` accordingly. The explicit web proxy will try connecting to the web server only by IPv4 or IPv6, even though both IPv4 and IPv6 may work.

Example

In this example, a client visits a web server through a FortiGate explicit web proxy that has IPv4 and IPv6 connections to the web server (www.example.com), which can resolve to IPv4 address 93.184.216.34 and IPv6 address 2606:2800:220:1:248:1893:25c8:1946.



The configuration uses sequential connection mode, the IPv4 first protocol, and the default connection timeout (200 ms).

To configure the FortiGate:

1. Configure the IPv4 static route:

```
config router static
    edit 1
        set gateway 172.16.200.251
        set device "port3"
    next
end
```

2. Configure the IPv6 static route:

```
config router static6
    edit 1
        set gateway 2000:172:16:200::254
        set device "port3"
    next
end
```

3. Configure the proxy destination connection fast fallback:

```
config web-proxy fast-fallback
    edit "ffbk"
        set status enable
        set connection-mode sequentially
        set protocol IPv4-first
        set connection-timeout 200
    next
end
```

4. Configure the exempt URL of the web server from web proxy forwarding and caching:

```
config web-proxy url-match
    edit "ffbk"
        set url-pattern "example.com"
        set fast-fallback "ffbk"
    next
end
```

5. Configure the proxy policy:

```
config firewall proxy-policy
    edit 1
```

```
        set proxy explicit-web
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set srcaddr6 "all"
        set dstaddr6 "all"
        set utm-status enable
        set ssl-ssh-profile "deep-custom"
        set av-profile "av"
    next
end
```

Verifying the connection

Scenario 1:

The TCP connection from the explicit web proxy to the web server is established successfully over IPv4 within 200 ms.

As shown in the forward traffic log, the web session data is transmitted over IPv4 between the explicit web proxy and the web server.

```
2: date=2023-06-26 time=18:46:18 eventtime=1687830378260927765 tz="-0700" logid="0000000010"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.11 srcport=33304
srcintf="port1" srcintfrole="undefined" dstcountry="United States" srccountry="Reserved"
dstip=93.184.216.34 dstport=80 dstintf="port3" dstintfrole="undefined" sessionid=1688881487
service="HTTP" proxyapptype="web-proxy" proto=6 action="accept" policyid=1
policytype="proxy-policy" poluuid="560d8520-fa7b-51ed-e06a-df05ec145542" trandisp="snat"
transip=0.0.0.0 transport=0 duration=0 wanin=0 rcvdbyte=0 wanout=0 lanin=131 sentbyte=131
lanout=1591 appcat="unscanned"
```

Scenario 2:

The TCP connection from the explicit web proxy to the web server is not established over IPv4 within 200 ms and falls back to IPv6 successfully.

The IPv4 path to the server is interrupted, and the TCP connection between the explicit web proxy and web server cannot be established. The explicit web proxy waits until the 200 ms connection timeout timer expires, then attempts to connect to the server by IPv6, which is successful. The web session data is transmitted over IPv6, as shown in the forward traffic log.

```
2: date=2023-06-26 time=18:47:27 eventtime=1687830447277653089 tz="-0700" logid="0000000010"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.11 srcport=36636
srcintf="port1" srcintfrole="undefined" dstcountry="United States" srccountry="Reserved"
dstip=2606:2800:220:1:248:1893:25c8:1946 dstport=80 dstintf="port3" dstintfrole="undefined"
sessionid=1688881488 service="HTTP" proxyapptype="web-proxy" proto=6 action="accept"
policyid=1 policytype="proxy-policy" poluuid="560d8520-fa7b-51ed-e06a-df05ec145542"
trandisp="snat" transport=0 duration=1 wanin=0 rcvdbyte=0 wanout=0 lanin=131 sentbyte=131
lanout=1591 appcat="unscanned"
```

Support webpages to properly display CORS content in an explicit proxy environment - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Display CORS content in an explicit proxy environment

Webpages can display Cross-Origin Resource Sharing (CORS) content in an explicit proxy environment when using session-based, cookie-enabled, and captive portal assisted authentication. This ensures that webpages are displayed correctly and improves the user experience.

```
config authentication rule
    edit <name>
        set web-auth-cookie enable
        set cors-stateful {enable | disable}
        set cors-depth <integer>
    next
end
```

`cors-stateful {enable | disable}`	Enable/disable allowing CORS access (default = disable). This setting is only available when `web-auth-cookie` is enabled.
`cors-depth <integer>`	Set the depth to allow CORS access (1 - 8, default = 3). For example, when visiting domain A, the returned web page may refer the browser to a cross-origin domain B (depth of 1). When the browser visits domain B, the returned web content may further refer the browser to another cross-origin domain C (depth of 2).

Example

CORS access is enabled in this example. When a user access the Microsoft *Sign in* page using an explicit proxy, the page appears and the user can log in. This example assumes the web proxy and user group have already been configured, and that the proxy captive portal setting has been enabled on the appropriate interface.

To view CORS content in an explicit proxy environment:

1. Configure the authentication scheme:

```
config authentication scheme
    edit "form"
        set method form
        set user-database "local-user-db"
    next
end
```

2. Configure the authentication rule:

```
config authentication rule
    edit "form"
        set srcaddr "all"
        set ip-based disable
```

```
        set active-auth-method "form"
        set web-auth-cookie enable
        set cors-stateful enable
        set cors-depth 3
    next
end
```

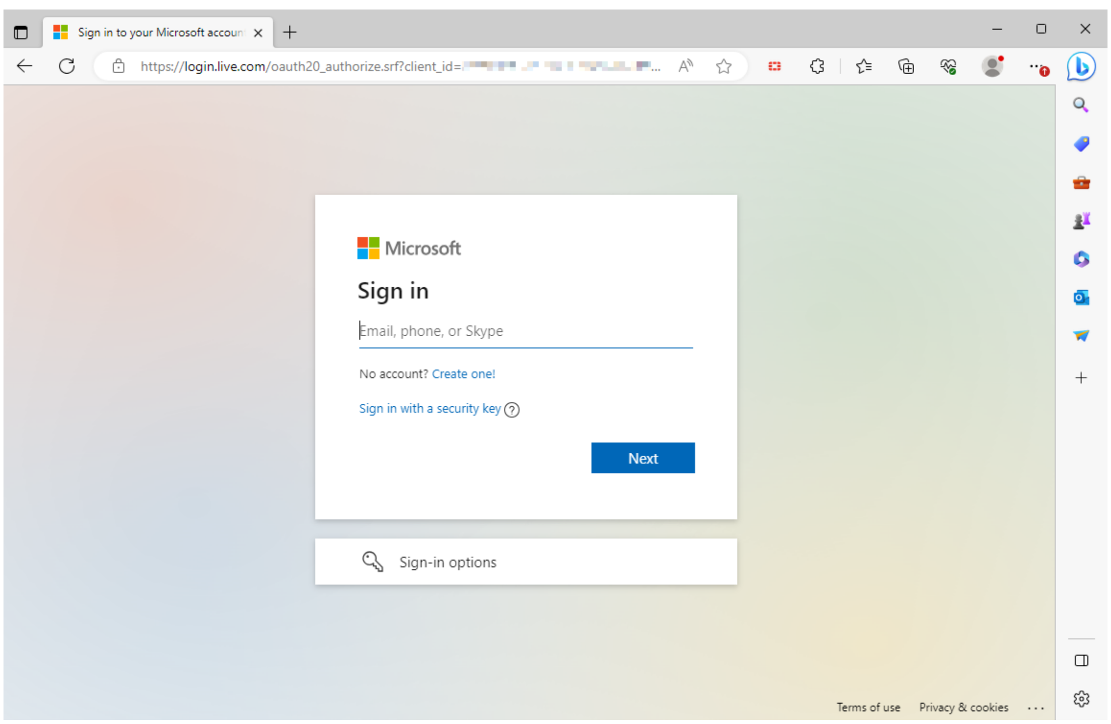
3. Configure the captive portal:

```
config authentication setting
    set captive-portal "fgt9.myqalab.local"
end
```

4. Configure the proxy policy:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set groups "localgroup"
        set utm-status enable
        set ssl-ssh-profile "deep-custom"
        set av-profile "av"
    next
end
```

5. Get a user to access login.microsoftonline.com trough the explicit web proxy. The *Sign in* page appears, and the user can log in.



If CORS access (`cors-stateful`) was disabled, the browser would load a blank page.

Forward HTTPS requests to a web server without the need for an HTTP CONNECT message - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

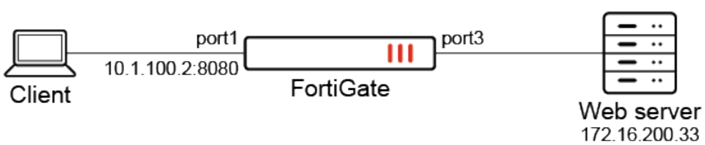
- Forward HTTPS requests to a web server without the need for an HTTP CONNECT message

An explicit web proxy can forward HTTPS requests to a web server without the need for an HTTP CONNECT message. The FortiGate explicit web proxy can be configured to detect the HTTPS scheme in the request line of a plain text HTTP request and forward it as an HTTPS request to the web server. This allows applications that cannot use the CONNECT message for sending an HTTPS request to communicate with the web server through an explicit web proxy.

```
config firewall proxy-policy
    edit <id>
        set detect-https-in-http-request {enable | disable}
    next
end
```

Example

Based on the following topology, an HTTPS request is sent to a web server through an explicit web proxy.



To enable detection of HTTPS in an HTTP request:

1. Configure the explicit web proxy:

```
config web-proxy explicit
    set status enable
    set ftp-over-http enable
    set socks enable
    set http-incoming-port 8080
    set ipv6-status enable
    set unknown-http-version best-effort
end
```

2. Enable the explicit web proxy on port1:

```
config system interface
    edit "port1"
        set ip 10.1.100.2 255.255.255.0
        set explicit-web-proxy enable
    next
end
```

3. Configure the proxy policy:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set utm-status enable
        set ssl-ssh-profile "deep-inspection"
        set av-profile "av"
        set detect-https-in-http-request enable
    next
end
```

An SSL-SSH profile with deep inspection must be applied in order to decrypt the server response in HTTPS and forward the response to the client by HTTP.

4. Using Telnet, send an HTTP request with an HTTPS scheme as follows:

```
telnet 10.1.100.2 8080
Trying 10.1.100.2...
Connected to 10.1.100.2.
Escape character is '^]'.
POST https://172.16.200.33/ HTTP/1.1
Host: 172.16.200.33
User-Agent: curl/7.68.0
Accept: */*
Content-Type: application/x-www-form-urlencoded
Content-Length: 0

HTTP/1.1 200 OK
```

5. Verify the traffic log. The HTTP request is forwarded to the server successfully by HTTPS:

```
# execute log filter category 3
...
2: date=2023-07-31 time=16:02:22 eventtime=1690844541296891542 tz="-0700"
logid="0000000010" type="traffic" subtype="forward" level="notice" vd="vdom1"
srcip=10.1.100.11 srcport=46074 srcintf="port1" srcintfrole="undefined"
dstcountry="Reserved" srccountry="Reserved" dstip=172.16.200.33 dstport=443
dstintf="port3" dstintfrole="undefined" sessionid=1799884153 service="HTTPS"
proxyapptype="web-proxy" proto=6 action="accept" policyid=1 policytype="proxy-policy"
poluuid="73379360-2d21-51ee-77d8-154efc517a6a" trandisp="snat" transip=172.16.200.2
transport=2713 duration=4 wanin=3053 rcvdbyte=3053 wanout=757 lanin=169 sentbyte=169
lanout=279 appcat="unscanned"
```

Support web proxy forward server over IPv6 - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Transparent web proxy forwarding over IPv6

The new IPv6-enabled forward server works the same way as the previous IPv4 forward server. For example, you can configure an IPv6 address or an FQDN that resolves to an IPv6 address for the forward server, and you can also use the IPv6 forward server in a forward server group.

```
config web-proxy forward-server
    edit <name>
        set addr-type {ip | ipv6 | fqdn}
        set ipv6 <IPv6-address>
    next
end
```

`addr-type`	Specify the type of IP address for the web proxy forward server: • `ip`: use an IPv4 address. • `ipv6`: use an IPv6 address. • `fqdn`: use a fully qualified domain name (FQDN).
`ipv6`	Specify the IPv6 address for the web proxy forward server. Available when `addr-type` is set to `ipv6`.

Example

In this example, an explicit web proxy with a forward server can be reached by an IPv6 address, and a client PC uses this explicit web proxy forward server to access a website, such as www.google.com.

The IPv6 address is configured for the web proxy forward server, and then the configuration is added to a proxy policy. The web proxy forward server configuration could also be added to a proxy mode policy or a transparent web proxy policy.

To configure an IPv6 address:

1. Configure an IPv6 address for the web proxy forward server.
   In this example, address type is set to IPv6, and an IPv6 address is specified in a configuration (fgt6) for a web proxy forward server.

```
config web-proxy forward-server
    edit "fgt6"
        set addr-type ipv6
        set ipv6 2000:172:16:200::8
        set port 8080
    next
end
```

2. Add the web proxy forward server to a proxy policy.
   The web proxy forward server configuration (fgt6) is added to the firewall proxy policy.

```
config firewall proxy-policy
    edit 1
        set uuid 560d8520-fa7b-51ed-e06a-df05ec145542
        set proxy explicit-web
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
```

```
        set action accept
        set schedule "always"
        set logtraffic all
        set srcaddr6 "all"
        set dstaddr6 "all"
        set webproxy-forward-server "fgt6"
        set utm-status enable
        set ssl-ssh-profile "deep-custom"
        set av-profile "av"
    next
end
```

3. View the traffic logs.

   An HTTP request to www.google.com was sent through the web proxy forward server over IPv6.

```
12: date=2023-08-10 time=23:44:43 eventtime=1691736283529768562 tz="-0700"
logid="0000000010" type="traffic" subtype="forward" level="notice" vd="vdom1"
srcip=2000:10:1:100::11 srcport=44190 srcintf="port1" srcintfrole="undefined"
dstcountry="United States" srccountry="Reserved" dstip=2607:f8b0:400a:807::2004
dstport=80 dstintf="port3" dstintfrole="undefined" sessionid=391251274 service="HTTP"
proxyapptype="web-proxy" proto=6 action="accept" policyid=1 policytype="proxy-policy"
poluuid="560d8520-fa7b-51ed-e06a-df05ec145542" trandisp="snat+dnat"
tranip=2000:172:16:200::8 tranport=8080 transip=2000:172:16:200::2 transport=21344
duration=22 wanin=2385 rcvdbyte=2385 wanout=369 lanin=129 sentbyte=129 lanout=795
appcat="unscanned"
```

SD-WAN

This section includes information about SD-WAN related new features:

Overlays and underlays

This section includes information about overlay and underlay related new features:

Using a single IKE elector in ADVPN to match all SD-WAN control plane traffic

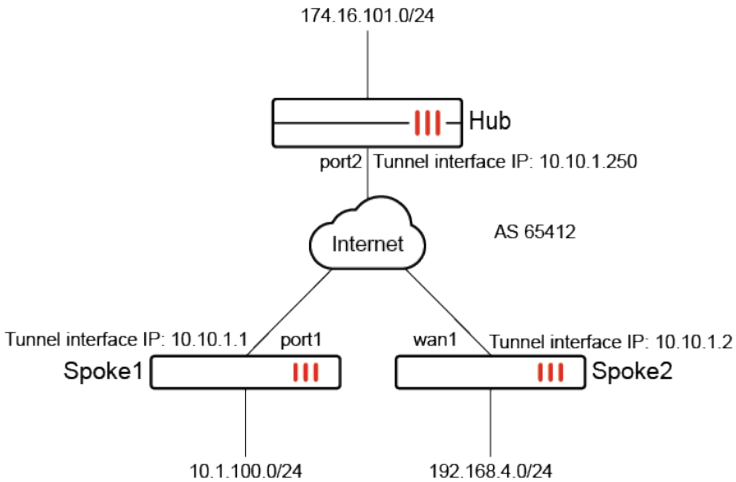
This information is also available in the FortiOS 7.4 Administration Guide:

- Using a single IKE elector in ADVPN to match all SD-WAN control plane traffic

In the SD-WAN with ADVPN use case, two spokes can communicate with each other on the control plane by an ADVPN shortcut. In order to separate the control traffic from data traffic, the IKE creates a dynamic selector for health check packets sent between the spokes. BGP traffic is also matched by this dynamic IKE selector. Therefore, when spokes establish BGP peering with other spokes, the BGP traffic does not count towards the data traffic and will not impact IPsec idle timeout and shortcut tunnel tear down.

Example

In this example, SD-WAN with ADVPN is configured. The IPsec ADVPN shortcut tunnel is required to tear down when it is idle. SD-WAN health checks are configured, and BGP neighbors established between the spokes is required.



To configure the Hub FortiGate:

1. Configure the phase 1 interface:

```
config vpn ipsec phase1-interface
    edit "Hub"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dpd on-idle
        set auto-discovery-sender enable
        set psksecret ************
        set dpd-retryinterval 60
    next
end
```

2. Configure the phase 2 interface:

```
config vpn ipsec phase2-interface
    edit "Hub"
        set phase1name "Hub"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
    next
end
```

3. Configure the VPN interface:

```
config system interface
    edit "Hub"
        set vdom "root"
        set ip 10.10.1.250 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 10.10.1.254 255.255.255.0
```

```
        set snmp-index 50
        set interface "port2"
    next
end
```

4. Configure the BGP settings:

```
config router bgp
    set as 65412
    config neighbor
        edit "10.10.1.1"
            set advertisement-interval 0
            set remote-as 65412
            set route-reflector-client enable
        next
        edit "10.10.1.2"
            set advertisement-interval 0
            set remote-as 65412
            set route-reflector-client enable
        next
    end
    config network
        edit 1
            set prefix 174.16.101.0 255.255.255.0
        next
    end
end
```

To configure the Spoke1 FortiGate:

1. Configure the phase 1 interface:

```
config vpn ipsec phase1-interface
    edit "Spoke1"
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set npu-offload disable
        set idle-timeout enable
        set idle-timeoutinterval 5
        set auto-discovery-receiver enable
        set remote-gw 172.16.200.4
        set psksecret ************
    next
end
```

2. Configure the phase 2 interface:

```
config vpn ipsec phase2-interface
    edit "Spoke1"
        set phase1name "Spoke1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
```

```
    next
end
```

3. Configure the VPN interface:

```
config system interface
    edit "Spoke1"
        set vdom "root"
        set ip 10.10.1.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 10.10.1.254 255.255.255.0
        set snmp-index 28
        set interface "port1"
    next
end
```

4. Configure the BGP settings:

```
config router bgp
    set as 65412
    config neighbor
        edit "10.10.1.250"
            set advertisement-interval 0
            set remote-as 65412
        next
        edit "10.10.1.2"
            set remote-as 65412
        next
    end
    config network
        edit 1
            set prefix 10.1.100.0 255.255.255.0
        next
    end
end
```

5. Configure the SD-WAN settings:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "Spoke1"
        next
    end
    config health-check
        edit "1"
            set server "174.16.101.44"
            set members 0
        next
    end
end
```

To configure the Spoke2 FortiGate:

1. Configure the phase 1 interface:

```
config vpn ipsec phase1-interface
    edit "Spoke2"
        set interface "wan1"
        set ike-version 2
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set npu-offload disable
        set idle-timeout enable
        set idle-timeoutinterval 5
        set auto-discovery-receiver enable
        set remote-gw 172.16.200.4
        set psksecret ************
    next
end
```

2. Configure the phase 2 interface:

```
config vpn ipsec phase2-interface
    edit "Spoke2"
        set phase1name "Spoke2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
    next
end
```

3. Configure the VPN interface:

```
config system interface
    edit "Spoke2"
        set vdom "root"
        set ip 10.10.1.2 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 10.10.1.254 255.255.255.0
        set snmp-index 15
        set interface "wan1"
    next
end
```

4. Configure the BGP settings:

```
config router bgp
    set as 65412
    config neighbor
        edit "10.10.1.250"
            set advertisement-interval 0
            set remote-as 65412
        next
        edit "10.10.1.1"
            set remote-as 65412
        next
```

```
    end
    config network
        edit 1
            set prefix 192.168.4.0 255.255.255.0
        next
    end
end
```

5. Configure the SD-WAN settings:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "Spoke2"
        next
    end
    config health-check
        edit "1"
            set server "174.16.101.44"
            set members 0
        next
    end
end
```

To verify the configuration:

1. Send traffic between the spokes to establish the ADVPN shortcut.
2. Verify the IPsec tunnel state on the Spoke1 FortiGate:

```
Spoke1 # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=Spoke1_0 ver=2 serial=7 172.16.200.1:0->172.16.200.3:0 tun_id=10.10.1.2 tun_
id6=::10.0.0.3 dst_mtu=1500 dpd-link=on weight=1
bound_if=19 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/66224 options
[102b0]=create_dev rgwy-chg frag-rfc  role=primary accept_traffic=1 overlay_id=0

parent=Spoke1 index=0
proxyid_num=2 child_num=0 refcnt=6 ilast=0 olast=0 ad=r/2
stat: rxp=0 txp=1 rxb=0 txb=40
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=Spoke1 proto=0 sa=1 ref=5 serial=2 adr health-check
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.1.2-10.10.1.2:0
  SA:  ref=3 options=92626 type=00 soft=0 mtu=1438 expire=43055/0B replaywin=2048
       seqno=214 esn=0 replaywin_lastseq=00000213 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43189/43200
  dec: spi=17a473be esp=aes key=16 40dfada9532cefe5563de71ac5908aa1
       ah=sha1 key=20 36e967d9b6fce8807132c3923d0edfae6cb6c115
```

```
  enc: spi=75cde30a esp=aes key=16 9bf08196d6830455a75bc676e04c816f
       ah=sha1 key=20 638db13dc4db0a6e5f523047805d18413eea4d4d
  dec:pkts/bytes=1060/42958, enc:pkts/bytes=1062/77075
  npu_flag=00 npu_rgwy=172.16.200.3 npu_lgwy=172.16.200.1 npu_selid=c dec_npuid=0 enc_
npuid=0
proxyid=Spoke1 proto=0 sa=1 ref=2 serial=1 adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=12226 type=00 soft=0 mtu=1438 expire=43055/0B replaywin=2048
       seqno=2 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43189/43200
  dec: spi=17a473bd esp=aes key=16 c78e5085857d0c5842e394fc44b38822
       ah=sha1 key=20 0bb885a85f77aa491a1209e4d36b7cddd7caf152
  enc: spi=75cde309 esp=aes key=16 6717935721e4a25428d6a7a633da75a9
       ah=sha1 key=20 eaf092280cf5b9f9db09ac95258786ffbfacead0
  dec:pkts/bytes=0/0, enc:pkts/bytes=2/144
  npu_flag=00 npu_rgwy=172.16.200.3 npu_lgwy=172.16.200.1 npu_selid=b dec_npuid=0 enc_
npuid=0
------------------------------------------------------
name=Spoke1 ver=2 serial=1 172.16.200.1:0->172.16.200.4:0 tun_id=172.16.200.4 tun_
id6=::172.16.200.4 dst_mtu=1500 dpd-link=on weight=1
bound_if=19 lgwy=static/1 tun=intf mode=auto/1 encap=none/560 options[0230]=create_dev
frag-rfc  role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=1 refcnt=5 ilast=0 olast=0 ad=r/2
stat: rxp=542 txp=553 rxb=22117 txb=22748
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=Spoke1 proto=0 sa=1 ref=4 serial=1 adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=12226 type=00 soft=0 mtu=1438 expire=42636/0B replaywin=2048
       seqno=22a esn=0 replaywin_lastseq=0000021f qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42900/43200
  dec: spi=17a473bc esp=aes key=16 eff2dc03b48968bb55b9e3950ebde431
       ah=sha1 key=20 5db42a32aec15bc8a5fe392c256d1ae8ab3b4ef8
  enc: spi=bdc3bd80 esp=aes key=16 d0ec06b61ad572cc8813b599edde8c68
       ah=sha1 key=20 0306850f0184d957e9475da33d7971653a95c233
  dec:pkts/bytes=1084/44234, enc:pkts/bytes=1106/80932
  npu_flag=00 npu_rgwy=172.16.200.4 npu_lgwy=172.16.200.1 npu_selid=0 dec_npuid=0 enc_
npuid=0
```

The dynamic selector is created (highlighted) for SD-WAN control traffic, SD-WAN health checks, and BGP between spokes traffic.

3. Verify the BGP neighbors and check the routing table:

```
Spoke1 # get router info bgp summary

VRF 0 BGP router identifier 172.16.200.1, local AS number 65412
BGP table version is 8
1 BGP AS-PATH entries
0 BGP community entries

Neighbor    V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.10.1.2   4      65412      52      76        7    0    0 00:06:27        1
```

```
10.10.1.250 4      65412      70      69        1    0    0 00:58:44        2

Total number of neighbors 2
```

4. Stop sending traffic between the spokes, and wait for a few minutes (idle timeout).
5. Verify the IPsec tunnel state on the Spoke1 FortiGate:

```
Spoke1 # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=Spoke1 ver=2 serial=1 172.16.200.1:0->172.16.200.4:0 tun_id=172.16.200.4 tun_
id6=::172.16.200.4 dst_mtu=1500 dpd-link=on weight=1
bound_if=19 lgwy=static/1 tun=intf mode=auto/1 encap=none/560 options[0230]=create_dev
frag-rfc  role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=0 olast=0 ad=r/2
stat: rxp=1467 txp=1469 rxb=60190 txb=60214
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=Spoke1 proto=0 sa=1 ref=3 serial=1 adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=12226 type=00 soft=0 mtu=1438 expire=42199/0B replaywin=2048
       seqno=5be esn=0 replaywin_lastseq=000005bc qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42903/43200
  dec: spi=76fdf7d1 esp=aes key=16 b26fd2dae76665f580d255b67f79df1e
       ah=sha1 key=20 14b0acc3c8c92a0af8ab43ff0437d2141b6d3f65
  enc: spi=bdc3bd85 esp=aes key=16 3eae3ad42aa32d7cdd972dfca286acd1
       ah=sha1 key=20 3655f67ee135f38e3f0790f1c7e3bd19c4a9285c
  dec:pkts/bytes=2934/120380, enc:pkts/bytes=2938/214606
  npu_flag=00 npu_rgwy=172.16.200.4 npu_lgwy=172.16.200.1 npu_selid=0 dec_npuid=0 enc_
npuid=0
```

The shortcut tunnel between the spokes has been torn down. When data traffic is idle, the BGP traffic does not get sent on the data traffic selector, so the tunnel is not kept alive. This behavior is the expected, which consequently allows the shortcut tunnel to be torn down when idle.

6. Verify the IKE debugs messages to confirm the ADVPN shortcut was torn down:

```
Spoke1 # diagnose debug enable
Spoke1 # diagnose debug application ike -1
...
ike 0:Spoke1_0: connection idle time-out
ike 0:Spoke1_0: deleting
ike 0:Spoke1_0: flushing
ike 0:Spoke1_0: deleting IPsec SA with SPI 75cde338
ike 0:Spoke1_0:Spoke1: deleted IPsec SA with SPI 75cde338, SA count: 0
ike 0:Spoke1_0: sending SNMP tunnel DOWN trap for Spoke1
ike 0:Spoke1_0: tunnel down event 0.0.0.0
ike 0:Spoke1_0:Spoke1: delete
ike 0:Spoke1_0: deleting IPsec SA with SPI 75cde337
ike 0:Spoke1_0:Spoke1: deleted IPsec SA with SPI 75cde337, SA count: 0
ike 0:Spoke1_0: sending SNMP tunnel DOWN trap for Spoke1
ike 0:Spoke1_0: tunnel down event 0.0.0.0
ike 0:Spoke1_0:Spoke1: delete
ike 0:Spoke1_0: flushed
```

```
ike 0:Spoke1_0:23:86: send informational
ike 0:Spoke1_0:23: sent IKE msg (INFORMATIONAL): 172.16.200.1:500->172.16.200.3:500,
len=80, vrf=0, id=0304e1284a432105/fa7d3fd75e7f481e:00000004
ike 0:Spoke1_0: delete connected route 10.10.1.1 -> 10.10.1.2
ike 0:Spoke1_0: delete dynamic
ike 0:Spoke1_0: deleted
ike 0:Spoke1: schedule auto-negotiate
ike 0: comes 172.16.200.3:500->172.16.200.1:500,ifindex=19,vrf=0....
ike 0: IKEv2 exchange=INFORMATIONAL_RESPONSE
id=0304e1284a432105/fa7d3fd75e7f481e:00000004 len=80
```

Improve client-side settings for SD-WAN network monitor - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Speed test examples

Improvements have been made to the client-side settings of the SD-WAN network bandwidth monitoring service to increase the flexibility of the speed tests, and to optimize the settings to produce more accurate measurements. The changes include:

- Support UDP speed tests.
- Support multiple TCP connections to the server instead of a single connection.
- Measure the latency to speed test servers and select the server with the smallest latency to perform the test.
- Support the auto mode speed test, which selects either UDP or TCP testing automatically based on the latency threshold.

Summary of related CLI commands

To configure the speed test settings:

```
config system speed-test-setting
    set latency-threshold <integer>
    set multiple-tcp-stream <integer>
end
```

latency-threshold <integer>	Set the speed test threshold for the auto mode, in milliseconds (0 - 2000, default = 60). If the latency exceeds this threshold, the speed test will use the UDP protocol; otherwise, it will use the TCP protocol.
multiple-tcp-stream <integer>	Set the number of parallel client streams for the TCP protocol to run during the speed test (1 - 64, default = 4).

To run a manual interface speed test:

```
# execute speed-test <interface> <server> {Auto | TCP | UDP}
```

```
# diagnose netlink interface speed-test <interface> <server> {Auto | TCP | UDP}
```

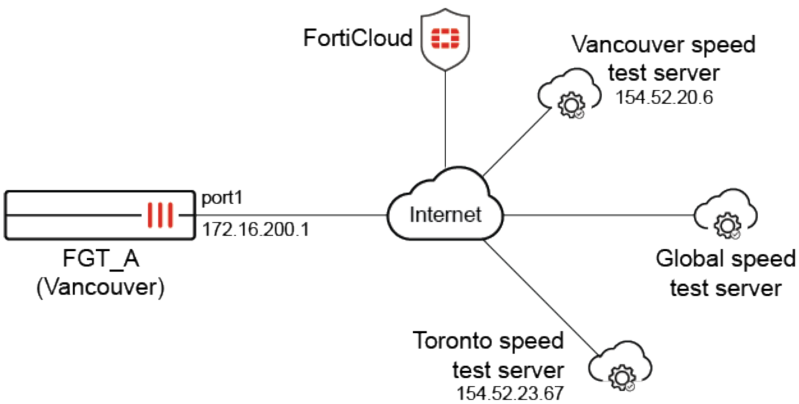
To configure the protocol mode for a speed test:

```
config system speed-test-schedule
    edit <interface>
        set mode {Auto | TCP | UDP}
    next
end
```

`Auto` is the default setting.

Examples

The following examples show various tests based on different modes (Auto, TCP, UDP), latency thresholds, and test servers. Some test protocols and servers are manually configured, while others are chosen by the FortiGate.



These examples assume the FortiGate is connected to the internet, has a valid SD-WAN Network Monitor license, and has downloaded the server list of speed tests from FortiCloud.

To download the server list of speed tests:

1. Download the server list from FortiCloud:

```
# execute speed-test-server download
Download completed.
```

2. Verify the list:

```
# execute speed-test-server list
...
FTNT_CA_Toronto valid
        Host: 154.52.23.67 5200 fortinet
        ...
FTNT_CA_Vancouver valid
        Host: 154.52.20.6 5200 fortinet
        ...
FTNT_Global valid
        Host: 154.52.6.95 5203 fortinet
        ...
```

Example 1: executing a speed test without specifying the interface, server, and mode

Geographically, the Vancouver server (154.52.20.6) has the smallest latency (around 7 ms) to FGT_A, so it will be automatically selected for the speed test because the latency 7 ms to 154.52.20.6 is less than the default `latency-threshold` of 60 ms. Meanwhile, four TCP connections will be initiated to perform the test since the default `multiple-tcp-stream` is 4.

To execute the speed test without specifying parameters:

1. Configure the speed test settings:

```
config system speed-test-setting
    set latency-threshold 60
    set multiple-tcp-stream 4
end
```

2. Execute a ping to the closest test server, 154.52.20.6, to learn the latency for the connection:

```
# execute ping 154.52.20.6
PING 154.52.20.6 (154.52.20.6): 56 data bytes
64 bytes from 154.52.20.6: icmp_seq=0 ttl=50 time=7.5 ms
64 bytes from 154.52.20.6: icmp_seq=1 ttl=50 time=7.2 ms
64 bytes from 154.52.20.6: icmp_seq=2 ttl=50 time=7.1 ms
64 bytes from 154.52.20.6: icmp_seq=3 ttl=50 time=7.1 ms
64 bytes from 154.52.20.6: icmp_seq=4 ttl=50 time=9.1 ms

--- 154.52.20.6 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 7.1/7.6/9.1 ms
```

3. Run the speed test with no parameters:

```
# execute speed-test
Initializing speed test.
current vdom=root
Run in uploading mode.
Connecting to host 154.52.20.6, port 5203
[  7] local 172.16.200.1 port 21219 connected to 154.52.20.6 port 5203
[  9] local 172.16.200.1 port 21220 connected to 154.52.20.6 port 5203
[ 11] local 172.16.200.1 port 21221 connected to 154.52.20.6 port 5203
[ 13] local 172.16.200.1 port 21222 connected to 154.52.20.6 port 5203
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  7]   0.00-1.00   sec  22.4 MBytes   188 Mbits/sec   17    140 KBytes
[  9]   0.00-1.00   sec  9.71 MBytes  81.4 Mbits/sec    6   73.5 KBytes
[ 11]   0.00-1.00   sec  18.5 MBytes   155 Mbits/sec   12    117 KBytes
[ 13]   0.00-1.00   sec  12.4 MBytes   104 Mbits/sec    7   87.7 KBytes
[SUM]   0.00-1.00   sec  63.1 MBytes   529 Mbits/sec   42
...
[ ID] Interval           Transfer     Bitrate         Retr
[  7]   0.00-5.00   sec  97.8 MBytes   164 Mbits/sec   45             sender
[  7]   0.00-5.02   sec  97.7 MBytes   163 Mbits/sec                  receiver
[  9]   0.00-5.00   sec  63.1 MBytes   106 Mbits/sec   14             sender
[  9]   0.00-5.02   sec  62.9 MBytes   105 Mbits/sec                  receiver
[ 11]   0.00-5.00   sec  80.1 MBytes   134 Mbits/sec   29             sender
[ 11]   0.00-5.02   sec  79.9 MBytes   134 Mbits/sec                  receiver
[ 13]   0.00-5.00   sec  80.3 MBytes   135 Mbits/sec   49             sender
[ 13]   0.00-5.02   sec  80.2 MBytes   134 Mbits/sec                  receiver
```

```
[SUM]   0.00-5.00   sec   321 MBytes   539 Mbits/sec  137             sender
[SUM]   0.00-5.02   sec   321 MBytes   536 Mbits/sec                  receiver

speed test Done.
Run in reverse downloading mode.
Connecting to host 154.52.20.6, port 5203
Reverse mode, remote host 154.52.20.6 is sending
[  7] local 172.16.200.1 port 21228 connected to 154.52.20.6 port 5203
[ 11] local 172.16.200.1 port 21229 connected to 154.52.20.6 port 5203
[ 15] local 172.16.200.1 port 21230 connected to 154.52.20.6 port 5203
[ 17] local 172.16.200.1 port 21231 connected to 154.52.20.6 port 5203
[ ID] Interval           Transfer     Bitrate
[  7]   0.00-1.00   sec  30.6 MBytes   256 Mbits/sec
[ 11]   0.00-1.00   sec  20.2 MBytes   170 Mbits/sec
[ 15]   0.00-1.00   sec  23.0 MBytes   193 Mbits/sec
[ 17]   0.00-1.00   sec  18.1 MBytes   152 Mbits/sec
[SUM]   0.00-1.00   sec  91.9 MBytes   771 Mbits/sec
...
[ ID] Interval           Transfer     Bitrate         Retr
[  7]   0.00-5.01   sec   101 MBytes   169 Mbits/sec  458             sender
[  7]   0.00-5.00   sec  97.4 MBytes   163 Mbits/sec                  receiver
[ 11]   0.00-5.01   sec  93.1 MBytes   156 Mbits/sec  266             sender
[ 11]   0.00-5.00   sec  91.8 MBytes   154 Mbits/sec                  receiver
[ 15]   0.00-5.01   sec  76.3 MBytes   128 Mbits/sec  201             sender
[ 15]   0.00-5.00   sec  74.7 MBytes   125 Mbits/sec                  receiver
[ 17]   0.00-5.01   sec  68.7 MBytes   115 Mbits/sec  219             sender
[ 17]   0.00-5.00   sec  66.8 MBytes   112 Mbits/sec                  receiver
[SUM]   0.00-5.01   sec   339 MBytes   568 Mbits/sec  1144             sender
[SUM]   0.00-5.00   sec   331 MBytes   555 Mbits/sec                  receiver

speed test Done.
```

The tested upload/download speed for port1 is 536 Mbps/555 Mbps when connecting to the closest server with four TCP connections.

Example 2: executing a speed test with a lower latency threshold setting

The `latency-threshold` setting is changed to 5 ms, which is less than the latency 7 ms to 154.52.20.6. When executing the speed test, one UDP connection will be initiated as expected.

To execute the speed test with a lower latency threshold setting:

1. Edit the speed test settings:

```
config system speed-test-setting
    set latency-threshold 5
end
```

2. Run the speed test:

```
# execute speed-test
Speed test quota for 7/19 is 4
current vdom=root
Run in uploading mode.
Connecting to host 154.52.20.6, port 5202
[  7] local 172.16.200.1 port 5315 connected to 154.52.20.6 port 5202
```

```
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  7]   0.00-1.00   sec   111 MBytes   931 Mbits/sec  80337
[  7]   1.00-2.00   sec   111 MBytes   932 Mbits/sec  80476
[  7]   2.00-3.00   sec   111 MBytes   932 Mbits/sec  80451
[  7]   3.00-4.00   sec   111 MBytes   932 Mbits/sec  80460
[  7]   4.00-5.00   sec   111 MBytes   934 Mbits/sec  80640
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  7]   0.00-5.00   sec   556 MBytes   932 Mbits/sec  0.000 ms  0/402364 (0%)  sender
[  7]   0.00-5.04   sec   550 MBytes   917 Mbits/sec  0.017 ms  3787/402339 (0.94%)
receiver

speed test Done.
Run in reverse downloading mode.
Connecting to host 154.52.20.6, port 5202
Reverse mode, remote host 154.52.20.6 is sending
[  7] local 172.16.200.1 port 19940 connected to 154.52.20.6 port 5202
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  7]   0.00-1.00   sec  72.4 MBytes   607 Mbits/sec  0.013 ms  59813/112240 (53%)
[  7]   1.00-2.00   sec  70.9 MBytes   595 Mbits/sec  0.015 ms  58130/109486 (53%)
[  7]   2.00-3.00   sec  69.2 MBytes   581 Mbits/sec  0.012 ms  60192/110329 (55%)
[  7]   3.00-4.00   sec  71.3 MBytes   598 Mbits/sec  0.012 ms  58107/109710 (53%)
[  7]   4.00-5.00   sec  71.1 MBytes   596 Mbits/sec  0.014 ms  58786/110260 (53%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  7]   0.00-5.04   sec   764 MBytes  1.27 Gbits/sec  0.000 ms  0/553023 (0%)  sender
[SUM]  0.0- 5.0 sec  2 datagrams received out-of-order
[  7]   0.00-5.00   sec   355 MBytes   595 Mbits/sec  0.014 ms  295028/552025 (53%)
receiver

speed test Done.
```

The tested upload/download speed for port1 is 917 Mbps/595 Mbps when connecting to the closest server with one UDP connection.

Example 3: executing a speed test with 10 TCP client streams

The `latency-threshold` setting is back to the 60 ms default, and the `multiple-tcp-stream` setting is changed to 10.

To execute the speed test with the default latency threshold and higher client stream value:

1. Edit the speed test settings:

```
config system speed-test-setting
    set latency-threshold 60
    set multiple-tcp-stream 10
end
```

2. Run the speed test:

```
# execute speed-test
Speed test quota for 7/19 is 3
current vdom=root
Run in uploading mode.
Connecting to host 154.52.20.6, port 5203
[  7] local 172.16.200.1 port 22373 connected to 154.52.20.6 port 5203
```

```
[  9] local 172.16.200.1 port 22374 connected to 154.52.20.6 port 5203
[ 11] local 172.16.200.1 port 22375 connected to 154.52.20.6 port 5203
[ 13] local 172.16.200.1 port 22376 connected to 154.52.20.6 port 5203
[ 15] local 172.16.200.1 port 22377 connected to 154.52.20.6 port 5203
[ 17] local 172.16.200.1 port 22378 connected to 154.52.20.6 port 5203
[ 19] local 172.16.200.1 port 22379 connected to 154.52.20.6 port 5203
[ 21] local 172.16.200.1 port 22380 connected to 154.52.20.6 port 5203
[ 23] local 172.16.200.1 port 22381 connected to 154.52.20.6 port 5203
[ 25] local 172.16.200.1 port 22382 connected to 154.52.20.6 port 5203
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  7]   0.00-1.00   sec  15.1 MBytes   127 Mbits/sec   14   72.1 KBytes
[  9]   0.00-1.00   sec  8.42 MBytes  70.6 Mbits/sec    9   43.8 KBytes
[ 11]   0.00-1.00   sec  11.9 MBytes  99.8 Mbits/sec   11   82.0 KBytes
[ 13]   0.00-1.00   sec  8.12 MBytes  68.0 Mbits/sec   10   55.1 KBytes
[ 15]   0.00-1.00   sec  5.49 MBytes  46.1 Mbits/sec   11   32.5 KBytes
[ 17]   0.00-1.00   sec  5.77 MBytes  48.3 Mbits/sec    7   59.4 KBytes
[ 19]   0.00-1.00   sec  17.8 MBytes   149 Mbits/sec   16    133 KBytes
[ 21]   0.00-1.00   sec  9.52 MBytes  79.8 Mbits/sec    7   67.9 KBytes
[ 23]   0.00-1.00   sec  4.84 MBytes  40.6 Mbits/sec    7   35.4 KBytes
[ 25]   0.00-1.00   sec  7.92 MBytes  66.4 Mbits/sec    9   79.2 KBytes
[SUM]   0.00-1.00   sec  94.9 MBytes   796 Mbits/sec  101
...
[ ID] Interval           Transfer     Bitrate         Retr
[  7]   0.00-5.00   sec  52.7 MBytes  88.3 Mbits/sec   34             sender
[  7]   0.00-5.01   sec  52.5 MBytes  88.0 Mbits/sec                  receiver
[  9]   0.00-5.00   sec  40.8 MBytes  68.5 Mbits/sec   22             sender
[  9]   0.00-5.01   sec  40.7 MBytes  68.2 Mbits/sec                  receiver
[ 11]   0.00-5.00   sec  42.8 MBytes  71.7 Mbits/sec   26             sender
[ 11]   0.00-5.01   sec  42.7 MBytes  71.5 Mbits/sec                  receiver
[ 13]   0.00-5.00   sec  34.8 MBytes  58.4 Mbits/sec   27             sender
[ 13]   0.00-5.01   sec  34.7 MBytes  58.1 Mbits/sec                  receiver
[ 15]   0.00-5.00   sec  38.7 MBytes  64.8 Mbits/sec   23             sender
[ 15]   0.00-5.01   sec  38.6 MBytes  64.6 Mbits/sec                  receiver
[ 17]   0.00-5.00   sec  35.7 MBytes  59.9 Mbits/sec   22             sender
[ 17]   0.00-5.01   sec  35.7 MBytes  59.8 Mbits/sec                  receiver
[ 19]   0.00-5.00   sec  58.2 MBytes  97.5 Mbits/sec   39             sender
[ 19]   0.00-5.01   sec  57.9 MBytes  97.0 Mbits/sec                  receiver
[ 21]   0.00-5.00   sec  34.2 MBytes  57.4 Mbits/sec   29             sender
[ 21]   0.00-5.01   sec  34.1 MBytes  57.2 Mbits/sec                  receiver
[ 23]   0.00-5.00   sec  29.6 MBytes  49.7 Mbits/sec   26             sender
[ 23]   0.00-5.01   sec  29.6 MBytes  49.5 Mbits/sec                  receiver
[ 25]   0.00-5.00   sec  54.6 MBytes  91.5 Mbits/sec   28             sender
[ 25]   0.00-5.01   sec  54.5 MBytes  91.3 Mbits/sec                  receiver
[SUM]   0.00-5.00   sec   422 MBytes   708 Mbits/sec  276             sender
[SUM]   0.00-5.01   sec   421 MBytes   705 Mbits/sec                  receiver

speed test Done.
Run in reverse downloading mode.
Connecting to host 154.52.20.6, port 5203
Reverse mode, remote host 154.52.20.6 is sending
[  7] local 172.16.200.1 port 22384 connected to 154.52.20.6 port 5203
[ 11] local 172.16.200.1 port 22385 connected to 154.52.20.6 port 5203
[ 15] local 172.16.200.1 port 22386 connected to 154.52.20.6 port 5203
[ 19] local 172.16.200.1 port 22387 connected to 154.52.20.6 port 5203
[ 23] local 172.16.200.1 port 22388 connected to 154.52.20.6 port 5203
[ 27] local 172.16.200.1 port 22389 connected to 154.52.20.6 port 5203
```

```
[ 29] local 172.16.200.1 port 22390 connected to 154.52.20.6 port 5203
[ 31] local 172.16.200.1 port 22391 connected to 154.52.20.6 port 5203
[ 33] local 172.16.200.1 port 22392 connected to 154.52.20.6 port 5203
[ 35] local 172.16.200.1 port 22393 connected to 154.52.20.6 port 5203
[ ID] Interval           Transfer     Bitrate
[  7]   0.00-1.00   sec  11.5 MBytes  96.7 Mbits/sec
[ 11]   0.00-1.00   sec  7.97 MBytes  66.9 Mbits/sec
[ 15]   0.00-1.00   sec  6.19 MBytes  52.0 Mbits/sec
[ 19]   0.00-1.00   sec  8.27 MBytes  69.4 Mbits/sec
[ 23]   0.00-1.00   sec  8.34 MBytes  69.9 Mbits/sec
[ 27]   0.00-1.00   sec  5.85 MBytes  49.0 Mbits/sec
[ 29]   0.00-1.00   sec  7.64 MBytes  64.1 Mbits/sec
[ 31]   0.00-1.00   sec  5.61 MBytes  47.0 Mbits/sec
[ 33]   0.00-1.00   sec  6.95 MBytes  58.3 Mbits/sec
[ 35]   0.00-1.00   sec  6.43 MBytes  53.9 Mbits/sec
[SUM]   0.00-1.00   sec  74.8 MBytes   627 Mbits/sec
...
[ ID] Interval           Transfer     Bitrate         Retr
[  7]   0.00-5.01   sec  39.4 MBytes  65.9 Mbits/sec  197             sender
[  7]   0.00-5.00   sec  37.6 MBytes  63.0 Mbits/sec                  receiver
[ 11]   0.00-5.01   sec  49.0 MBytes  82.1 Mbits/sec  216             sender
[ 11]   0.00-5.00   sec  48.1 MBytes  80.8 Mbits/sec                  receiver
[ 15]   0.00-5.01   sec  27.4 MBytes  45.9 Mbits/sec  206             sender
[ 15]   0.00-5.00   sec  26.4 MBytes  44.3 Mbits/sec                  receiver
[ 19]   0.00-5.01   sec  42.6 MBytes  71.3 Mbits/sec  158             sender
[ 19]   0.00-5.00   sec  42.1 MBytes  70.6 Mbits/sec                  receiver
[ 23]   0.00-5.01   sec  37.1 MBytes  62.2 Mbits/sec  174             sender
[ 23]   0.00-5.00   sec  36.6 MBytes  61.4 Mbits/sec                  receiver
[ 27]   0.00-5.01   sec  34.6 MBytes  58.0 Mbits/sec  161             sender
[ 27]   0.00-5.00   sec  34.1 MBytes  57.2 Mbits/sec                  receiver
[ 29]   0.00-5.01   sec  40.2 MBytes  67.4 Mbits/sec  135             sender
[ 29]   0.00-5.00   sec  39.6 MBytes  66.5 Mbits/sec                  receiver
[ 31]   0.00-5.01   sec  40.9 MBytes  68.5 Mbits/sec  172             sender
[ 31]   0.00-5.00   sec  40.4 MBytes  67.8 Mbits/sec                  receiver
[ 33]   0.00-5.01   sec  35.4 MBytes  59.3 Mbits/sec  164             sender
[ 33]   0.00-5.00   sec  34.9 MBytes  58.5 Mbits/sec                  receiver
[ 35]   0.00-5.01   sec  37.2 MBytes  62.3 Mbits/sec  148             sender
[ 35]   0.00-5.00   sec  36.7 MBytes  61.5 Mbits/sec                  receiver
[SUM]   0.00-5.01   sec   384 MBytes   643 Mbits/sec  1731             sender
[SUM]   0.00-5.00   sec   377 MBytes   632 Mbits/sec                  receiver

speed test Done.
```

The tested upload/download speed for port1 is 705 Mbps/632 Mbps when connecting to the closest server with 10 TCP connections.

Example 4: executing a speed test by specifying the interface, server, and UDP mode

The speed test will test the Toronto server using UDP mode on port1.

To execute the speed test:

```
# execute speed-test port1 FTNT_CA_Toronto UDP
Speed test quota for 7/19 is 1
bind to local ip 172.16.200.1
current vdom=root
```

```
Run in uploading mode.
Connecting to host 154.52.23.67, port 5201
[  7] local 172.16.200.1 port 10860 connected to 154.52.23.67 port 5201
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  7]   0.00-1.00   sec   112 MBytes   936 Mbits/sec  80759
[  7]   1.00-2.00   sec   112 MBytes   937 Mbits/sec  80886
[  7]   2.00-3.00   sec   112 MBytes   937 Mbits/sec  80903
[  7]   3.00-4.00   sec   111 MBytes   935 Mbits/sec  80677
[  7]   4.00-5.00   sec   111 MBytes   934 Mbits/sec  80600
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  7]   0.00-5.00   sec   558 MBytes   936 Mbits/sec  0.000 ms  0/403825 (0%)  sender
[  7]   0.00-5.09   sec   552 MBytes   908 Mbits/sec  0.013 ms  4435/403815 (1.1%)  receiver

speed test Done.
Run in reverse downloading mode.
Connecting to host 154.52.23.67, port 5201
Reverse mode, remote host 154.52.23.67 is sending
[  7] local 172.16.200.1 port 15370 connected to 154.52.23.67 port 5201
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  7]   0.00-1.00   sec  58.8 MBytes   493 Mbits/sec  0.017 ms  60888/103447 (59%)
[  7]   1.00-2.00   sec  58.3 MBytes   489 Mbits/sec  0.012 ms  93083/135310 (69%)
[  7]   2.00-3.00   sec  59.4 MBytes   499 Mbits/sec  0.017 ms  95066/138106 (69%)
[  7]   3.00-4.00   sec  54.0 MBytes   453 Mbits/sec  0.024 ms  97539/136672 (71%)
[  7]   4.00-5.00   sec  58.6 MBytes   491 Mbits/sec  0.015 ms  93797/136213 (69%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  7]   0.00-5.10   sec   908 MBytes  1.49 Gbits/sec  0.000 ms  0/657629 (0%)  sender
[  7]   0.00-5.00   sec   289 MBytes   485 Mbits/sec  0.015 ms  440373/649748 (68%)
receiver

speed test Done.
```

The tested upload/download speed for port1 is 908 Mbps/485 Mbps when connecting to the Toronto server with one UDP connection.

Example 5: executing a speed test by specifying the interface, server, and auto mode

The speed test will test the Toronto server using auto mode on port1. Since the latency to the Toronto server is less than 60 ms, 10 TCP connections are initiated.

To execute the speed test:

```
# execute speed-test port1 FTNT_CA_Toronto Auto
Speed test quota for 7/19 is 8
bind to local ip 172.16.200.1
current vdom=root
Run in uploading mode.
Connecting to host 154.52.23.67, port 5200
[  7] local 172.16.200.1 port 4333 connected to 154.52.23.67 port 5200
[  9] local 172.16.200.1 port 4334 connected to 154.52.23.67 port 5200
[ 11] local 172.16.200.1 port 4335 connected to 154.52.23.67 port 5200
[ 13] local 172.16.200.1 port 4336 connected to 154.52.23.67 port 5200
[ 15] local 172.16.200.1 port 4337 connected to 154.52.23.67 port 5200
[ 17] local 172.16.200.1 port 4338 connected to 154.52.23.67 port 5200
[ 19] local 172.16.200.1 port 4339 connected to 154.52.23.67 port 5200
```

```
[ 21] local 172.16.200.1 port 4340 connected to 154.52.23.67 port 5200
[ 23] local 172.16.200.1 port 4341 connected to 154.52.23.67 port 5200
[ 25] local 172.16.200.1 port 4342 connected to 154.52.23.67 port 5200
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  7]   0.00-1.00   sec  1.61 MBytes  13.5 Mbits/sec    1    264 KBytes
[  9]   0.00-1.00   sec  1.06 MBytes  8.90 Mbits/sec    0    160 KBytes
[ 11]   0.00-1.00   sec  1.35 MBytes  11.3 Mbits/sec    0    184 KBytes
[ 13]   0.00-1.00   sec  1.46 MBytes  12.2 Mbits/sec    0    222 KBytes
[ 15]   0.00-1.00   sec  1.32 MBytes  11.1 Mbits/sec    0    182 KBytes
[ 17]   0.00-1.00   sec  1.79 MBytes  15.0 Mbits/sec    0    263 KBytes
[ 19]   0.00-1.00   sec   912 KBytes  7.46 Mbits/sec    0   97.6 KBytes
[ 21]   0.00-1.00   sec  1.47 MBytes  12.3 Mbits/sec    0    188 KBytes
[ 23]   0.00-1.00   sec  1.04 MBytes  8.75 Mbits/sec    0    175 KBytes
[ 25]   0.00-1.00   sec   929 KBytes  7.60 Mbits/sec    0   94.7 KBytes
[SUM]   0.00-1.00   sec  12.9 MBytes   108 Mbits/sec    1
...
[ ID] Interval           Transfer     Bitrate         Retr
[  7]   0.00-5.00   sec  28.1 MBytes  47.1 Mbits/sec    8             sender
[  7]   0.00-5.05   sec  27.5 MBytes  45.7 Mbits/sec                  receiver
[  9]   0.00-5.00   sec  11.8 MBytes  19.8 Mbits/sec   10             sender
[  9]   0.00-5.05   sec  11.1 MBytes  18.5 Mbits/sec                  receiver
[ 11]   0.00-5.00   sec  40.5 MBytes  68.0 Mbits/sec   11             sender
[ 11]   0.00-5.05   sec  40.1 MBytes  66.7 Mbits/sec                  receiver
[ 13]   0.00-5.00   sec  18.0 MBytes  30.2 Mbits/sec    6             sender
[ 13]   0.00-5.05   sec  17.6 MBytes  29.2 Mbits/sec                  receiver
[ 15]   0.00-5.00   sec  38.8 MBytes  65.2 Mbits/sec    1             sender
[ 15]   0.00-5.05   sec  38.8 MBytes  64.4 Mbits/sec                  receiver
[ 17]   0.00-5.00   sec  15.0 MBytes  25.2 Mbits/sec   10             sender
[ 17]   0.00-5.05   sec  14.8 MBytes  24.5 Mbits/sec                  receiver
[ 19]   0.00-5.00   sec  20.5 MBytes  34.4 Mbits/sec    1             sender
[ 19]   0.00-5.05   sec  20.3 MBytes  33.7 Mbits/sec                  receiver
[ 21]   0.00-5.00   sec  13.9 MBytes  23.2 Mbits/sec   12             sender
[ 21]   0.00-5.05   sec  13.2 MBytes  21.9 Mbits/sec                  receiver
[ 23]   0.00-5.00   sec  7.59 MBytes  12.7 Mbits/sec   13             sender
[ 23]   0.00-5.05   sec  7.37 MBytes  12.2 Mbits/sec                  receiver
[ 25]   0.00-5.00   sec  17.7 MBytes  29.7 Mbits/sec   10             sender
[ 25]   0.00-5.05   sec  17.4 MBytes  28.9 Mbits/sec                  receiver
[SUM]   0.00-5.00   sec   212 MBytes   355 Mbits/sec   82             sender
[SUM]   0.00-5.05   sec   208 MBytes   346 Mbits/sec                  receiver

speed test Done.
Run in reverse downloading mode.
Connecting to host 154.52.23.67, port 5200
Reverse mode, remote host 154.52.23.67 is sending
[  7] local 172.16.200.1 port 4344 connected to 154.52.23.67 port 5200
[ 11] local 172.16.200.1 port 4345 connected to 154.52.23.67 port 5200
[ 15] local 172.16.200.1 port 4346 connected to 154.52.23.67 port 5200
[ 19] local 172.16.200.1 port 4347 connected to 154.52.23.67 port 5200
[ 23] local 172.16.200.1 port 4348 connected to 154.52.23.67 port 5200
[ 27] local 172.16.200.1 port 4349 connected to 154.52.23.67 port 5200
[ 29] local 172.16.200.1 port 4350 connected to 154.52.23.67 port 5200
[ 31] local 172.16.200.1 port 4351 connected to 154.52.23.67 port 5200
[ 33] local 172.16.200.1 port 4352 connected to 154.52.23.67 port 5200
[ 35] local 172.16.200.1 port 4353 connected to 154.52.23.67 port 5200
[ ID] Interval           Transfer     Bitrate
[  7]   0.00-1.00   sec  2.31 MBytes  19.3 Mbits/sec
```

```
[ 11]   0.00-1.00   sec  2.70 MBytes  22.6 Mbits/sec
[ 15]   0.00-1.00   sec  1.80 MBytes  15.1 Mbits/sec
[ 19]   0.00-1.00   sec  2.33 MBytes  19.5 Mbits/sec
[ 23]   0.00-1.00   sec  1.30 MBytes  10.9 Mbits/sec
[ 27]   0.00-1.00   sec  1.55 MBytes  13.0 Mbits/sec
[ 29]   0.00-1.00   sec  3.65 MBytes  30.5 Mbits/sec
[ 31]   0.00-1.00   sec  1.35 MBytes  11.3 Mbits/sec
[ 33]   0.00-1.00   sec  3.26 MBytes  27.3 Mbits/sec
[ 35]   0.00-1.00   sec  2.85 MBytes  23.8 Mbits/sec
[SUM]   0.00-1.00   sec  23.1 MBytes   193 Mbits/sec
...
[ ID] Interval           Transfer     Bitrate         Retr
[  7]   0.00-5.06   sec  16.2 MBytes  26.9 Mbits/sec   33             sender
[  7]   0.00-5.00   sec  14.6 MBytes  24.5 Mbits/sec                  receiver
[ 11]   0.00-5.06   sec  13.9 MBytes  23.0 Mbits/sec   64             sender
[ 11]   0.00-5.00   sec  12.9 MBytes  21.6 Mbits/sec                  receiver
[ 15]   0.00-5.06   sec  8.61 MBytes  14.3 Mbits/sec   75             sender
[ 15]   0.00-5.00   sec  7.63 MBytes  12.8 Mbits/sec                  receiver
[ 19]   0.00-5.06   sec  11.9 MBytes  19.7 Mbits/sec   65             sender
[ 19]   0.00-5.00   sec  10.8 MBytes  18.2 Mbits/sec                  receiver
[ 23]   0.00-5.06   sec  7.37 MBytes  12.2 Mbits/sec   13             sender
[ 23]   0.00-5.00   sec  6.77 MBytes  11.4 Mbits/sec                  receiver
[ 27]   0.00-5.06   sec  7.44 MBytes  12.3 Mbits/sec   86             sender
[ 27]   0.00-5.00   sec  6.47 MBytes  10.8 Mbits/sec                  receiver
[ 29]   0.00-5.06   sec  19.0 MBytes  31.5 Mbits/sec   27             sender
[ 29]   0.00-5.00   sec  17.7 MBytes  29.6 Mbits/sec                  receiver
[ 31]   0.00-5.06   sec  7.11 MBytes  11.8 Mbits/sec   51             sender
[ 31]   0.00-5.00   sec  6.43 MBytes  10.8 Mbits/sec                  receiver
[ 33]   0.00-5.06   sec  21.5 MBytes  35.7 Mbits/sec   23             sender
[ 33]   0.00-5.00   sec  20.4 MBytes  34.2 Mbits/sec                  receiver
[ 35]   0.00-5.06   sec  18.4 MBytes  30.5 Mbits/sec   48             sender
[ 35]   0.00-5.00   sec  17.0 MBytes  28.6 Mbits/sec                  receiver
[SUM]   0.00-5.06   sec   131 MBytes   218 Mbits/sec  485             sender
[SUM]   0.00-5.00   sec   121 MBytes   202 Mbits/sec                  receiver

speed test Done.
```

The tested upload/download speed for port1 is 346 Mbps/202 Mbps when connecting to the Toronto server with 10 TCP connections.

Example 6: executing the speed test with diagnose netlink interface speed-test

After running this diagnose command, the results are recorded in the interface settings for reference as `measured-upstream-bandwidth` and `measured-downstream-bandwidth`.

To execute the speed test:

```
# diagnose netlink interface speed-test port1 FTNT_CA_Vancouver TCP
speed-test test ID is b0066
...
```

To view the interface settings:

```
show system interface port1
config system interface
```

```
    edit "port1"
        ...
        set measured-upstream-bandwidth 735682
        set measured-downstream-bandwidth 746573
        set bandwidth-measure-time 1689811319
        ...
    next
end
```

Example 7: executing the speed test according to the schedule

After running the speed test, the results are recorded in the interface settings for reference as `measured-upstream-bandwidth` and `measured-downstream-bandwidth`.

To execute the speed test according to the schedule:

1. Configure the recurring schedule:

```
config firewall schedule recurring
    edit "speedtest_recurring"
        set start 17:07
        set day sunday monday tuesday wednesday thursday friday saturday
    next
end
```

2. Configure the speed test schedule:

```
config system speed-test-schedule
    edit "port1"
        set mode TCP
        set schedules "speedtest_recurring"
    next
end
```

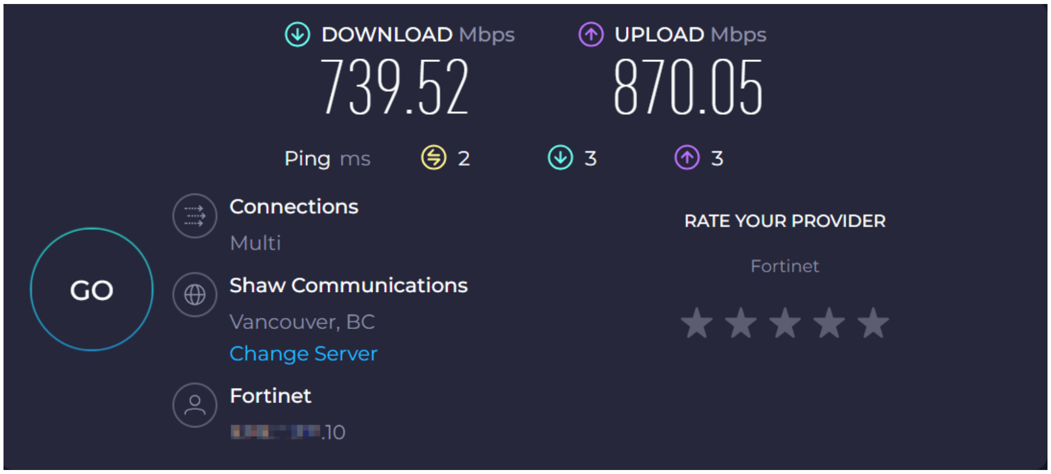
   The speed test will be initiated at 17:07 based on 10 TCP connections. The results will be recorded in port1's interface settings.

3. Verify the speed test results:

```
show system interface port1
config system interface
    edit "port1"
        ...
        set measured-upstream-bandwidth 715636
        set measured-downstream-bandwidth 819682
        set bandwidth-measure-time 1689811759
        ...
    next
end
```

Example 8: executing multiple speed tests with TCP and UDP connections

A speed test is executed to the closest server using 64 TCP connections and another speed test is executed using one UDP connection. The results can be checked with a third-party platform (such as Ookla), which returns comparable results.



To execute multiple speed tests with TCP and UDP connections:

1. Edit the speed test settings:

```
config system speed-test-setting
    set multiple-tcp-stream 64
end
```

2. Run the TCP speed test:

```
# execute speed-test port1 FTNT_CA_Vancouver TCP
...
Run in uploading mode.
...
[SUM]   0.00-5.00   sec   559 MBytes   938 Mbits/sec  2165                 sender
[SUM]   0.00-5.01   sec   558 MBytes   933 Mbits/sec                      receiver

speed test Done.
Run in reverse downloading mode.
...
[SUM]   0.00-5.01   sec   505 MBytes   846 Mbits/sec  9329                 sender
[SUM]   0.00-5.00   sec   491 MBytes   823 Mbits/sec                      receiver
```

3. Run the UDP speed test:

```
# execute speed-test port1 FTNT_CA_Vancouver UDP
...
Run in uploading mode.
...
[  7]   0.00-5.00   sec   556 MBytes   933 Mbits/sec  0.000 ms  0/402727 (0%)  sender
[  7]   0.00-5.04   sec   556 MBytes   925 Mbits/sec  0.020 ms  393/402717 (0.098%)
receiver
...
Run in reverse downloading mode.
...
[  7]   0.00-5.04   sec   869 MBytes  1.45 Gbits/sec  0.000 ms  0/629383 (0%)  sender
[SUM]  0.0- 5.0 sec  2 datagrams received out-of-order
[  7]   0.00-5.00   sec   489 MBytes   821 Mbits/sec  0.005 ms  274103/628393 (44%)
receiver

speed test Done.
```

Support the new SD-WAN Overlay-as-a-Service - 7.4.1

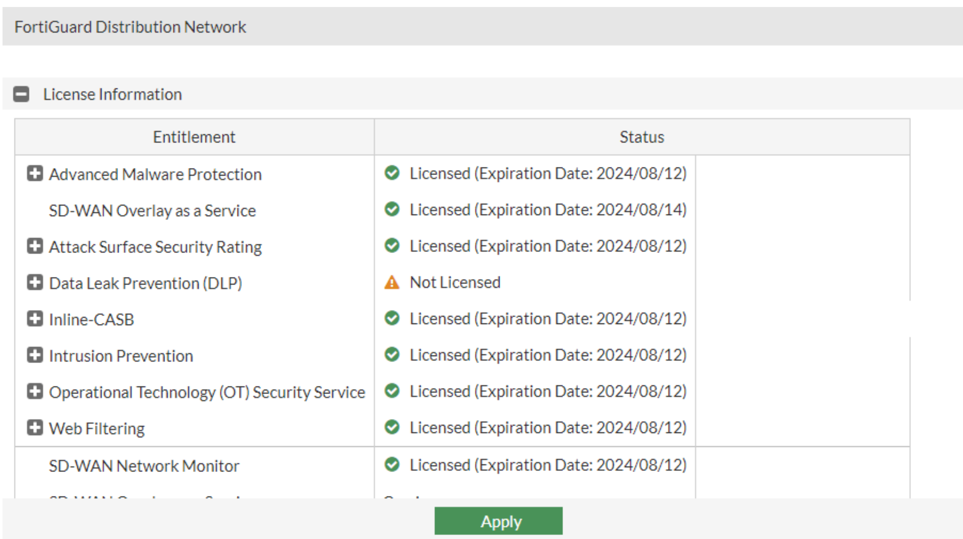
This information is also available in the FortiOS 7.4 Administration Guide:

- SD-WAN Overlay-as-a-Service

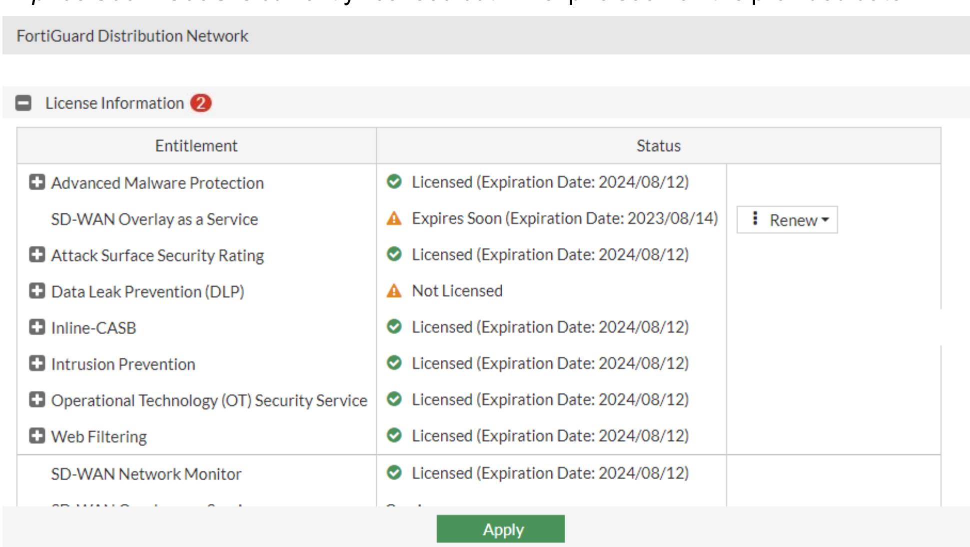
SD-WAN Overlay-as-a-Service (OaaS) is supported through a license displayed as *SD-WAN Overlay as a Service* on the *System* > *FortiGuard* page. Each FortiGate used by the FortiCloud Overlay-as-a-Service portal must have this license applied to it.

To view the status of the OaaS license in the GUI:

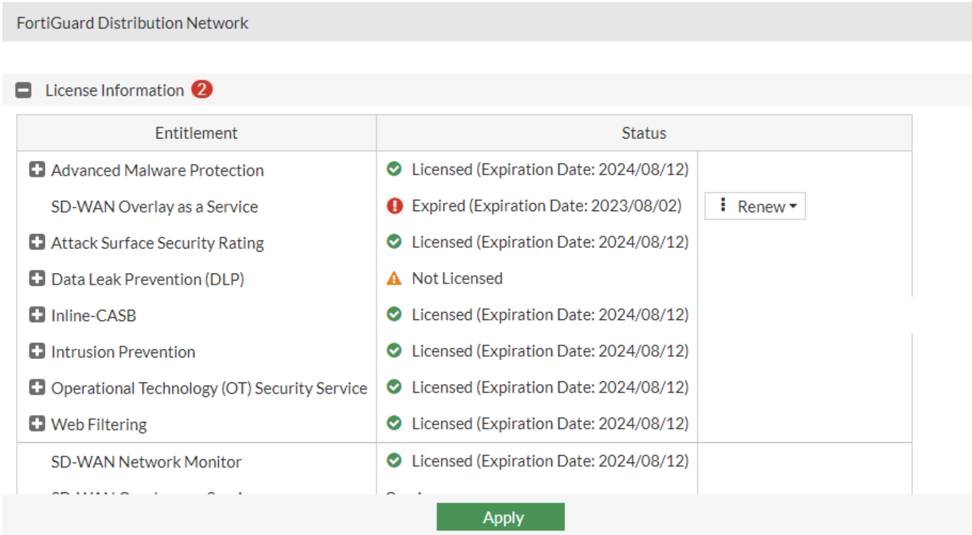
1. Go to *System* > *FortiGuard*.
2. Expand *License Information*. The *SD-WAN Overlay as a Service* license status is listed as:
   - *Licensed*: OaaS is currently licensed and will expire on the provided date.



   - *Expires Soon*: OaaS is currently licensed but will expire soon on the provided date.



- *Expired*: The OaaS license has already expired on the provided date.



To view the status of the OaaS license in the CLI:

1. Verify that the entitlement can be updated:

   The SD-WAN Overlay-as-a-Service license is listed as `SWOS` in the CLI.

```
# diagnose test update info

System contracts:
    FMWR,Wed Dec 20 16:00:00 2023
    SPAM,Wed Dec 20 16:00:00 2023
    SBCL,Wed Dec 20 16:00:00 2023
    SWNO,Wed Dec 20 16:00:00 2023
    SWNM,Wed Sep 27 17:00:00 2023
    SWOS,Mon Aug 14 17:00:00 2023
    SPRT,Wed Dec 20 16:00:00 2023
    SDWN,Sun Dec 10 16:00:00 2023
    SBCL,Wed Dec 20 16:00:00 2023
    SBEN,Wed Dec 20 16:00:00 2023
```

2. Verify that the expiration date log can be generated:

```
# execute log display

1: date=2023-08-10 time=00:00:01 eventtime=1691650800645347120 tz="-0700"
logid="0100020138" type="event" subtype="system" level="warning" vd="root"
logdesc="FortiGuard SD-WAN Overlay as a Service license expiring" msg="FortiGuard SD-WAN
Overlay Service license will expire in 4 day(s)"
```

IPv6 support for SD-WAN segmentation over a single overlay - 7.4.2

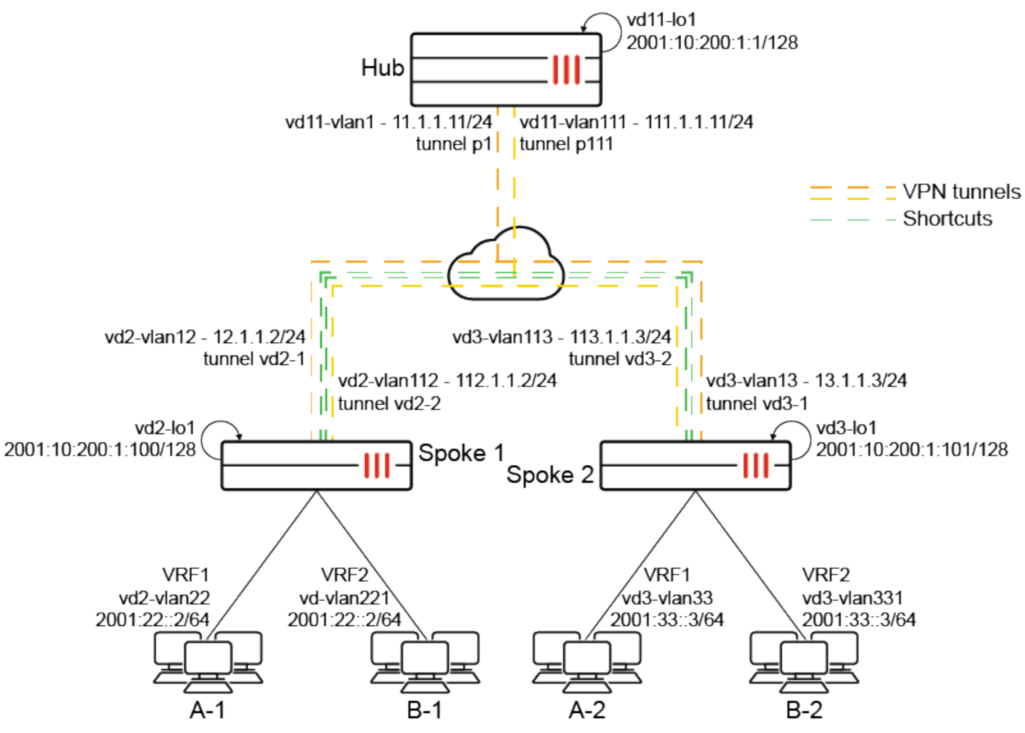
This information is also available in the FortiOS 7.4 Administration Guide:

- SD-WAN segmentation over a single overlay using IPv6

IPv6 is supported for SD-WAN segmentation over a single overlay. This allows seamless communication between IPv6 devices within virtual routing and forwarding (VRF) overlay networks, benefiting organizations transitioning to IPv6 or operating in a dual-stack environment.

Example

In this example, multiple companies (or departments of a company) share the ADVPN. Company A and company B each have two branches in two different locations. Company A's branches (A-1 and A-2) can talk to each other using the VPN shortcut, but not to company B's branches (B-1 and B-2). Likewise, company B's branches can talk to each other using the VPN shortcut, but not to company A's branches. Traffic can share the tunnels and shortcuts, but cannot be mixed up.



In this example, two spokes each have two tunnels to the hub.

- Each spoke has two VRFs behind it that can use the same IP address or subnets.
- The computers in VRF1 behind spoke 1 can talk to the computers in VRF1 behind spoke 2, but not to any of the computers in the VRF2s behind either spoke.
- The computers in VRF2 behind spoke 1 can talk to the computers in VRF2 behind spoke 2, but not to any of the computers in the VRF1s behind either spoke.
- Loopback addresses are used for communication between the spokes and the hub instead of tunnel IP address.

The `exchange-ip-addr6` option allows a loopback IPv6 address to be exchanged between the spokes and the hub in a network. This means that instead of using the tunnel IP address, which is typically used for communication, the loopback IPv6 address is used.

See config router bgp and config router route-map in the CLI Reference for a comprehensive list of commands.

To configure the hub:

1. Configure the BGP settings:

```
config router bgp
    set as 65100
    set router-id 10.200.1.1
    set keepalive-timer 5
    set holdtime-timer 15
    set ibgp-multipath enable
    set network-import-check disable
    set additional-path6 enable
    set additional-path-vpnv6 enable
    set additional-path-select6 4
    config neighbor-group
        edit "EDGEv6"
            set advertisement-interval 1
            set activate disable
            set activate-vpnv4 disable
            set capability-graceful-restart enable
            set next-hop-self-rr6 enable
            set soft-reconfiguration6 enable
            set remote-as 65100
            set update-source "vd11-lo1"
            set additional-path6 both
            set adv-additional-path6 4
            set route-reflector-client6 enable
            set route-reflector-client-vpnv6 enable
        next
    end
    config neighbor-range6
        edit 2
            set prefix6 2001::10:200:1:0/112
            set neighbor-group "EDGEv6"
        next
    end
    config network6
        edit 1
            set prefix6 2001::10:200:1:0/112
        next
    end
    config vrf6
        edit "0"
            set role pe
        next
        edit "1"
            set role ce
            set rd "1:1"
```

```
                set export-rt "1:1"
                set import-rt "1:1"
            next
            edit "2"
                set role ce
                set rd "2:1"
                set export-rt "2:1"
                set import-rt "2:1"
            next
        end
    end
```

2. Configure the IPsec phase 1 interface settings:

```
config vpn ipsec phase1-interface
    edit "p1"
        set type dynamic
        set interface "vd11-vlan1"
        set ike-version 2
        set peertype any
        set net-device disable
        set exchange-ip-addr6 2001::10:200:1:1
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dpd on-idle
        set npu-offload disable
        set dhgrp 5
        set auto-discovery-sender enable
        set encapsulation vpn-id-ipip
        set psksecret **********
        set dpd-retryinterval 60
    next
    edit "p111"
        set type dynamic
        set interface "vd11-vlan111"
        set ike-version 2
        set peertype any
        set net-device disable
        set exchange-ip-addr6 2001::10:200:1:1
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dpd on-idle
        set npu-offload disable
        set dhgrp 5
        set auto-discovery-sender enable
        set encapsulation vpn-id-ipip
        set psksecret **********
        set dpd-retryinterval 60
    next
end
```

3. Configure the IPsec phase 2 interface settings:

```
config vpn ipsec phase2-interface
    edit "p1-v6"
```

```
        set phase1name "p1"
        set proposal aes128-sha1
        set replay disable
        set src-addr-type subnet6
        set dst-addr-type subnet6
    next
    edit "p111-v6"
        set phase1name "p111"
        set proposal aes128-sha1
        set replay disable
        set src-addr-type subnet6
        set dst-addr-type subnet6
    next
end
```

To configure a spoke:

1. Configure the BGP settings:

```
config router bgp
    set as 65100
    set router-id 10.200.1.100
    set keepalive-timer 5
    set holdtime-timer 15
    set ibgp-multipath enable
    set additional-path6 enable
    set additional-path-vpnv6 enable
    set recursive-next-hop enable
    set tag-resolve-mode merge
    set graceful-restart enable
    set additional-path-select6 4
    config neighbor
        edit "2001::10:200:1:1"
            set advertisement-interval 1
            set activate disable
            set activate-vpnv4 disable
            set capability-dynamic enable
            set capability-graceful-restart6 enable
            set capability-graceful-restart-vpnv6 enable
            set soft-reconfiguration6 enable
            set remote-as 65100
            set route-map-in6 "tag"
            set route-map-in-vpnv6 "tag"
            set connect-timer 10
            set update-source "vd2-lo1"
            set additional-path6 both
            set additional-path-vpnv6 both
        next
    end
    config network6
        edit 1
            set prefix6 2001:22::/64
        next
        edit 2
            set prefix6 2001::10:200:1:100/128
        next
```

```
    end
    config vrf6
        edit "0"
            set role pe
        next
        edit "1"
            set role ce
            set rd "1:1"
            set export-rt "1:1"
            set import-rt "1:1"
        next
        edit "2"
            set role ce
            set rd "2:1"
            set export-rt "2:1"
            set import-rt "2:1"
        next
    end
end
```

2. Configure the IPsec phase 1 interface settings:

```
config vpn ipsec phase1-interface
    edit "vd2-1"
        set interface "vd2-vlan12"
        set ike-version 2
        set peertype any
        set net-device enable
        set exchange-ip-addr6 2001::10:200:1:100
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set npu-offload disable
        set dhgrp 5
        set auto-discovery-receiver enable
        set encapsulation vpn-id-ipip
        set remote-gw 11.1.1.11
        set psksecret **********
    next
    edit "vd2-2"
        set interface "vd2-vlan112"
        set ike-version 2
        set peertype any
        set net-device enable
        set exchange-ip-addr6 2001::10:200:1:100
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set npu-offload disable
        set dhgrp 5
        set auto-discovery-receiver enable
        set encapsulation vpn-id-ipip
        set remote-gw 111.1.1.11
        set psksecret **********
    next
end
```

3. Configure the IPsec phase 2 interface settings:

```
config vpn ipsec phase2-interface
    edit "vd2-1-6"
        set phase1name "vd2-1"
        set proposal aes128-sha1
        set dhgrp 5
        set replay disable
        set auto-negotiate enable
        set src-addr-type subnet6
        set dst-addr-type subnet6
    next
    edit "vd2-2-6"
        set phase1name "vd2-2"
        set proposal aes128-sha1
        set dhgrp 5
        set replay disable
        set auto-negotiate enable
        set src-addr-type subnet6
        set dst-addr-type subnet6
    next
end
```

4. Configure the SD-WAN settings:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "vd2-1"
            set cost 10
        next
        edit 2
            set interface "vd2-2"
            set cost 20
        next
    end
    config health-check
        edit "ping6"
            set addr-mode ipv6
            set server "2001::10:200:1:1"
            set source6 2001::10:200:1:100
            set members 1 2
            config sla
                edit 1
                next
            end
        next
    end
    config service
        edit 61
            set addr-mode ipv6
            set priority-members 1
```

```
                set dst6 "6001-100"
            next
            edit 62
                set addr-mode ipv6
                set priority-members 2
                set dst6 "6100-200"
            next
        end
    end
```

To check the spoke 1 routes:

```
# get router info6 routing-table bgp
Routing table for VRF=0
B       2001::10:200:1:0/112 [200/0] via 2001::10:200:1:1 tag 100 (recursive via vd2-1
tunnel ::11.1.1.11), 1d15h41m
(recursive via vd2-2 tunnel ::111.1.1.11), 1d15h41m, [1024/0]
B       2001::10:200:1:101/128 [200/0] via 2001::10:200:1:1 tag 100 (recursive via vd2-1
tunnel ::11.1.1.11), 1d15h41m
(recursive via vd2-2 tunnel ::111.1.1.11), 1d15h41m, [1024/0]

Routing table for VRF=1
B V     2001:33::/64 [200/0] via 2001::10:200:1:101 tag 100 (recursive via vd2-1 tunnel
::11.1.1.11), 1d15h41m
(recursive via vd2-2 tunnel ::111.1.1.11), 1d15h41m, [1024/0]

Routing table for VRF=2
B V     2001:33::/64 [200/0] via 2001::10:200:1:101 tag 100 (recursive via vd2-1 tunnel
::11.1.1.11), 1d15h41m
(recursive via vd2-2 tunnel ::111.1.1.11), 1d15h41m, [1024/0]
```

To test the configuration on shortcut 1:

1. From VRF1 of spoke 1, ping VRF1 of spoke 2.
2. From VRF2 of spoke 1, ping VRF2 spoke 2. Both VRF1 and VRF2 source and destination IP addresses are the same, so you can see how the traffic is isolated.
3. Verify the session list:

```
# diagnose sys session6 list
session6 info: proto=58 proto_state=00 duration=3 expire=59 timeout=0 refresh_dir=both
flags=00000000 sockport=0 socktype=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=may_dirty
statistic(bytes/packets/allow_err): org=416/4/0 reply=416/4/0 tuples=2
tx speed(Bps/kbps): 136/1 rx speed(Bps/kbps): 136/1
orgin->sink: org pre->post, reply pre->post dev=100->223/223->100
hook=pre dir=org act=noop 2001:22::55:398->2001:33::44:128(:::0)
hook=post dir=reply act=noop 2001:33::44:398->2001:22::55:129(:::0)
src_mac=02:4c:a5:fc:77:6f
misc=0 policy_id=1 pol_uuid_idx=1070 auth_info=0 chk_client_info=0 vd=3:2
serial=0001104d tos=ff/ff ips_view=0 app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=61
```

```
rpdb_link_id=ff00003d ngfwid=n/a
npu_state=0x1040001 no_offload
no_ofld_reason:  disabled-by-policy non-npu-intf
total session6: 1
```

In the output, `vd=<vdom_ID>:<VRF_ID>` indicates that sessions are created in and stay in the corresponding VRFs.

SD-WAN hub and spoke speed test improvements - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- Hub and spoke speed tests

SD-WAN hub and spoke speed tests include the following improvements:

- Speed test servers can be deployed on a hub or a spoke. When deployed on a hub, speed tests can be initiated from spokes, even when a spoke is behind a NAT device.
- Tests can be in upload or download direction.
- Both TCP and UDP protocols are supported.
- An egress-shaping profile can be applied to local, remote, or both local and remote IPsec tunnels or no IPsec tunnels.
- Custom speed-test listening ports can be configured.

The test measures the speeds of the link to each spoke so that QoS can be applied on the hub to the dynamic IPsec overlay tunnels to each spoke. When the speed test is initiated from the spoke, the results are cached on the spoke, but sent to the hub to be applied to the egress traffic shaping profile assigned to the IPsec overlay tunnel interface and the respective tunnel. For more information about SD-WAN hub and spoke speed tests, see Running speed tests from the hub to the spokes in dial-up IPsec tunnels.

When a speed-test server is enabled, two speed test daemons are started and listen on different ports for different purposes:

- The controller speed test daemon listens on the IPsec overlay interfaces to assign an access token to each incoming speed test for authentication.
- The speed test daemon listens on the IPsec underlay interfaces to handle the speed tests.

Each incoming speed test request must present the obtained access token to prevent random, unauthorized requests. Otherwise, the connection is closed immediately. As such, speed test access must be enabled on both the underlay and the IPsec overlay tunnel interfaces on the hub.

```
config system interface
    edit <interface>
        set allowaccess speed-test [other access] ...
    next
end
```

If the IPsec tunnel has a configured `exchange-ip`, speed test access must also be configured on the associated interface, such as the loopback interface.

New commands are available to configure custom speed-test listening ports for the speed test server:

```
config system global
    set speedtestd-server-port <integer>
    set speedtestd-ctrl-port <integer>
end
```

`set speedtestd-server-port <integer>`	Specify a custom port number (1024 - 65535, default = 5201) for the speed test daemon. The port is used to perform the speed test.
`set speedtestd-ctrl-port <integer>`	Specify a custom port number (1024 - 65535. default = 5200) for the controller speed test daemon. The port is used to assign access tokens for authentication prior to performing the speed test.

The speed test client can be a hub or a spoke and must have `system speed-test-schedule` configured and the `dynamic-server` setting enabled.

On the speed test client, specify whether and how to apply the test results in a shaping profile. The shaping profile must be configured in the phase1 interface before it can be used with a speed test.

```
config system speed-test-schedule
    edit <interface>
        set server-port <integer>
        set ctrl-port <integer>
        set update-shaper {disable | local | remote | both}
    next
end
```

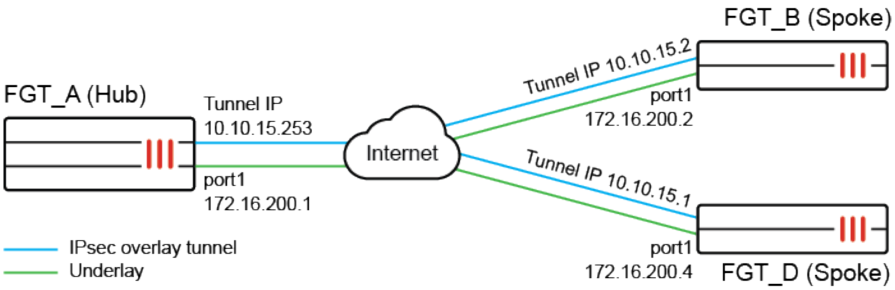
`set server-port <integer>`	Specify the port number for the speed-test server used for speed tests (1 - 65535, default = 5201).
`set ctrl-port <integer>`	Specify the port number for the controller on the speed-test server used for authentication (1 - 65535, default = 5200).
`set update-shaper {disable | local | remote | both}`	Set the egress shaper to use the speed test results: • `disable`: Disable updating the egress shaper (default). • `local`: Update the speed-test client egress shaper. • `remote`: Update the speed-test server egress shaper. • `both`: Update both the local and remote egress shapers.

Example

In this hub and spoke example, the hub is configured as an IPsec VPN dial-up server with two IPsec tunnels, and each tunnel is connected to a spoke. The VPN interfaces and IP addresses are:

FortiGate	Interface	IP Address
FGT_A (Hub)	hub-phase1	10.10.15.253
FGT_B (Spoke)	spoke11-p1	10.10.15.2
FGT_D (Spoke)	spoke21-p1	10.10.15.1

The hub (FGT_A) is configured as a speed-test server to listen on custom ports (6000 and 7000), and the spokes (FGT_B and FGT_D) are configured as speed-test clients. This setup allows speed tests to successfully perform when spokes are behind NAT devices. The results of the speed test will be applied to the hub-phase1 overlay tunnel(s) as specified by the speed-test clients.



The spokes are configured to initiate speed tests on a schedule on UDP. After the speed test completes, the results are sent to the hub, and the hub applies the results on its IPsec tunnels as egress traffic shaping. The results are also cached and can be used if an IPsec tunnel is disconnected and reconnected again.

This example focuses on the key settings required to enable a hub as the speed-test server and the spokes as speed-test clients that initiate the speed tests. For a complete example about running speed tests from the hub, see Running speed tests from the hub to the spokes in dial-up IPsec tunnels.

To configure the hub FortiGate (FGT_A):

1. Configure a shaping profile:
   In this example, the shaping profile is named profile_1.

```
config firewall shaping-profile
    edit "profile_1"
        set default-class-id 2
        config shaping-entries
            edit 1
                set class-id 2
                set priority low
                set guaranteed-bandwidth-percentage 10
                set maximum-bandwidth-percentage 10
            next
            edit 2
                set class-id 3
                set priority medium
                set guaranteed-bandwidth-percentage 30
                set maximum-bandwidth-percentage 40
            next
            edit 3
                set class-id 4
                set guaranteed-bandwidth-percentage 20
                set maximum-bandwidth-percentage 60
            next
        end
    end
end
```

Three classes are used in the profile for low, medium, and high priority traffic. Each class is assigned a guaranteed and maximum bandwidth as a percentage of the measured bandwidth from the speed test.

2. Configure a shaping policy to assign certain traffic as a class ID:

   In this example, all traffic destined to the dialup tunnels are assigned class 3.

```
config firewall shaping-policy
    edit 2
        set service "ALL"
        set schedule "always"
        set dstintf "hub-phase1" "hub2-phase1"
        set class-id 3
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

3. Enable a speed test server with custom speed-test listening ports:

   A speed test server is enabled on the hub. Port 7000 will run speed tests, and port 6000 will be the controller used to issue access tokens for speed test authentication.

```
config system global
    ...
    set speedtest-server enable
    set speedtestd-ctrl-port 6000
    set speedtestd-server-port 7000
end
```

4. Allow the speed test on the underlay:

```
config system interface
    edit "port1"
        set ip 172.16.200.1 255.255.255.0
        set allowaccess ping https ssh snmp http telnet fgfm radius-acct probe-response 
fabric speed-test
        ...
    next
end
```

5. Allow the speed test on the overlay and use the shaping profile in the interface:

   In this example, speed tests are allowed on the overlay, and the shaping profile (profile_1) is used on the hub phase1 interface (port1).

```
config system interface
    edit "hub-phase1"
        set ip 10.10.15.253 255.255.255.255
        set allowaccess ping speed-test
        set egress-shaping-profile "profile_1"
        ...
        set interface "port1"
    next
end
```

To configure the first spoke FortiGate (FGT_B):

1. Configure `system speed-test-schedule`:

   The protocol mode is set to UDP. The custom controller port used for authentication is set to 6000, and the custom port used to run the speed tests is set to 7000. The shaping profile is set to remote.

```
config system speed-test-schedule
    edit "spoke11-p1"
        set mode UDP
        set schedules "1"
        set dynamic-server enable
        set ctrl-port 6000
        set server-port 7000
        set update-shaper remote
    next
end
```

2. Configure a recurring schedule for the speed tests:

   Schedule 1 is set to start at 08:37 every day of the week.

```
config firewall schedule recurring
    edit "1"
        set start 08:37
        set day sunday monday tuesday wednesday thursday friday saturday
    next
end
```

To configure the second spoke FortiGate (FGT_D):

1. Configure a speed test schedule:

   The protocol mode is set to UDP. The custom controller port used for authentication is set to 6000, and the custom port used to run the speed tests is set to 7000. The shaping profile is set to remote.

```
config system speed-test-schedule
    edit "spoke21-p1"
        set mode UDP
        set schedules "1"
        set dynamic-server enable
        set ctrl-port 6000
        set server-port 7000
        set update-shaper remote
    next
end
```

2. Configure a recurring schedule for the speed tests:

   Schedule 1 is set to start at 08:37 every day of the week.

```
config firewall schedule recurring
    edit "1"
        set start 08:37
        set day sunday monday tuesday wednesday thursday friday saturday
    next
end
```

To view the speed test results:

1. After the speed test schedule runs, view the result on spoke FGT_B:
   On spoke FGT_B, authentication succeeds through port 6000, and the test runs on port 7000. UDP mode is used, and the test is successful.

```
# diagnose debug application speedtest -1

......
fcron_speedtest_ipsec_request_init()-464: root: spoke11-p1(spoke11-p1) id=003900d5
fd=24, init request=0.0.0.0:0 -> 10.10.15.253:6000, test=172.16.200.2:0 ->
172.16.200.1:7000: succeed.
......
[speedtest(2181)] start uploading test.
[speedtest(2181)] Connecting to host 172.16.200.1, port 7000
[speedtest(2181)] [ 26] local 172.16.200.2 port 17553 connected to 172.16.200.1 port
7000
[speedtest(2181)] [ ID] Interval           Transfer     Bitrate         Total Datagrams
[speedtest(2181)] [ 26]   0.00-1.00   sec   150 MBytes  1.26 Gbits/sec  107570
[speedtest(2181)] [ 26]   1.00-2.00   sec   149 MBytes  1.25 Gbits/sec  107120
[speedtest(2181)] [ 26]   2.00-3.00   sec   149 MBytes  1.25 Gbits/sec  107030
[speedtest(2181)] [ 26]   3.00-4.00   sec   149 MBytes  1.25 Gbits/sec  107210
[speedtest(2181)] [ 26]   4.00-5.00   sec   149 MBytes  1.25 Gbits/sec  107260
[speedtest(2181)] [ ID] Interval           Transfer     Bitrate         Jitter
 Lost/Total Datagrams
[speedtest(2181)] [ 26]   0.00-5.00   sec   747 MBytes  1.25 Gbits/sec  0.000 ms
 0/536190 (0%)  sender
[speedtest(2181)] [ 26]   0.00-5.00   sec   271 MBytes   454 Mbits/sec  0.000 ms
 341627/535995 (64%)  receiver
[speedtest(2181)] client(sender): bytes_recv=283777280, bytes_sent=782837400, sender_
time=5.000, recver_time=5.000
[speedtest(2181)] client(sender): up_speed:  454 Mbits/sec
[speedtest(2181)]
[speedtest(2181)] speed test Done.
[speedtest(2181)] start downloading test.
[speedtest(2181)] Connecting to host 172.16.200.1, port 7000
[speedtest(2181)] Reverse mode, remote host 172.16.200.1 is sending
[speedtest(2181)] [ 26] local 172.16.200.2 port 7998 connected to 172.16.200.1 port 7000
[speedtest(2181)] [ ID] Interval           Transfer     Bitrate         Jitter
 Lost/Total Datagrams
[speedtest(2181)] [ 26]   0.00-1.00   sec  54.6 MBytes   458 Mbits/sec  0.007 ms
 70745/109978 (64%)
[speedtest(2181)] [ 26]   1.00-2.00   sec  54.8 MBytes   460 Mbits/sec  0.008 ms
 67547/106917 (63%)
[speedtest(2181)] [ 26]   2.00-3.00   sec  54.9 MBytes   460 Mbits/sec  0.010 ms
 67543/106940 (63%)
[speedtest(2181)] [ 26]   3.00-4.00   sec  54.8 MBytes   460 Mbits/sec  0.006 ms
 67636/107024 (63%)
[speedtest(2181)] [ 26]   4.00-5.00   sec  54.9 MBytes   460 Mbits/sec  0.004 ms
 67421/106842 (63%)
[speedtest(2181)] [ ID] Interval           Transfer     Bitrate         Jitter
 Lost/Total Datagrams
[speedtest(2181)] [ 26]   0.00-5.00   sec   750 MBytes  1.26 Gbits/sec  0.000 ms
 0/538540 (0%)  sender
[speedtest(2181)] [ 26]   0.00-5.00   sec   274 MBytes   460 Mbits/sec  0.004 ms
 340892/537701 (63%)  receiver
```

```
[speedtest(2181)] client(recver): bytes_recv=287341140, bytes_sent=786268400, sender_
time=5.000, recver_time=5.001
[speedtest(2181)] client(recver): down_speed:  460 Mbits/sec
[speedtest(2181)]
[speedtest(2181)] speed test Done.
fcron_speedtest_notify_func()-1275: Speed test pid=2181 done

fcron_speedtest_on_test_finish()-1211: Test 3900d5 for 'spoke11-p1' succeed with
up=454043, down=459694
fcron_speedtest_save_results()-1144: Write logs to disk: succ=1, fail=0
fcron_speedtest_sync_results()-1172: Sync cached results to secondary devices.
```

2. After the speed test schedule runs, view the result on the spoke FGT_D:

   On spoke FGT_D, authentication succeeds through port 6000, and the test runs on port 7000. UDP mode is used, and the test is successful.

```
# diagnose debug application speedtest -1

......
fcron_speedtest_ipsec_request_init()-464: root: spoke21-p1(spoke21-p1) id=00380011
fd=25, init request=0.0.0.0:0 -> 10.10.15.253:6000, test=172.16.200.4:0 ->
172.16.200.1:7000: succeed.
......
[speedtest(4309)] start uploading test.
[speedtest(4309)] Connecting to host 172.16.200.1, port 7000
[speedtest(4309)] [ 27] local 172.16.200.4 port 15349 connected to 172.16.200.1 port
7000
[speedtest(4309)] [ ID] Interval           Transfer     Bitrate         Total Datagrams
[speedtest(4309)] [ 27]   0.00-1.00   sec   148 MBytes  1.24 Gbits/sec  105940
[speedtest(4309)] [ 27]   1.00-2.00   sec   148 MBytes  1.24 Gbits/sec  105990
[speedtest(4309)] [ 27]   2.00-3.00   sec   147 MBytes  1.24 Gbits/sec  105860
[speedtest(4309)] [ 27]   3.00-4.00   sec   148 MBytes  1.24 Gbits/sec  105960
[speedtest(4309)] [ 27]   4.00-5.00   sec   148 MBytes  1.24 Gbits/sec  106090
[speedtest(4309)] [ ID] Interval           Transfer     Bitrate         Jitter
 Lost/Total Datagrams
[speedtest(4309)] [ 27]   0.00-5.00   sec   738 MBytes  1.24 Gbits/sec  0.000 ms
 0/529840 (0%)  sender
[speedtest(4309)] [ 27]   0.00-5.00   sec   271 MBytes   454 Mbits/sec  0.000 ms
 335130/529650 (63%)  receiver
[speedtest(4309)] client(sender): bytes_recv=283999200, bytes_sent=773566400, sender_
time=5.000, recver_time=5.000
[speedtest(4309)] client(sender): up_speed:  454 Mbits/sec
[speedtest(4309)]
[speedtest(4309)] speed test Done.
[speedtest(4309)] start downloading test.
[speedtest(4309)] Connecting to host 172.16.200.1, port 7000
[speedtest(4309)] Reverse mode, remote host 172.16.200.1 is sending
[speedtest(4309)] [ 27] local 172.16.200.4 port 19586 connected to 172.16.200.1 port
7000
[speedtest(4309)] [ ID] Interval           Transfer     Bitrate         Jitter
 Lost/Total Datagrams
[speedtest(4309)] [ 27]   0.00-1.00   sec  56.1 MBytes   471 Mbits/sec  0.005 ms
 70258/110574 (64%)
[speedtest(4309)] [ 27]   1.00-2.00   sec  56.0 MBytes   470 Mbits/sec  0.006 ms
 66496/106740 (62%)
[speedtest(4309)] [ 27]   2.00-3.00   sec  56.0 MBytes   470 Mbits/sec  0.005 ms
```

```
 66481/106736 (62%)
[speedtest(4309)] [ 27]   3.00-4.00   sec  56.1 MBytes   471 Mbits/sec  0.007 ms
 66403/106690 (62%)
[speedtest(4309)] [ 27]   4.00-5.00   sec  56.3 MBytes   473 Mbits/sec  0.008 ms
 65991/106454 (62%)
[speedtest(4309)] [ ID] Interval           Transfer     Bitrate         Jitter
 Lost/Total Datagrams
[speedtest(4309)] [ 27]   0.00-5.00   sec   749 MBytes  1.26 Gbits/sec  0.000 ms
 0/538110 (0%)  sender
[speedtest(4309)] [ 27]   0.00-5.00   sec   281 MBytes   471 Mbits/sec  0.008 ms
 335629/537194 (62%)  receiver
[speedtest(4309)] client(recver): bytes_recv=294284900, bytes_sent=785640600, sender_
time=5.000, recver_time=5.001
[speedtest(4309)] client(recver): down_speed:  471 Mbits/sec
[speedtest(4309)]
[speedtest(4309)] speed test Done.
fcron_speedtest_notify_func()-1275: Speed test pid=4309 done

fcron_speedtest_on_test_finish()-1211: Test 380011 for 'spoke21-p1' succeed with
up=454398, down=470794
fcron_speedtest_save_results()-1144: Write logs to disk: succ=1, fail=0
fcron_speedtest_sync_results()-1172: Sync cached results to secondary devices.
```

3. After the speed test schedule runs, view the result on the hub (FGT_A):

> The server side uses `speedtestd`, while the client side uses `speedtest`.

The speed test results are applied on hub-phase1_0 and hub_phase1_1 as egress traffic shaping.

```
# diagnose debug application speedtestd -1

......
[speedtest(2771)] [  7] local 172.16.200.1 port 7000 connected to 172.16.200.2 port
17553
......
[speedtest(2771)] [  7] local 172.16.200.1 port 7000 connected to 172.16.200.2 port 7998
......
[sptestd::ctrl(0377):root] set shaper: if=hub-phase1, tun=hub-phase1_0, sp=profile_1,
bw=459745
......
[speedtest(2771)] [  7] local 172.16.200.1 port 7000 connected to 172.16.200.4 port
15349
......
[speedtest(2771)] [  7] local 172.16.200.1 port 7000 connected to 172.16.200.4 port
19586
......
[sptestd::ctrl(0377):root] set shaper: if=hub-phase1, tun=hub-phase1_1, sp=profile_1,
bw=470855
......
```

4. Verify the result is cached on the spokes.

- On FGT_B, the speed test results are cached:

```
#  diagnose test application forticron 10
Speed test results:
1: vdom=root, phase1intf=spoke11-p1, peer-id='172.16.200.1', up=454043, dw=459694,
time=12/13 12:32:19
```

- On FGT_D, the speed test results are cached:

```
#   diagnose test application forticron 10
Speed test results:
1: vdom=root, phase1intf=spoke21-p1, peer-id='172.16.200.1', up=454398, dw=470794,
time=12/12 16:33:18
```

5. On the hub (FGT_A), verify the speed test results are applied to the hub's IPsec tunnels as egress traffic shaping: On hub-phase1_0 and hub-phase1_1, the correct traffic control is displayed.

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
......
------------------------------------------------------
name=hub-phase1_0 ver=2 serial=16 172.16.200.1:0->172.16.200.2:0 tun_id=10.10.15.1 tun_
id6=2000:10:10:15::1 dst_mtu=1500 dpd-link=on weight=1
bound_if=11 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74408 options[122a8]=npu
rgwy-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=10

parent=hub-phase1 index=0
......
egress traffic control:
        bandwidth=459745(kbps) lock_hit=0 default_class=2 n_active_class=3
        class-id=2      allocated-bandwidth=45974(kbps)         guaranteed-
bandwidth=45974(kbps)
                        max-bandwidth=45974(kbps)       current-bandwidth=0(kbps)
                        priority=low    forwarded_bytes=86K
                        dropped_packets=0       dropped_bytes=0
        class-id=3      allocated-bandwidth=137923(kbps)        guaranteed-
bandwidth=137923(kbps)
                        max-bandwidth=183897(kbps)      current-bandwidth=0(kbps)
                        priority=medium         forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=4      allocated-bandwidth=275846(kbps)        guaranteed-
bandwidth=91948(kbps)
                        max-bandwidth=275846(kbps)      current-bandwidth=0(kbps)
                        priority=high   forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
------------------------------------------------------
name=hub-phase1_1 ver=2 serial=17 172.16.200.1:0->172.16.200.4:0 tun_id=10.10.15.2 tun_
id6=2000:10:10:15::2 dst_mtu=1500 dpd-link=on weight=1
bound_if=11 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74408 options[122a8]=npu
rgwy-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=10

parent=hub-phase1 index=1
......
egress traffic control:
        bandwidth=470855(kbps) lock_hit=0 default_class=2 n_active_class=3
        class-id=2      allocated-bandwidth=47085(kbps)         guaranteed-
bandwidth=47085(kbps)
                        max-bandwidth=47085(kbps)       current-bandwidth=0(kbps)
```

```
                        priority=low    forwarded_bytes=81K
                        dropped_packets=0       dropped_bytes=0
        class-id=3      allocated-bandwidth=141256(kbps)        guaranteed-
bandwidth=141256(kbps)
                        max-bandwidth=188341(kbps)      current-bandwidth=0(kbps)
                        priority=medium         forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=4      allocated-bandwidth=282512(kbps)        guaranteed-
bandwidth=94170(kbps)
                        max-bandwidth=282512(kbps)      current-bandwidth=0(kbps)
                        priority=high   forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
```

ADVPN 2.0 edge discovery and path management - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:
- ADVPN 2.0 edge discovery and path management

The SD-WAN with ADVPN solution has evolved to version 2.0 with major changes to ADVPN design and operation, including the introduction of edge discovery and path management for ADVPN spokes.

ADVPN 2.0 incorporates intelligence into the spokes to ensure shortcut tunnels (also known as shortcuts) are established using underlays available on both spokes and chosen based on matching certain link health criteria.

ADVPN 2.0 provides a more flexible SD-WAN solution than the original ADVPN to achieve resiliency against underlay outages or degraded underlay performance because it no longer depends on specific BGP routing designs or mechanisms, including route reflection, BGP next hop recursive resolution, BGP per overlay, and BGP on loopback.

ADVPN 2.0 only supports IPv4.

The topic includes the following sections:

- Overview on page 190
- Example on page 192

Overview

The overview covers the following information:

- How this solution differs from SD-WAN with previous ADVPN on page 190
- SD-WAN CLI configuration commands on page 191

How this solution differs from SD-WAN with previous ADVPN

With the previous version of ADVPN and SD-WAN, shortcut path selection relied entirely on the overlays between the spokes. The hub and overlays were used to exchange IKE shortcut messages, and policy routes were configured on the hub to ensure shortcuts were established on the same overlay. In addition, user traffic was needed to trigger the process of establishing shortcuts.

With the latest version of ADVPN and SD-WAN, shortcut path selection is achieved through edge discovery and path management functionality on the ADVPN spokes.

1. Edge discovery:
    - Expand IKE Shortcut-Reply message to allow the local spoke (spoke where user traffic is initiated) to obtain the remote spoke (destination spoke for user traffic) WAN link information, which includes IP address, transport group, link quality, link cost, and member configuration order.
    - After shortcut establishment, WAN link information can be exchanged on the shortcut regularly every 5 seconds through UDP traffic. The path management function on the local spoke is regularly updated to pick up changes to remote or local overlays and select the best shortcut path accordingly.
2. Path management:
    - The local spoke handles the remote spoke WAN link information, calculates the best shortcut path per SD-WAN service or rule, and then advises IKE to establish a shortcut using the selected path.

SD-WAN CLI configuration commands

The following SD-WAN CLI configuration commands are used to configure ADVPN 2.0 on the spokes:

```
config system sdwan
    config zone
        edit <zone-name>
            set advpn-select {enable | disable}
            set advpn-health-check <health-check name>
        next
    end
    config members
        edit <integer>
            set transport-group <integer>
        next
    end
    config service
        edit <integer>
            set shortcut-priority {enable | disable | auto}
        next
    end
end
```

`set advpn-select {enable	disable}`	Enable or disable SDWAN/ADVPN-2.0 (default=disabled).
`set advpn-health-check <health-check name>`	Specify the health check for the spoke whose info will be sent to the peer spoke.
`set transport-group <integer>`	Specify different group ID between (1 -255) to differentiate link-type, such as Internet, MPLS, LTE, Satellite.
`set shortcut-priority {enable	disable	auto}`	Enable or disable making ADVPN shortcut a high priority over overlay parent interfaces, if SLA mode or link cost factor mode conditions are met: • `enable`: enable a high priority of ADVPN shortcut for this service. • `disable`: disable a high priority of ADVPN shortcut for this service. • `auto`: automatically enable a high priority of ADVPN shortcut for this service

	if ADVPN2.0 is enabled.
`diagnose sys sdwan advpn-session`	Diagnostic command run on local spoke to view remote spoke WAN link information and path manager shortcut path selection.

As with the previous version of ADVPN, on the hub, you must enable ADVPN and configure firewall policies between spokes.

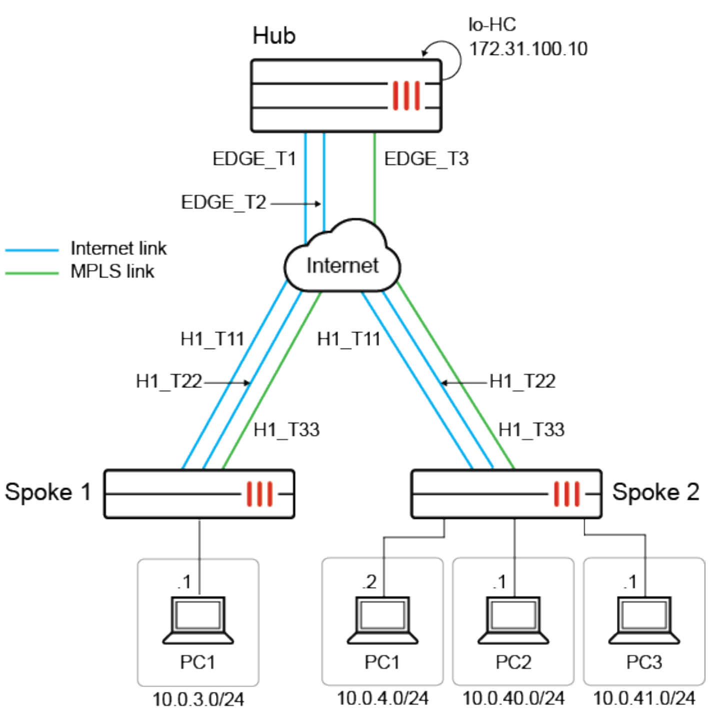
Example

The configuration example illustrates the edge discovery and path management processes for a typical hub and spoke topology. This example focuses on SD-WAN configuration for steering traffic and establishing shortcuts in the direction from Spoke 1 to Spoke 2.

- Network Topology on page 192
- SD-WAN configuration and health check status on page 193
- Scenario 1: Traffic matching SD-WAN rule 1 on page 196
- Scenario 2: Traffic matching SD-WAN rule 2 on page 197
- Scenario 3: Traffic matching SD-WAN rule 3 on page 199
- Scenario 4: Spoke 2 H1_T22 overlay link out-of-SLA on page 200

Network Topology

In this example, BGP per overlay was used for dynamic routing to distribute the LAN routes behind each spoke to the other spoke. However, this was a design choice. You can also use BGP on loopback for this example.



Spokes 1 and 2 have the following VPN overlays between themselves and the hub:

VPN Overlays	IP address on Spoke 1	IP address on Spoke 2
H1_T11	172.31.80.1/32	172.31.80.2/32
H1_T22	172.31.81.1/32	172.31.81.2/32
H1_T33	172.31.82.1/32	172.31.82.2/32

SD-WAN Rules/Services defined on Spoke 1:

	SD-WAN Rule/Service 1	SD-WAN Rule/Service 2	SD-WAN Rule/Service 3
	H1_T11	H1_T22	H1_T33
	H1_T22	H1_T11	H1_T11
	H1_T33	H1_T33	H1_T22
Strategy for choosing outgoing interfaces	Lowest cost (SLA)	Lowest cost (SLA)	Best quality, link cost factor: packet loss

Throughout this example, transport group 1 is used for VPN overlays over Internet links while transport group 2 is used for the VPN overlay over an MPLS link.

In this example, user traffic is initiated behind Spoke 1 and destined to Spoke 2. Because of this, Spoke 1 is considered the local spoke, and Spoke 2 is considered the remote spoke.

SD-WAN configuration and health check status

SD-WAN configuration and health check status on Spoke 1:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "overlay"
            set advpn-select enable
            set advpn-health-check "HUB"
        next
    end
    config members
        edit 1
            set interface "H1_T11"
            set zone "overlay"
            set transport-group 1
        next
        edit 2
            set interface "H1_T22"
            set zone "overlay"
            set transport-group 1
        next
        edit 3
            set interface "H1_T33"
            set zone "overlay"
            set transport-group 2
```

```
        next
    end
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set members 1 2 3
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
    config service
        edit 1
            set name "1"
            set mode sla
            set shortcut-priority enable
            set dst "spoke-2_LAN-1" "Tunnel_IPs"
            set src "spoke-1_LAN-1" "Tunnel_IPs"
            config sla
                edit "HUB"
                    set id 1
                next
            end
            set priority-members 1 2 3
        next
        edit 2
            set name "2"
            set mode sla
            set shortcut-priority enable
            set dst "spoke-2_LAN-2" "Tunnel_IPs"
            set src "spoke-1_LAN-1" "Tunnel_IPs"
            config sla
                edit "HUB"
                    set id 1
                next
            end
            set priority-members 2 1 3
        next
        edit 3
            set name "3"
            set mode priority
            set dst "spoke-2_LAN-3" "Tunnel_IPs"
            set src "spoke-1_LAN-1" "Tunnel_IPs"
            set health-check "HUB"
            set link-cost-factor packet-loss
            set priority-members 3 1 2
        next
    end
end
```

```
# diagnose sys sdwan health-check
Health Check(HUB):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.231), jitter(0.029), mos(4.404), 
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
```

```
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(0.193), jitter(0.010), mos(4.404),
bandwidth-up(999994), bandwidth-dw(999997), bandwidth-bi(1999991) sla_map=0x1
Seq(3 H1_T33): state(alive), packet-loss(0.000%) latency(0.144), jitter(0.007), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
```

SD-WAN configuration and health check status on Spoke 2:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "overlay"
            set advpn-select enable
            set advpn-health-check "HUB"
        next
    end
    config members
        edit 1
            set interface "H1_T11"
            set zone "overlay"
            set cost 100
            set transport-group 1
        next
        edit 2
            set interface "H1_T22"
            set zone "overlay"
            set transport-group 1
        next
        edit 3
            set interface "H1_T33"
            set zone "overlay"
            set transport-group 2
        next
    end
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set members 3 1 2
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
end
```

```
# diagnose sys sdwan health-check
Health Check(HUB):
Seq(3 H1_T33): state(alive), packet-loss(0.000%) latency(0.124), jitter(0.009), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.216), jitter(0.043), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(0.184), jitter(0.012), mos(4.404),
bandwidth-up(999994), bandwidth-dw(999998), bandwidth-bi(1999992) sla_map=0x1
```

Scenario 1: Traffic matching SD-WAN rule 1

In this scenario, PC 1 connected to Spoke 1 initiates an ICMP ping destined for PC1 connected to Spoke 2. Therefore, this user traffic matches SD-WAN rule 1 and triggers shortcut path selection and establishment.

The Path Manager of Spoke 1 will calculate the best shortcut path by comparing transport group, link quality (for SLA mode), link cost, and member configuration order between Spoke 1 and Spoke 2.

For an SLA mode service, the following algorithm is followed for considering endpoints of the best shortcut path:

1. Overlays with the same transport group
2. In-SLA overlays
3. Lowest link cost overlays
4. Member configuration order as a final tiebreaker

Based on this algorithm, the Path Manager on Spoke 1 selects Spoke 1 H1_T11 because:

It is first in the priority-members order for SD-WAN rule 1, it has the lowest link cost, and it is within SLA. Likewise, the Path Manager on Spoke 1 selects Spoke 2 H1_T22 since it has the lowest link cost compared to Spoke 2 H1_T11 (which has a cost of 100), it is within SLA, and has the same transport group as Spoke 1 H1_T11. Therefore, the Path Manager of Spoke 1 calculates the best shortcut path as Spoke 1 H1_T11 to Spoke 2 H1_T22.

The Path Manager will advise IKE to establish the best shortcut and add it to SD-WAN rule 1 as follows:

```
Branch1_FGT# diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 1
  Gen(11), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-
order
  Member sub interface(4):
    2: seq_num(1), interface(H1_T11):
       1: H1_T11_0(71)
  Members(4):
    1: Seq_num(1 H1_T11_0 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(1 H1_T11 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    3: Seq_num(2 H1_T22 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    4: Seq_num(3 H1_T33 overlay), alive, sla(0x1), gid(0), cfg_order(2), local cost(0),
selected
  Src address(2):
        172.31.0.0-172.31.255.255
        10.0.3.0-10.0.3.255
  Dst address(2):
        172.31.0.0-172.31.255.255
        10.0.4.0-10.0.4.255
…
```

Since shortcut-priority is enabled, we observe that the shortcut is formed over the selected overlay path and prioritized over the parent overlay.

From the diagnostic command on Spoke 1, we observe the selected shortcut path in **bold**. (Note that the remote IP matches Spoke 2 H1_T22 in the corresponding table above.)

```
Branch1_FGT# diagnose sys sdwan advpn-session
Session head(Branch2_FGT-0-overlay:1)
(1) Service ID(1), last access(7809088), remote health check info(3)
```

```
Selected path: local(H1_T11, port1) gw: 172.31.3.1  remote IP: 172.31.3.105(172.31.81.2)
Remote information:
1: latency: 0.176267 jitter: 0.005733 pktloss: 0.000000 mos: 4.404302 sla: 0x1 cost: 0
transport_group: 1 bandwidth up: 999994 down: 999997 bidirection: 1999991
ipv4: 172.31.3.105(172.31.81.2) ipv6 2000:172:31:3::105(::)
2: latency: 0.119133 jitter: 0.004800 pktloss: 0.000000 mos: 4.404331 sla: 0x1 cost: 0
transport_group: 2 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.4.101(172.31.82.2) ipv6 1410:4b02::f088:93ee:7f00:0
(c010:4b02::788a:93ee:7f00:0)
3: latency: 0.182400 jitter: 0.008800 pktloss: 0.000000 mos: 4.404295 sla: 0x1 cost: 100
transport_group: 1 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.3.101(172.31.80.2) ipv6 2000:172:31:3::101(d88a:93ee:7f00:0:d88a:93ee:7f00:0)
```

From the diagnostic command on Spoke 2, we observe the selected shortcut in **bold**.

```
Branch2_FGT# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(3 H1_T33): state(alive), packet-loss(0.000%) latency(0.122), jitter(0.004), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.186), jitter(0.011), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(0.180), jitter(0.005), mos(4.404),
bandwidth-up(999994), bandwidth-dw(999997), bandwidth-bi(1999991) sla_map=0x1
Seq(2 H1_T22_0): state(alive), packet-loss(0.000%) latency(0.265), jitter(0.011), mos
(4.404), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
```

Scenario 2: Traffic matching SD-WAN rule 2

In this scenario, PC 1 connected to Spoke 1 initiates an ICMP ping destined for PC2 connected to Spoke 2. Therefore, this user traffic matches SD-WAN rule 2, and traffic will go through shortcut H1_T11_0 of Spoke 1 previously established in Scenario 1 above.

The local spoke generates local-out UDP packets and sends them to the hub to trigger an IKE shortcut message exchange with updated remote spoke WAN link information. The local spoke will receive this updated remote spoke WAN link information. Then the Path Manager of Spoke 1 will recalculate the best shortcut path by comparing transport group, link quality (for SLA mode), link cost, and member configuration order between Spoke 1 and Spoke 2.

For an SLA mode service, the following algorithm is followed for considering endpoints of the best shortcut path:

1. Overlays with the same transport group
2. In-SLA overlays
3. Lowest link cost overlays
4. Member configuration order as a final tiebreaker

Based on this algorithm, the Path Manager on Spoke 1 selects Spoke 1 H1_T22 because it is the first in the priority-members order for SD-WAN rule 2, it has the lowest link cost, and it is within SLA. Likewise, the Path Manager on Spoke 1 selects Spoke 2 H1_T22 since it has the lowest link cost compared to Spoke 2 H1_T11 (which has a cost of 100), it is within SLA, and has the same transport group as Spoke 1 H1_T11. Therefore, the Path Manager of Spoke 1 calculates the best shortcut path as Spoke 1 H1_T22 to Spoke 2 H1_T22.

The Path Manager will advise IKE to establish the best shortcut and add it to SD-WAN rule 2 as follows:

```
Branch1_FGT# diagnose sys sdwan service
…
Service(2): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 1
```

```
  Gen(12), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-
order
  Member sub interface(5):
    3: seq_num(2), interface(H1_T22):
       1: H1_T22_0(72)
    4: seq_num(1), interface(H1_T11):
       1: H1_T11_0(71)
  Members(5):
    1: Seq_num(2 H1_T22_0 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(1 H1_T11_0 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected, last_used=2023-12-05 14:34:07
    3: Seq_num(2 H1_T22 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    4: Seq_num(1 H1_T11 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    5: Seq_num(3 H1_T33 overlay), alive, sla(0x1), gid(0), cfg_order(2), local cost(0),
selected
  Src address(2):
        172.31.0.0-172.31.255.255
        10.0.3.0-10.0.3.255
  Dst address(2):
        172.31.0.0-172.31.255.255
        10.0.40.0-10.0.40.255
…
```

The newly selected shortcut is prioritized over the previously selected shortcut as seen in the **bolded** output above.

From the diagnostic command on Spoke 1, we observe the selected shortcut path in **bold**. (Note that the remote IP matches Spoke 2 H1_T22 in the corresponding table above.)

```
Branch1_FGT# diagnose sys sdwan advpn-session
Session head(Branch2_FGT-0-overlay:2)
(1) Service ID(1), last access(8024725), remote health check info(3)
Selected path: local(H1_T11, port1) gw: 172.31.3.1  remote IP: 172.31.3.105(172.31.81.2)
Remote information:
1: latency: 0.118267 jitter: 0.004633 pktloss: 0.000000 mos: 4.404331 sla: 0x1 cost: 0
transport_group: 2 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.4.101(172.31.82.2) ipv6 180:adfb::d88a:93ee:7f00:0
(d88a:93ee:7f00:0:d88a:93ee:7f00:0)
2: latency: 0.176067 jitter: 0.006567 pktloss: 0.000000 mos: 4.404301 sla: 0x1 cost: 0
transport_group: 1 bandwidth up: 999994 down: 999997 bidirection: 1999991
ipv4: 172.31.3.105(172.31.81.2) ipv6 2000:172:31:3::105(::)
3: latency: 0.170333 jitter: 0.008133 pktloss: 0.000000 mos: 4.404302 sla: 0x1 cost: 100
transport_group: 1 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.3.101(172.31.80.2) ipv6 2000:172:31:3::101(c010:4b02::788a:93ee:7f00:0)
(1) Service ID(2), last access(8024725), remote health check info(3)
Selected path: local(H1_T22, port2) gw: 172.31.3.5  remote IP: 172.31.3.105(172.31.81.2)
Remote information:
1: latency: 0.118267 jitter: 0.004633 pktloss: 0.000000 mos: 4.404331 sla: 0x1 cost: 0
transport_group: 2 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.4.101(172.31.82.2) ipv6 180:adfb::d88a:93ee:7f00:0
(d88a:93ee:7f00:0:d88a:93ee:7f00:0)
2: latency: 0.176067 jitter: 0.006567 pktloss: 0.000000 mos: 4.404301 sla: 0x1 cost: 0
transport_group: 1 bandwidth up: 999994 down: 999997 bidirection: 1999991
ipv4: 172.31.3.105(172.31.81.2) ipv6 2000:172:31:3::105(::)
3: latency: 0.170333 jitter: 0.008133 pktloss: 0.000000 mos: 4.404302 sla: 0x1 cost: 100
```

```
transport_group: 1 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.3.101(172.31.80.2) ipv6 2000:172:31:3::101(c010:4b02::788a:93ee:7f00:0)
…
```

From the diagnostic command on Spoke 2, we observe the selected shortcut in **bold**:

```
Branch2_FGT# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(3 H1_T33): state(alive), packet-loss(0.000%) latency(0.118), jitter(0.005), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.171), jitter(0.005), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(0.175), jitter(0.006), mos(4.404),
bandwidth-up(999994), bandwidth-dw(999998), bandwidth-bi(1999992) sla_map=0x1
Seq(2 H1_T22_0): state(alive), packet-loss(0.000%) latency(0.240), jitter(0.009), mos
(4.404), bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
Seq(2 H1_T22_1): state(alive), packet-loss(0.000%) latency(0.259), jitter(0.019), mos
(4.404), bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
```

Scenario 3: Traffic matching SD-WAN rule 3

In this scenario, PC 1 connected to Spoke 1 initiates an ICMP ping destined for PC 3 connected to Spoke 2. Therefore, this user traffic matches SD-WAN rule 3, and traffic will go through shortcut H1_T11_0 of Spoke 1 previously established in Scenario 1 above.

The local spoke generates local-out UDP packets and sends them to the hub to trigger an IKE shortcut message exchange with updated remote spoke WAN link information. The local spoke will receive this updated remote spoke WAN link information. Then the Path Manager of Spoke 1 will recalculate the best shortcut path by comparing transport group, best quality (based on link cost factor), and member configuration order between Spoke 1 and Spoke 2.

For a best quality mode service, the following algorithm is followed for considering endpoints of the best shortcut path:

1. Overlays with the same transport group
2. Best quality overlays (link cost factor of packet loss, in this scenario)
3. Member configuration order as a final tiebreaker

Based on this algorithm, the Path Manager on Spoke 1 selects Spoke 1 H1_T33 because it is the first in the priority-members order for SD-WAN rule 3, and it has the best quality link. Likewise, the Path Manager on Spoke 1 selects Spoke 2 H1_T33 since it has the same transport group as Spoke 1 H1_T33. Therefore, the Path Manager of Spoke 1 calculates the best shortcut path as Spoke 1 H1_T33 to Spoke 2 H1_T33.

The Path Manager will advise IKE to establish the best shortcut and add it to SD-WAN rule 3 as follows:

```
Branch1_FGT# diagnose sys sdwan service
…
Service(3): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 3
  Gen(13), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(priority), link-
cost-factor(packet-loss), link-cost-threshold(10), heath-check(HUB)
  Member sub interface(6):
    4: seq_num(3), interface(H1_T33):
       1: H1_T33_0(73)
    5: seq_num(1), interface(H1_T11):
       1: H1_T11_0(71)
    6: seq_num(2), interface(H1_T22):
       1: H1_T22_0(72)
```

```
  Members(6):
    1: Seq_num(3 H1_T33_0 overlay), alive, packet loss: 0.000%, selected
    2: Seq_num(1 H1_T11_0 overlay), alive, packet loss: 0.000%, selected, last_used=2023-12-
05 14:38:02
    3: Seq_num(2 H1_T22_0 overlay), alive, packet loss: 0.000%, selected
    4: Seq_num(3 H1_T33 overlay), alive, packet loss: 0.000%, selected
    5: Seq_num(1 H1_T11 overlay), alive, packet loss: 0.000%, selected
    6: Seq_num(2 H1_T22 overlay), alive, packet loss: 0.000%, selected
  Src address(2):
        172.31.0.0-172.31.255.255
        10.0.3.0-10.0.3.255
  Dst address(2):
        172.31.0.0-172.31.255.255
        10.0.41.0-10.0.41.255
```

From the diagnostic command on Spoke 1, we observe the selected shortcut path in **bold**. (Note that the remote IP matches Spoke 2 H1_T33 in the corresponding table above.)

```
Branch1_FGT# diagnose sys sdwan advpn-session
Session head(Branch2_FGT-0-overlay:3)
(1) Service ID(3), last access(8047297), remote health check info(3)
Selected path: local(H1_T33, port3) gw: 172.31.4.1  remote IP: 172.31.4.101(172.31.82.2)
Remote information:
1: latency: 0.116600 jitter: 0.004600 pktloss: 0.000000 mos: 4.404332 sla: 0x1 cost: 0
transport_group: 2 bandwidth up: 999999 down: 999998 bidirection: 1999997
ipv4: 172.31.4.101(172.31.82.2) ipv6 180:adfb::d88a:93ee:7f00:0
(d88a:93ee:7f00:0:d88a:93ee:7f00:0)
2: latency: 0.174767 jitter: 0.005533 pktloss: 0.000000 mos: 4.404303 sla: 0x1 cost: 0
transport_group: 1 bandwidth up: 999994 down: 999998 bidirection: 1999992
ipv4: 172.31.3.105(172.31.81.2) ipv6 2000:172:31:3::105(c010:4b02::788a:93ee:7f00:0)
3: latency: 0.172900 jitter: 0.005167 pktloss: 0.000000 mos: 4.404304 sla: 0x1 cost: 100
transport_group: 1 bandwidth up: 999999 down: 999998 bidirection: 1999997
ipv4: 172.31.3.101(172.31.80.2) ipv6 2000:172:31:3::101(::)
```

From the diagnostic command on Spoke 2, we observe the selected shortcut in **bold**:

```
Branch2_FGT# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(3 H1_T33): state(alive), packet-loss(0.000%) latency(0.116), jitter(0.005), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(3 H1_T33_0): state(alive), packet-loss(0.000%) latency(0.113), jitter(0.005), mos
(4.404), bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.171), jitter(0.004), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(0.174), jitter(0.008), mos(4.404),
bandwidth-up(999994), bandwidth-dw(999998), bandwidth-bi(1999992) sla_map=0x1
Seq(2 H1_T22_0): state(alive), packet-loss(0.000%) latency(0.239), jitter(0.007), mos
(4.404), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(2 H1_T22_1): state(alive), packet-loss(0.000%) latency(0.260), jitter(0.014), mos
(4.404), bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
```

Scenario 4: Spoke 2 H1_T22 overlay link out-of-SLA

In this scenario, we place remote Spoke 2 H1_T22 out-of-SLA and observe that this link quality change is sensed by the local spoke through regular WAN link information updates on shortcuts. Then the local Spoke 1 will generate local-out UDP packets and send them to the hub to trigger an IKE shortcut message exchange. Once Spoke 1 receives a shortcut reply, it will start to calculate new best shortcut paths for SD-WAN rules 1 and 2 because these are the only rules that have new best shortcut paths when Spoke 2 H1_T22 is out-of-SLA.

For an SLA mode service, the following algorithm is followed for considering endpoints of the best shortcut path:

1. Overlays with the same transport group
2. In-SLA overlays
3. Lowest link cost overlays
4. Member configuration order as a final tiebreaker

Based on this algorithm, the Path Manager on Spoke 1 still selects these Spoke 1 interfaces:

- SD-WAN Rule 1: H1_T11
- SD-WAN Rule 2: H1_T22

These are the first in the priority-members order for SD-WAN rules 1 and 2, respectively.

Based on the updated WAN link information, the Path Manager on Spoke 1 selects these Spoke 2 interfaces because they are the only remaining in-SLA VPN overlays over Internet links (transport group 1):

- SD-WAN Rule 1: H1_T11
- SD-WAN Rule 2: H1_T11

Therefore, the Path Manager of Spoke 1 calculates the best shortcut paths as follows:

- SD-WAN Rule 1: Spoke 1 H1_T11 to Spoke 2 H1_T11
- SD-WAN Rule 2: Spoke 1 H1_T22 to Spoke 2 H1_T11

The Path Manager will advise IKE to establish the best shortcuts and add them to SD-WAN rules 1 and 2 as follows:

- For SD-WAN Rule 1, H1_T11_1 is the new best shortcut.
- For SD-WAN Rule 2, H1_T22_1 is the new best shortcut.

```
# diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 1
  Gen(17), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-
order
  Member sub interface(8):
    6: seq_num(1), interface(H1_T11):
       1: H1_T11_0(74)
       2: H1_T11_1(75)
    7: seq_num(2), interface(H1_T22):
       1: H1_T22_0(72)
       2: H1_T22_1(76)
    8: seq_num(3), interface(H1_T33):
       1: H1_T33_0(73)
  Members(8):
    1: Seq_num(1 H1_T11_0 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(1 H1_T11_1 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    3: Seq_num(2 H1_T22_0 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    4: Seq_num(2 H1_T22_1 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    5: Seq_num(3 H1_T33_0 overlay), alive, sla(0x1), gid(0), cfg_order(2), local cost(0),
```

```
selected
    6: Seq_num(1 H1_T11 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    7: Seq_num(2 H1_T22 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    8: Seq_num(3 H1_T33 overlay), alive, sla(0x1), gid(0), cfg_order(2), local cost(0),
selected
  Src address(2):
        172.31.0.0-172.31.255.255
        10.0.3.0-10.0.3.255
  Dst address(2):
        172.31.0.0-172.31.255.255
        10.0.4.0-10.0.4.255

Service(2): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 1
  Gen(17), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-
order
  Member sub interface(8):
    6: seq_num(2), interface(H1_T22):
       1: H1_T22_0(72)
       2: H1_T22_1(76)
    7: seq_num(1), interface(H1_T11):
       1: H1_T11_0(74)
       2: H1_T11_1(75)
    8: seq_num(3), interface(H1_T33):
       1: H1_T33_0(73)
  Members(8):
    1: Seq_num(2 H1_T22_0 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(2 H1_T22_1 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    3: Seq_num(1 H1_T11_1 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    4: Seq_num(1 H1_T11_0 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    5: Seq_num(3 H1_T33_0 overlay), alive, sla(0x1), gid(0), cfg_order(2), local cost(0),
selected
    6: Seq_num(2 H1_T22 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    7: Seq_num(1 H1_T11 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    8: Seq_num(3 H1_T33 overlay), alive, sla(0x1), gid(0), cfg_order(2), local cost(0),
selected
  Src address(2):
        172.31.0.0-172.31.255.255
        10.0.3.0-10.0.3.255
  Dst address(2):
        172.31.0.0-172.31.255.255
        10.0.40.0-10.0.40.255
…
```

From the diagnostic command on Spoke 1, we observe the newly selected shortcut paths in **bold**. (Note that the remote IP 172.31.80.2 matches Spoke 2 H1_T11, which is the VPN overlay over the Internet link with cost 100 in the corresponding table above.)

```
# diagnose sys sdwan advpn-session
Session head(Branch2_FGT-0-overlay:3)
(1) Service ID(1), last access(8293060), remote health check info(3)
Selected path: local(H1_T11, port1) gw: 172.31.3.1  remote IP: 172.31.3.101(172.31.80.2)
Remote information:
1: latency: 0.119500 jitter: 0.006067 pktloss: 0.000000 mos: 4.404329 sla: 0x1 cost: 0
transport_group: 2 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.4.101(172.31.82.2) ipv6 180:adfb::d88a:93ee:7f00:0
(d88a:93ee:7f00:0:d88a:93ee:7f00:0)
2: latency: 250.170761 jitter: 0.011500 pktloss: 0.000000 mos: 3.992655 sla: 0x0 cost: 0
transport_group: 1 bandwidth up: 999994 down: 999997 bidirection: 1999991
ipv4: 172.31.3.105(172.31.81.2) ipv6 2000:172:31:3::105(c010:4b02::788a:93ee:7f00:0)
3: latency: 0.182200 jitter: 0.012000 pktloss: 0.000000 mos: 4.404292 sla: 0x1 cost: 100
transport_group: 1 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.3.101(172.31.80.2) ipv6 2000:172:31:3::101(::)
(1) Service ID(2), last access(8293060), remote health check info(3)
Selected path: local(H1_T22, port2) gw: 172.31.3.5  remote IP: 172.31.3.101(172.31.80.2)
Remote information:
1: latency: 0.119500 jitter: 0.006067 pktloss: 0.000000 mos: 4.404329 sla: 0x1 cost: 0
transport_group: 2 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.4.101(172.31.82.2) ipv6 180:adfb::d88a:93ee:7f00:0
(d88a:93ee:7f00:0:d88a:93ee:7f00:0)
2: latency: 250.170761 jitter: 0.011500 pktloss: 0.000000 mos: 3.992655 sla: 0x0 cost: 0
transport_group: 1 bandwidth up: 999994 down: 999997 bidirection: 1999991
ipv4: 172.31.3.105(172.31.81.2) ipv6 2000:172:31:3::105(c010:4b02::788a:93ee:7f00:0)
3: latency: 0.182200 jitter: 0.012000 pktloss: 0.000000 mos: 4.404292 sla: 0x1 cost: 100
transport_group: 1 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.3.101(172.31.80.2) ipv6 2000:172:31:3::101(::)
```

From the diagnostic command on Spoke 2, we observe the selected shortcuts in **bold**:

```
Branch2_FGT# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(3 H1_T33): state(alive), packet-loss(0.000%) latency(0.120), jitter(0.007), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
Seq(3 H1_T33_0): state(alive), packet-loss(0.000%) latency(0.128), jitter(0.003), mos
(4.404), bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.180), jitter(0.008), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
Seq(1 H1_T11_0): state(alive), packet-loss(0.000%) latency(0.259), jitter(0.023), mos
(4.404), bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
Seq(1 H1_T11_1): state(alive), packet-loss(0.000%) latency(0.257), jitter(0.014), mos
(4.404), bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(250.169), jitter(0.009), mos
(3.993), bandwidth-up(999994), bandwidth-dw(999997), bandwidth-bi(1999991) sla_map=0x0
Seq(2 H1_T22_1): state(alive), packet-loss(0.000%) latency(0.245), jitter(0.013), mos
(4.404), bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
Seq(2 H1_T22_0): state(alive), packet-loss(0.000%) latency(0.223), jitter(0.005), mos
(4.404), bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
```

Routing

This section includes information about routing related new features:

Add option to keep sessions in established ADVPN shortcuts while they remain in SLA

This information is also available in the FortiOS 7.4 Administration Guide:

- Keeping sessions in established ADVPN shortcuts while they remain in SLA

In an SD-WAN hub and spoke configuration where ADVPN is used, when a primary shortcut goes out of SLA, traffic switches to the backup shortcut. During idle timeout, sessions will prefer using the primary parent tunnel and try to establish a new primary shortcut. However, because it is out of SLA, traffic switches back to the backup shortcut, which causes unnecessary traffic interruption.

The `shortcut-stickiness` option keeps existing sessions on the established ADVPN shortcuts while they remain in SLA instead of switching to a new link every idle timeout. New sessions will be routed through the primary shortcut if it is in SLA.

```
config system sdwan
    config service
        edit <id>
            set shortcut-stickiness {enable | disable}
        next
    end
end
```

The `shortcut-stickiness` option can be applied in the following use cases.

Use case 1:

1. The sessions will switch over to the backup shortcut due to the primary shortcut being out of SLA.
2. After an idle timeout, the primary shortcut is torn down, and the routes will be reinstalled on the primary parent tunnel.
3. When `shortcut-stickiness` is enabled, even though the primary parent tunnel is preferred, established ADVPN sessions will remain on the backup shortcut (stickiness) instead of switching to the primary parent tunnel.
4. New sessions will be routed to the primary parent tunnel and trigger the primary shortcut, then traffic switches to the primary shortcut if it is in SLA.

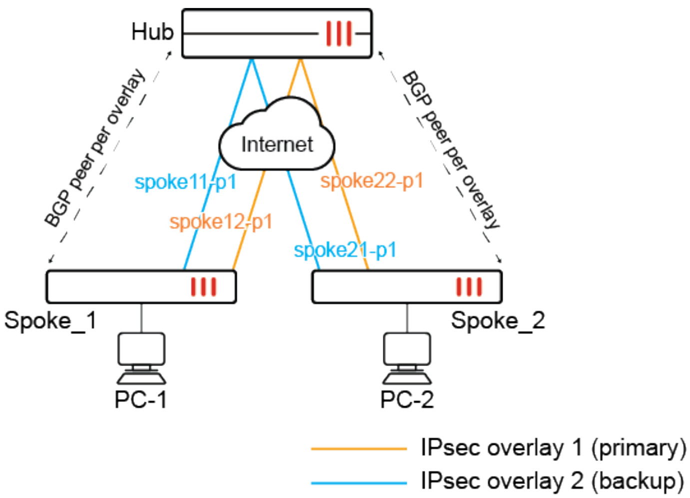
Use case 2:

1. The sessions will switch over to the backup shortcut due to the primary shortcut being out of SLA.
2. After some time, the primary shortcut becomes in SLA.
3. When `shortcut-stickiness` is enabled, even though primary shortcut is preferred, established ADVPN sessions will remain on the backup shortcut (stickiness) instead of switching to the primary shortcut.
4. New sessions will be routed through the primary shortcut.

In FortiOS 7.4.1 and later, the `shortcut-stickiness` setting is changed to `sla-stickiness` and requires `set mode sla` to be configured before enabling it. See Keeping sessions in established ADVPN shortcuts while they remain in SLA for an example configuration.

Example configuration

The following example demonstrates using the `shortcut-stickiness` option in use case 1.



After an idle timeout occurs, existing sessions remain on the spoke12-p1_0 backup shortcut tunnel. New sessions will try to create a shortcut over spoke11-p1, but will fall back to spoke12-p1_0 when it detects spoke11-p1 is out of SLA.

To configure shortcut stickiness for ADVPN shortcuts:

1. Configure SD-WAN on the Spoke_1 FortiGate:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "spoke11-p1"
        next
        edit 2
            set interface "spoke12-p1"
        next
    end
    config health-check
        edit "1"
            set server "9.0.0.1"
            set members 1 2
            config sla
                edit 1
```

```
                next
            end
        next
    end
    config service
        edit 1
            set name "1"
            set shortcut-stickiness enable
            set mode sla
            set dst "all"
            set src "10.1.100.0"
            config sla
                edit "1"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
```

2. Verify the SD-WAN configuration.
   a. Verify the health check status:

```
# diagnose sys sdwan health-check
Health Check(1):
Seq(1 spoke11-p1): state(alive), packet-loss(0.000%) latency(0.368), jitter(0.051),
mos(4.404), bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_
map=0x1
Seq(2 spoke12-p1): state(alive), packet-loss(0.000%) latency(0.211), jitter(0.019),
mos(4.404), bandwidth-up(999999), bandwidth-dw(999979), bandwidth-bi(1999978) sla_
map=0x1
```

   b. Verify the service status:

```
# diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x2200 use-shortcut-sla shortcut-stickiness
 Tie break: cfg
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(1 spoke11-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(2 spoke12-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
  Src address(1):
        10.1.100.0-10.1.100.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

   The SD-WAN service rule prefers the primary parent tunnel (spoke11-p1) over the backup parent tunnel (spoke12-p1) before shortcuts are established.

3. Send traffic from PC-1 to PC-2 to trigger the primary shortcut. Verify the diagnostics.

a. Run a sniffer trace:

```
# diagnose sniffer packet any 'host 192.168.5.44' 4
interfaces=[any]
filters=[host 192.168.5.44]
14.878761 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
14.878905 spoke11-p1 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
14.879842 spoke11-p1 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
14.880082 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
15.879761 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
15.879882 spoke11-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
15.880433 spoke11-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
15.880496 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
```

The SD-WAN service rule sends traffic to the parent tunnel (spoke11-p1) initially, and then switches to the primary shortcut tunnel (spoke11-p1_0) once it is established.

b. Verify the service status:

```
# diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x2200 use-shortcut-sla shortcut-stickiness
 Tie break: cfg
  Gen(2), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Member sub interface(3):
    2: seq_num(1), interface(spoke11-p1):
       1: spoke11-p1_0(57)
  Members(3):
    1: Seq_num(1 spoke11-p1_0), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), 
selected
    2: Seq_num(1 spoke11-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), 
selected
    3: Seq_num(2 spoke12-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), 
selected
  Src address(1):
        10.1.100.0-10.1.100.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

The SD-WAN service rule prefers the primary shortcut tunnel (spoke11-p1_0) over other tunnels.

4. Make the primary shortcut be out of SLA. The traffic will switch to the backup parent tunnel and trigger the backup shortcut. Verify the diagnostics.

a. Run a sniffer trace:

```
# diagnose sniffer packet any 'host 192.168.5.44' 4
interfaces=[any]
filters=[host 192.168.5.44]
20.588046 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
20.588157 spoke12-p1 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
20.588791 spoke12-p1 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
20.588876 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
21.589079 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
21.589190 spoke12-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
21.589661 spoke12-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
21.589733 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
```

When the primary shortcut tunnel goes out of SLA (spoke11-p1_0, alive, sla(0x0)), traffic reroutes to the backup parent tunnel (spoke12-p1) and then to the backup shortcut tunnel (spoke12-p1_0) once established.

b. Verify the service status:

```
# diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x2200 use-shortcut-sla shortcut-stickiness
 Tie break: cfg
  Gen(23), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Member sub interface(4):
    1: seq_num(1), interface(spoke11-p1):
       1: spoke11-p1_0(62)
    3: seq_num(2), interface(spoke12-p1):
       1: spoke12-p1_0(63)
  Members(4):
    1: Seq_num(1 spoke11-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(2 spoke12-p1_0), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    3: Seq_num(2 spoke12-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    4: Seq_num(1 spoke11-p1_0), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
selected
  Src address(1):
        10.1.100.0-10.1.100.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

The backup shortcut tunnel (spoke12-p1_0) is now preferred.

5. After an idle timeout, the primary shortcut is torn down. The primary parent tunnel is now preferred, but traffic is still kept on the backup shortcut due to `shortcut-stickiness` being enabled. Verify the diagnostics.

a. Verify the service status:

```
# diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x2200 use-shortcut-sla shortcut-stickiness
 Tie break: cfg
  Gen(24), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Member sub interface(3):
    3: seq_num(2), interface(spoke12-p1):
       1: spoke12-p1_0(63)
  Members(3):
    1: Seq_num(1 spoke11-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(2 spoke12-p1_0), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    3: Seq_num(2 spoke12-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
  Src address(1):
        10.1.100.0-10.1.100.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

b. Run a sniffer trace:

```
# diagnose sniffer packet any 'host 192.168.5.44' 4
interfaces=[any]
filters=[host 192.168.5.44]
1.065143 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
1.065218 spoke12-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
1.065471 spoke12-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
1.065508 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
2.066155 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
2.066198 spoke12-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
2.066442 spoke12-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
2.066480 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
3.067201 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
3.067255 spoke12-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
3.067507 spoke12-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
3.067544 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
```

6. Send new traffic from PC1 to PC2. The traffic is routed to the primary parent tunnel and triggers the primary shortcut, then traffic will switch to the primary shortcut if it is in SLA. Verify the connection.
   a. Run a sniffer trace:

```
# diagnose sniffer packet any 'host 192.168.5.4' 4
interfaces=[any]
filters=[host 192.168.5.4]
17.120310 port2 in 10.1.100.22 -> 192.168.5.4: icmp: echo request
17.120475 spoke11-p1 out 10.1.100.22 -> 192.168.5.4: icmp: echo request
17.121096 spoke11-p1 in 192.168.5.4 -> 10.1.100.22: icmp: echo reply
17.121151 port2 out 192.168.5.4 -> 10.1.100.22: icmp: echo reply
18.121331 port2 in 10.1.100.22 -> 192.168.5.4: icmp: echo request
18.121480 spoke11-p1_0 out 10.1.100.22 -> 192.168.5.4: icmp: echo request
18.121954 spoke11-p1_0 in 192.168.5.4 -> 10.1.100.22: icmp: echo reply
18.122007 port2 out 192.168.5.4 -> 10.1.100.22: icmp: echo reply
...
```

   At first, traffic tries to go to the primary parent tunnel so that it can trigger the primary shortcut to establish. The primary shortcut (spoke11-p1_0) is in SLA and new traffic flows through it.

```
...
14.194066 port2 in 10.1.100.22 -> 192.168.5.4: icmp: echo request
14.194247 spoke12-p1_0 out 10.1.100.22 -> 192.168.5.4: icmp: echo request
14.194499 spoke12-p1_0 in 192.168.5.4 -> 10.1.100.22: icmp: echo reply
14.194565 port2 out 192.168.5.4 -> 10.1.100.22: icmp: echo reply
15.195093 port2 in 10.1.100.22 -> 192.168.5.4: icmp: echo request
15.195174 spoke12-p1_0 out 10.1.100.22 -> 192.168.5.4: icmp: echo request
15.195326 spoke12-p1_0 in 192.168.5.4 -> 10.1.100.22: icmp: echo reply
15.195361 port2 out 192.168.5.4 -> 10.1.100.22: icmp: echo reply
```

   After the primary shortcut goes out of SLA, the traffic switches to the backup shortcut (spoke12-p1_0).
   b. Verify the service status:

```
# diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x2200 use-shortcut-sla shortcut-stickiness
 Tie break: cfg
  Gen(36), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Member sub interface(4):
    1: seq_num(1), interface(spoke11-p1):
       1: spoke11-p1_0(67)
```

```
        3: seq_num(2), interface(spoke12-p1):
           1: spoke12-p1_0(66)
      Members(4):
        1: Seq_num(1 spoke11-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
    selected
        2: Seq_num(2 spoke12-p1_0), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
    selected
        3: Seq_num(2 spoke12-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
    selected
        4: Seq_num(1 spoke11-p1_0), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
    selected
      Src address(1):
            10.1.100.0-10.1.100.255

      Dst address(1):
            0.0.0.0-255.255.255.255
```

New traffic switches back to the backup shortcut while the primary shortcut is still out of SLA.

Allow better control over the source IP used by each egress interface for local out traffic

This information is also available in the FortiOS 7.4 Administration Guide:

- Defining a preferred source IP for local-out egress interfaces
- Defining a preferred source IP for local-out egress interfaces on BGP routes
- Defining a preferred source IP for local-out egress interfaces on SD-WAN members

Better control over the source IP used by each egress interface is feasible by allowing a preferred source IP to be defined in each of these scenarios.

- Configuring a static route:

```
config router static
    edit <id>
        set preferred-source <ip_address>
    next
end
```

- Configuring a route map so that a BGP route can support a preferred source:

```
config router route-map
    edit <name>
        config rule
            edit <id>
                set set-ip-prefsrc <ip_address>
            next
        end
    next
end
```

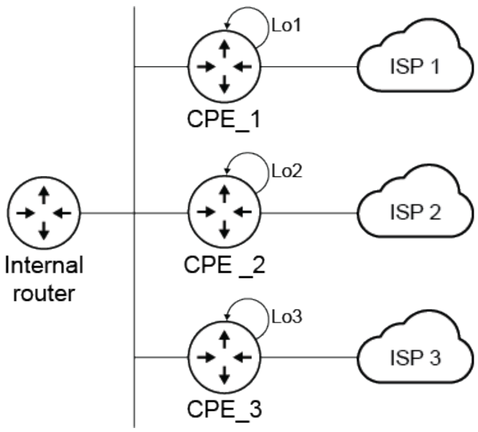
- Configuring an SD-WAN member:

```
config system sdwan
    config members
```

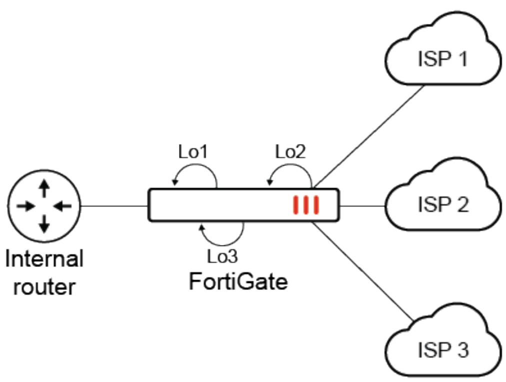
```
            edit <id>
                set preferred-source <ip_address>
            next
        end
    end
```

Example configurations

In scenarios where multiple CPE (customer premise equipment) routers are used for each transport, it is easy to define a public IP per router as a loopback IP. Then, locally sourced traffic and BGP routes can use the public loopback IP as source.



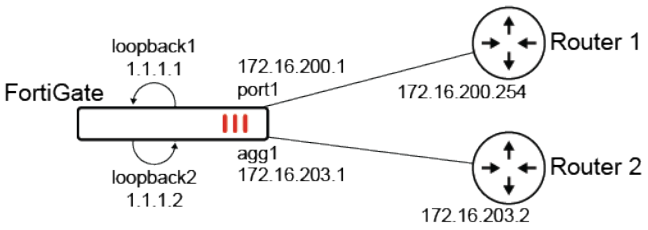
When a FortiGate is used to replace multiple CPE routers, it must be able to source traffic with the public IP assigned by their respective ISP that is assigned to the loopback interfaces.



This feature allows the preferred source IP to be configured in the following scenarios so that local out traffic is sourced from these IPs.

Example 1

In this example, a source IP is defined per static route. Local traffic that uses the static route will use the source IP instead of the interface IP associated with the route.



To configure preferred source IPs for static routes:

1. Configure the static routes:

```
config router static
    edit 22
        set dst 172.17.254.0 255.255.255.0
        set gateway 172.16.200.254
        set preferred-source 1.1.1.1
        set distance 2
        set device "port1"
    next
    edit 23
        set dst 172.17.254.0 255.255.255.0
        set gateway 172.16.203.2
        set preferred-source 1.1.1.2
        set distance 2
        set device "agg1"
    next
end
```

2. Configure the primary DNS server IP address:

```
config system dns
    set primary 172.17.254.148
end
```

To verify the configuration:

1. Verify the kernel routing table:

```
# get router info kernel
...
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->172.17.254.0/24
pref=0.0.0.0
        gwy=172.16.200.254 flag=14 hops=0 oif=9(port1) pref=1.1.1.1
        gwy=172.16.203.2 flag=14 hops=0 oif=33(agg1) pref=1.1.1.2
```

2. Verify the routing table for 172.17.254.148:

```
# get router info routing-table details 172.17.254.148
Routing table for VRF=0
Routing entry for 172.17.254.0/24
  Known via "static", distance 2, metric 0, best
  * vrf 0 172.16.200.254, via port1, prefsrc 1.1.1.1
  * vrf 0 172.16.203.2, via agg1, prefsrc 1.1.1.2
```

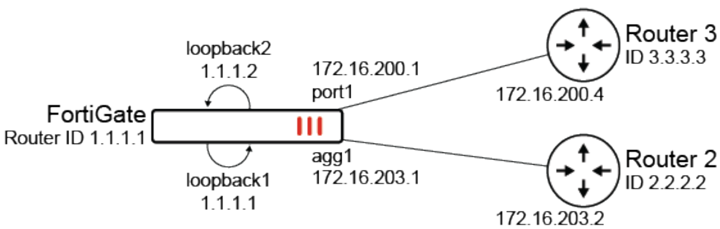
3. Run a sniffer trace after some traffic passes:

```
# diagnose sniffer packet any "host 172.17.254.148" 4
interfaces=[any]
filters=[host 172.17.254.148]
1.319811 port1 out 1.1.1.1.1371 -> 172.17.254.148.53: udp 43
1.320095 port1 in 172.17.254.148.53 -> 1.1.1.1.1371: udp 310
1.921718 port1 out 1.1.1.1.1371 -> 172.17.254.148.53: udp 27
2.031520 port1 in 172.17.254.148.53 -> 1.1.1.1.1371: udp 213
```

When DNS traffic leaves the FortiGate and is routed through port1, the source address 1.1.1.1 is used.

Example 2:

In this example, a route map is configured to set the preferred source IP so that the BGP route can support the preferred source.



To configure preferred source IPs for BGP routing:

1. Configure the route maps:

```
config router route-map
    edit "map1"
        config rule
            edit 1
                set set-ip-prefsrc 1.1.1.1
            next
        end
    next
    edit "map2"
        config rule
            edit 1
                set set-ip-prefsrc 1.1.1.2
            next
        end
    next
end
```

2. Configure the BGP settings:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    set ibgp-multipath enable
    set cluster-id 1.1.1.1
    set graceful-restart enable
    config aggregate-address
        edit 1
            set prefix 172.28.0.0 255.255.0.0
            set as-set enable
```

```
                set summary-only enable
            next
        end
        config neighbor
            edit "3.3.3.3"
                set capability-graceful-restart enable
                set soft-reconfiguration enable
                set prefix-list-out "local-out"
                set remote-as 65412
                set route-map-in "map2"
                set route-map-out "as-prepend"
                set keep-alive-timer 30
                set holdtime-timer 90
                set update-source "loopback1"
                set route-reflector-client enable
            next
            edit "2.2.2.2"
                set advertisement-interval 5
                set activate6 disable
                set capability-graceful-restart enable
                set soft-reconfiguration enable
                set distribute-list-out "local-out-FGTB-deny"
                set remote-as 65412
                set route-map-in "map1"
                set route-map-out "as-rewrite"
                set keep-alive-timer 30
                set holdtime-timer 90
                set update-source "loopback1"
            next
        end
    end
```

To verify the configuration:

1. Verify the BGP routing table for 172.25.1.0/24:

```
# get router info bgp network 172.25.1.0/24
VRF 0 BGP routing table entry for 172.25.1.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local
    2.2.2.2 (metric 10050) from 2.2.2.2 (2.2.2.2)
      Origin IGP metric 0, localpref 100, valid, internal, best, prefsrc 1.1.1.1
      Last update: Wed Jan 25 15:15:48 2023
```

2. Verify the BGP routing table for 172.28.5.0/24:

```
# get router info bgp network 172.28.5.0/24
VRF 0 BGP routing table entry for 172.28.5.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table, Advertisements suppressed
by an aggregate.)
  Not advertised to any peer
  Original VRF 0
  65050, (Received from a RR-client)
    3.3.3.3 (metric 11000) from 3.3.3.3 (3.3.3.3)
```

```
      Origin IGP metric 0, localpref 100, valid, internal, best, prefsrc 1.1.1.2
      Last update: Wed Jan 25 15:15:48 2023
```

3. Verify the kernel routing table for 172.28.5.0/24:

```
# get router info kernel | grep -B 2 172.28.5.0/24
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->172.28.1.0/24
pref=1.1.1.2 gwy=172.16.200.4 dev=9(port1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->172.28.2.0/24
pref=1.1.1.2 gwy=172.16.200.4 dev=9(port1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->172.28.5.0/24
pref=1.1.1.2 gwy=172.16.200.4 dev=9(port1)
```

4. Verify the kernel routing table for 172.25.1.0/24:

```
# get router info kernel | grep -A 2 172.25.1.0/24
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->172.25.1.0/24
pref=1.1.1.1 gwy=172.16.203.2 dev=33(agg1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->172.26.1.0/24
pref=1.1.1.1 gwy=172.16.203.2 dev=33(agg1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->172.26.2.0/24
pref=1.1.1.1 gwy=172.16.203.2 dev=33(agg1)
```

   The FortiGate learns routes from router 3.3.3.3 and prefers the source IP of 1.1.1.2. It learns routes from router 2.2.2.2 and prefers source IP of 1.1.1.1.

5. Run a sniffer trace after some traffic passes.
   a. When trying to reach a destination in the 172.25.1.0/0 subnet through router 2.2.2.2:

```
# diagnose sniffer packet any "icmp" 4
interfaces=[any]
filters=[icmp]
9.244334 agg1 out 1.1.1.1 -> 172.25.1.2: icmp: echo request
9.244337 port12 out 1.1.1.1 -> 172.25.1.2: icmp: echo request
10.244355 agg1 out 1.1.1.1 -> 172.25.1.2: icmp: echo request
10.244357 port12 out 1.1.1.1 -> 172.25.1.2: icmp: echo request
```

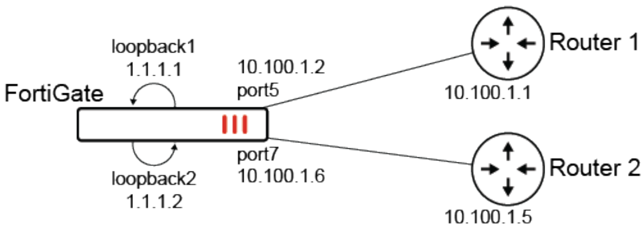
   b. When trying to reach a destination in the 172.28.5.0/24 subnet through router 3.3.3.3:

```
# diagnose sniffer packet any "icmp" 4
interfaces=[any]
filters=[icmp]
2.434035 port1 out 1.1.1.2 -> 172.28.5.2: icmp: echo request
3.434059 port1 out 1.1.1.2 -> 172.28.5.2: icmp: echo request
```

   Traffic destined for the 172.25.1.0/24 subnet uses 1.1.1.1 as source. Traffic destined for the 172.28.5.0/24 subnet uses 1.1.1.2 as source.

Example 3:

In this example, two SD-WAN members, port5 and port6, will use loopback1 and loopback2 as sources instead of their physical interface address. A static route is created for destination 200.0.0.0/24 to use the virtual-wan-link. In turn, the FortiGate will create two ECMP routes to the member gateways and source the traffic from the loopback IPs.



To configure preferred source IPs for SD-WAN members:

1. Configure the SD-WAN members and other settings:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "port5"
            set gateway 10.100.1.1
            set preferred-source 1.1.1.1
            set source 1.1.1.1
        next
        edit 2
            set interface "port7"
            set gateway 10.100.1.5
            set preferred-source 1.1.1.2
            set source 1.1.1.2
        next
    end
end
```

In the SD-WAN `config members` settings, configuring the `source` for the health check probes is still required. SD-WAN adds dedicated kernel routes (proto=17) for the health checks using the interface IP or source IP when specified. To view the kernel routes, use `diagnose ip route list`.

2. Configure the static route:

```
config router static
    edit 2000
        set dst 200.0.0.0 255.255.255.0
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
end
```

To verify the configuration:

1. Verify the kernel routing table for 200.0.0.0/24:

```
# get router info kernel | grep -A 2 200.0.0.0/24
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->200.0.0.0/24 pref=0.0.0.0
```

```
        gwy=10.100.1.1 flag=14 hops=255 oif=13(port5) pref=1.1.1.1
        gwy=10.100.1.5 flag=14 hops=254 oif=15(port7) pref=1.1.1.2
```

2. Verify the routing table for 200.0.0.0/24:

```
# get router info routing-table details 200.0.0.0/24
Routing table for VRF=0
Routing entry for 200.0.0.0/24
  Known via "static", distance 1, metric 0, best
  * vrf 0 10.100.1.1, via port5, prefsrc 1.1.1.1
  * vrf 0 10.100.1.5, via port7, prefsrc 1.1.1.2
```

3. Run a sniffer trace after some traffic passes.
   a. When traffic leaves port5:

```
# diagnose sniffer packet any "host 200.0.0.1" 4
interfaces=[any]
filters=[host 200.0.0.1]
6.592488 port5 out 1.1.1.1 -> 200.0.0.1: icmp: echo request
7.592516 port5 out 1.1.1.1 -> 200.0.0.1: icmp: echo request
8.592532 port5 out 1.1.1.1 -> 200.0.0.1: icmp: echo request
```

   b. When traffic leaves port7:

```
# diagnose sniffer packet any "host 200.0.0.1" 4
interfaces=[any]
filters=[host 200.0.0.1]
75.664173 port7 out 1.1.1.2 -> 200.0.0.1: icmp: echo request
76.664194 port7 out 1.1.1.2 -> 200.0.0.1: icmp: echo request
```

Traffic exiting each interface is sourced from the corresponding loopback IP.

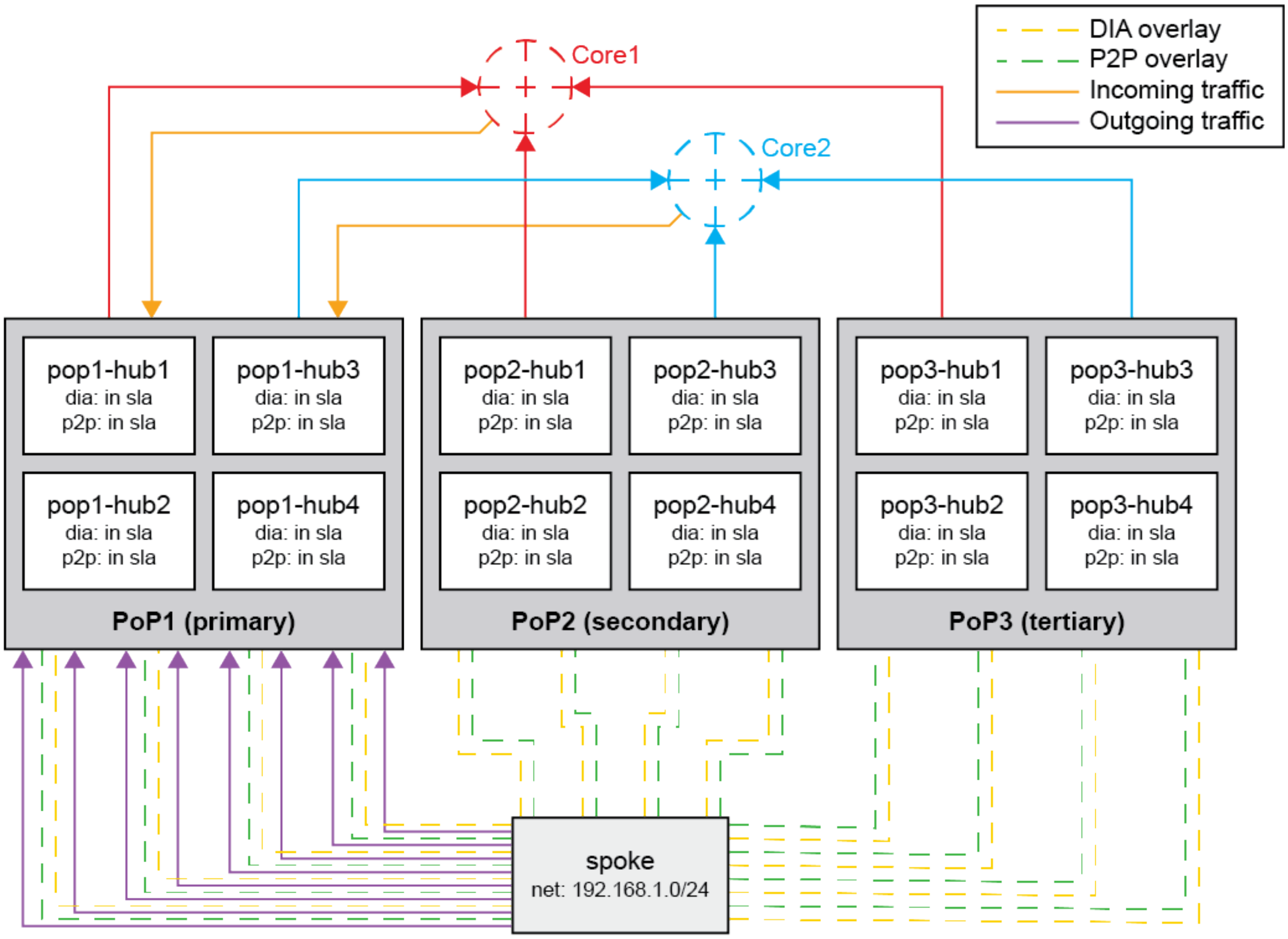
SD-WAN multi-PoP multi-hub large scale design and failover - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- SD-WAN multi-PoP multi-hub large scale design and failover

FortiOS 7.2.0 introduced a feature to define the minimum number of SD-WAN interface members that must meet SLA in order for the spoke to select a hub to process its SD-WAN traffic. This design is suitable for a single-PoP multi-hub architecture in order to achieve hub-to-hub failover. See Using multiple members per SD-WAN neighbor configuration for more information.

In FortiOS 7.4.1, the design is enhanced to support a multi-PoP multi-hub architecture in which incoming and outgoing traffic failover between PoPs is supported.



Based on the preceding diagram, incoming and outgoing traffic to the spoke is preferred over PoP1. If a single hub within PoP1 goes out of SLA, traffic will continue to flow through the PoP. If the minimum number of members to meet SLA in the PoP cannot be met, then traffic will fail over to PoP2.

The following enhancements have been made to support the multi-PoP failover scenario.

- Add `minimum-sla-meet-members` setting in the SD-WAN zone configurations and `zone-mode` setting in the SD-WAN service configurations:

```
config system sdwan
    config zone
        edit <name>
            set minimum-sla-meet-members <integer>
        next
    end
    config service
        edit <id>
            set mode sla
            set zone-mode {enable | disable}
        next
    end
end
```

When `zone-mode` is enabled on a SD-WAN service rule, the traffic is steered based on the status of the zone.

The state of the health check referenced in the SD-WAN service can be defined as follows:

- If the number of in SLA members in a zone is less than the `minimum-sla-meet-members`, then the zone's state is out of SLA; otherwise, it is in SLA.
- If a zone's state is out of SLA, then all members in the zone are out of SLA.
- If a zone's state is in SLA, then the health check's state of individual members in the zone is determined by its own state.

- Add `service-id` setting in the SD-WAN neighbor configurations:

```
config system sdwan
    config neighbor
        edit <bgp_neighbor_ip>
            set member <member_id>
            set service-id <id>
        next
    end
end
```

The SD-WAN neighbor's behavior can be determined by SD-WAN service and naturally synchronizes with SD-WAN service.

- The SD-WAN service defines priority zones, whose SLA state determines the advertised community preferable string.
- The SD-WAN service defines the `hold-down-time`, which determines how long an advertised community preferable string can be kept when it is expected to be changed.

- Add `sla-stickness` setting in the SD-WAN service configurations:

```
config system sdwan
    config service
        edit <id>
            set mode sla
            set sla-stickiness {enable | disable}
        next
    end
end
```

The switch-over of an existing session is determined as follows:

- If the outgoing interface of the session is in SLA, then the session can keep its outgoing interface.
- Otherwise, the session switches to a preferable path if one exists.

- Allow the neighbor group to be configured in the SD-WAN neighbor configurations:

```
config system sdwan
    config neighbor
        edit <bgp_neighbor_group>
            set member <member_id>
            set health-check <name>
            set sla-id <id>
        next
    end
end
```

Outgoing path control

The outgoing path from spoke to hub operates as follows:

1. Overlays to the primary and secondary PoP are assigned separately into an SD-WAN primary and secondary zone on the spoke.
2. One SD-WAN service rule is defined to include these zones as SD-WAN members.
3. When the primary zone is in SLA (`minimum-sla-meet-members` is met), the SD-WAN service rule steers traffic to the in SLA overlay members.
4. When the primary zone is out of SLA (`minimum-sla-meet-members` is not met), the SD-WAN service rule steers traffic to the in SLA overlay members in the secondary zone.
5. When the primary zone SLA is recovered:
    a. If `sla-stickness` is disabled on the SD-WAN service rule, then traffic will wait the duration of the `hold-down-time` before switching back to in SLA overlays in the primary zone.
    b. If `sla-stickness` is enabled on the SD-WAN service rule, then existing traffic will be kept on the in SLA overlays on the secondary zone, but new traffic will be steered to in SLA overlays in the primary zone.

Incoming path control

The incoming traffic from the core/external peers, to PoP, to spoke operates as follows:

1. When the primary zone is in SLA, the spoke uses the preferable route map to advertise local routes with the in SLA community to hubs in the primary and secondary PoPs.
    a. Hubs in the primary PoP translate the in SLA community into a short AS path and advertise it to external peers to attract incoming traffic.
    b. Hubs in the secondary PoP translate the in SLA community into a longer AS path and advertise it to external peers to deflect incoming traffic.
2. If the number of in SLA overlays in the primary zone is less than the `minimum-sla-meet-members`, then the spoke will use the default route map to advertise routes instead of with an out of SLA community to hubs in the primary PoP.
    a. Hubs in the primary PoP translate the out of SLA community into a longest AS path, and advertise it to external peers to deflect incoming traffic.
    b. As a result, inbound traffic is routed to hubs in the secondary PoP.
3. When the primary zone SLA is recovered:
    a. The spoke will wait the duration of the predefined `hold-down-time` in the SD-WAN service rule to use the preferable route map again to advertise routes with the in SLA community to hubs in the primary PoP.
    b. As a result, inbound traffic will be routed back to hubs in the primary PoP.

Neighbor group configuration

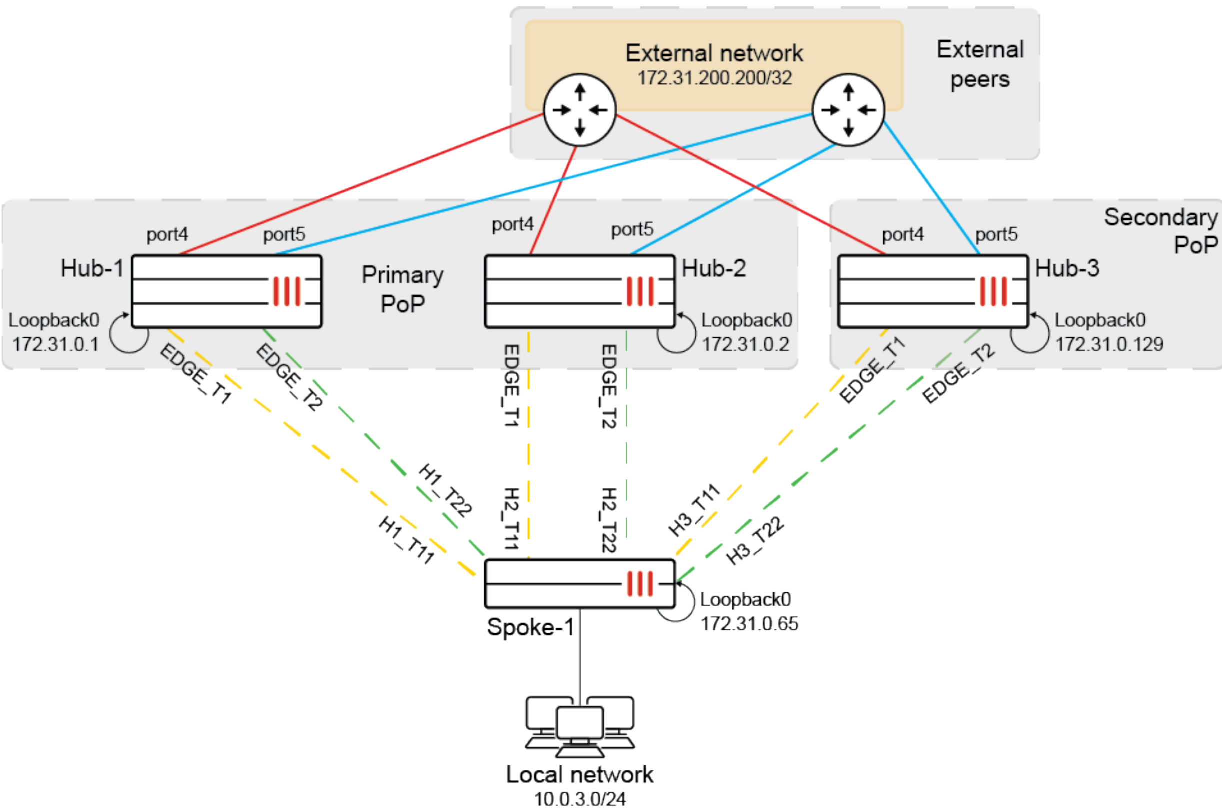
By configuring the neighbor group for spokes under the hub's SD-WAN neighbor configuration, if all paths from the hub to external peers are detected as out of SLA, then the hub will use the default route map to deny external routes to spokes that belong to this neighbor group defined on the hub. As a result, spokes will skip that specific hub and connect to external peers from other hubs.

This allows spokes to only measure overlay quality to each hub, and hubs to manage health checks to services by external peers. This significantly decreases the number of health check probes directly from the spoke to services and decreases the overall complexity. The complexity is further simplified by using multiple VRFs or segmentation where each spoke needs to send health check probes.

Example

This example configuration contains the following components:

- Two PoPs:
  - The primary PoP has two hubs (Hub-1 and Hub-2).
  - The secondary PoP has one hub (Hub-3).
- Spoke-1 has six overlays, with two overlay connections to each hub.
- Spoke-1 has three BGP neighbors, with one BGP neighbor for each hub.
  - All BGP neighbors are established on loopback IPs.
- Each hub has two paths to external peers.



Normally, outbound and inbound traffic go through hubs in the primary PoP. If the number of in SLA overlays to the primary PoP is less than the `minimum-sla-meet-members` (set to 2 in this example), bi-directional traffic needs to be switched to hubs in the secondary PoP. But when the primary PoP recovers and the `minimum-sla-meet-members` is met again, bi-directional traffic is forced back to hubs in the primary PoP after the predefined `hold-down-time` duration.

The hubs do not require SD-WAN configurations to the spokes. However, they use SD-WAN for connections to external peer routers.

Configuring the FortiGates

The following configurations highlight important routing and SD-WAN settings that must be configured on the spoke and the hubs. It is assumed that other configurations such as underlays, IPsec VPN overlays, loopbacks, static routes, and so on are already configured.

To configure Spoke-1:

1. Create the primary (PoP1) and secondary (PoP2) zones, and set the `minimum-sla-meet-members` to `2` on PoP1:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "PoP1"
            set minimum-sla-meet-members 2
        next
        edit "PoP2"
        next
    end
end
```

2. Add the overlay members to each zone. Four overlays are defined for PoP1, and two overlays are defined for PoP2:

```
config system sdwan
    config members
        edit 1
            set interface "H1_T11"
            set zone "PoP1"
        next
        edit 2
            set interface "H1_T22"
            set zone "PoP1"
        next
        edit 3
            set interface "H2_T11"
            set zone "PoP1"
        next
        edit 4
            set interface "H2_T22"
            set zone "PoP1"
        next
        edit 5
            set interface "H3_T11"
            set zone "PoP2"
        next
        edit 6
            set interface "H3_T22"
            set zone "PoP2"
        next
    end
end
```

3. Configure a performance SLA health check to a probe server behind the three hubs:

```
config system sdwan
    config health-check
        edit "Hubs"
            set server "172.31.100.100"
            set source 172.31.0.65
            set members 0
            config sla
```

```
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 200
                next
            end
        next
    end
end
```

4. Configure the service rule with the following settings: use SLA mode, enable zone mode to steer traffic based on the zone statuses, enable `sla-stickiness`, and use a 30-second hold down so that upon a recovery, existing sessions will remain on the secondary PoP while new sessions will switch back to the primary PoP once the 30-second duration ends:

```
config system sdwan
    config service
        edit 1
            set mode sla
            set zone-mode enable
            set dst "all"
            set src "CORP_LAN"
            set hold-down-time 30
            set sla-stickiness enable
            config sla
                edit "Hubs"
                    set id 1
                next
            end
            set priority-zone "PoP1" "PoP2"
        next
    end
end
```

   Since the PoP1 zone is specified before PoP2, PoP1 is regarded as the primary and preferred over the PoP2 zone.

5. Configure the in_sla and out_sla route maps that define the communities that are advertised to the hub when the zones are in and out of SLA.
   a. Configure the access list:

```
config router access-list
    edit "net10"
        config rule
            edit 1
                set prefix 10.0.3.0 255.255.255.0
            next
        end
    next
end
```

   b. Configure the route maps:

```
config router route-map
    edit "in_sla"
        config rule
            edit 1
                set match-ip-address "net10"
                set set-community "10:1"
            next
```

```
            end
        next
        edit "out_sla"
            config rule
                edit 1
                    set match-ip-address "net10"
                    set set-community "10:2"
                next
            end
        next
    end
```

6. Configure the default route map for out of SLA scenarios, preferable route map for in SLA scenarios, and the local network to be advertised:

```
config router bgp
    config neighbor
        edit "172.31.0.1"
            ...
            set route-map-out "out_sla"
            set route-map-out-preferable "in_sla"
            ...
        next
        edit "172.31.0.2"
            ...
            set route-map-out "out_sla"
            set route-map-out-preferable "in_sla"
            ...
        next
        edit "172.31.0.129"
            ...
            set route-map-out "out_sla"
            set route-map-out-preferable "in_sla"
            ...
        next
    end
    config network
        edit 1
            set prefix 10.0.3.0 255.255.255.0
        next
    end
    ...
end
```

7. Define SD-WAN neighbors for each hub. The `minimum-sla-meet-members` is configured for the Hub-1 neighbor so that bi-directional traffic goes through Hub-1 as long as the in SLA overlays to Hub-1 are no less than 1. Associate the previously defined service rule to each SD-WAN neighbor:

```
config system sdwan
    config neighbor
        edit "172.31.0.1"
            set member 1 2
            set minimum-sla-meet-members 1
            set service-id 1
        next
        edit "172.31.0.2"
            set member 3 4
```

```
            set service-id 1
        next
        edit "172.31.0.129"
            set member 5 6
            set service-id 1
        next
    end
end
```

To configure the hubs:

1. Configure the SD-WAN zone, members, and health check for the external connections to peer routers. Performance SLA health checks are sent to external servers in order to measure the health of the external connections:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "port4"
        next
        edit 2
            set interface "port5"
        next
    end
    config health-check
        edit "external_peers"
            set server "10.0.1.2"
            set members 1 2
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 200
                next
            end
        next
    end
end
```

2. Configure the route maps for in and out of SLA scenarios. When out of SLA (one of the external connections is down), external routes are denied to be advertised to the spokes that are part of the neighbor group.
   a. Configure the access list:

```
config router access-list
    edit "net_Lo"
        config rule
            edit 1
                set prefix 172.31.200.200 255.255.255.255
            next
        end
    next
end
```

b. Configure the route maps:

```
config router route-map
    edit "in_sla"
        config rule
            edit 1
                set match-ip-address "net_Lo"
            next
        end
    next
    edit "out_sla"
        config rule
            edit 1
                set action deny routes
                set match-ip-address "net_Lo"
            next
        end
    next
end
```

3. In the BGP settings, configure the external network prefix to advertise. Then configure the neighbor group and neighbor range for the spokes. Configure the preferable and default route maps to define the behavior when the external connections are in and out of SLA:

```
config router bgp
    ...
    config network
        edit 1
            set prefix 172.31.200.200 255.255.255.255
        next
    end
    config neighbor-group
        edit "EDGE"
            ...
            set route-map-out "out_sla"
            set route-map-out-preferable "in_sla"
            ...
        next
    end
    config neighbor-range
        edit 1
            set prefix 172.31.0.64 255.255.255.192
            set neighbor-group "EDGE"
        next
    end
    ...
end
```

4. Configure the SD-WAN neighbor to match the neighbor group that includes spokes as members. Specify that at least one of the external peer connections needs to be up to be considered in SLA:

```
config system sdwan
    config neighbor
        edit "EDGE"
            set member 1 2
            set minimum-sla-meet-members 1
            set health-check "external_peers"
```

```
                set sla-id 1
            next
        end
    end
```

Testing and verification

The following tests use diagnostic commands on various FortiGates to verify the connections in the SD-WAN configuration.

Test case 1: the primary PoP and Hub-1 are in SLA

To verify the configuration:

1. Verify the SD-WAN service rules status on Spoke-1. When all six overlays are in SLA on Spoke-1, the primary PoP and primary zone PoP1 are preferred. In particular, the overlay H1_T11 over PoP1 is preferred:

```
Spoke-1 (root) # diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x1c200 use-shortcut-sla use-shortcut sla-
stickiness
 Tie break: cfg
  Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-
compare-order
Hold down time(30) seconds, Hold start at 362646 second, now 362646
  Service role: standalone
  Members(6):
    1: Seq_num(1 H1_T11 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(2 H1_T22 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    3: Seq_num(3 H2_T11 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    4: Seq_num(4 H2_T22 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    5: Seq_num(5 H3_T11 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    6: Seq_num(6 H3_T22 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

2. Verify the BGP learned routes on Hub-1. The local route with in SLA community 10:1 is advertised to all hubs. Though, the AS paths on Hub-1 and Hub-2 are shorter than Hub-3:

```
PoP1-Hub1 (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:1
      Last update: Mon Jul 17 15:16:57 2023
```

3. Send traffic from a host behind Spoke-1 to 172.31.200.200.
4. Run a sniffer trace on Spoke-1. Traffic leaves and returns on the H1_T11 overlay :

```
Spoke-1 (root) # diagnose sniffer packet any 'host 172.31.200.200' 4
interfaces=[any]
filters=[host 172.31.200.200]
5.098248 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
5.098339 H1_T11 out 10.0.3.2 -> 172.31.200.200: icmp: echo request
5.098618 H1_T11 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
5.098750 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
```

Test case 2: a single SD-WAN member on Hub-1 is out of SLA

Hub-1 and PoP1 are still preferred in this scenario.

To verify the configuration:

1. Verify the health check status on Spoke-1. The H1_T11 overlay on Hub-1/PoP1 is out of SLA:

```
Spoke-1 (root) # diagnose sys sdwan health-check
Health Check(Hubs):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(220.214), jitter(0.015), mos
(4.104), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x0
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(0.196), jitter(0.014), mos
(4.404), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(3 H2_T11): state(alive), packet-loss(0.000%) latency(0.173), jitter(0.008), mos
(4.404), bandwidth-up(999998), bandwidth-dw(999997), bandwidth-bi(1999995) sla_map=0x1
…
```

2. Verify the SD-WAN neighbor status. The SD-WAN neighbor still displays Hub-1’s zone status as pass/alive:

```
Spoke-1 (root) # diagnose sys sdwan neighbor
SD-WAN neighbor status: hold-down(disable), hold-down-time(0), hold_boot_time(0)
        Selected role(standalone) last_secondary_select_time/current_time in seconds
0/436439
Neighbor(172.31.0.1): member(1 2)role(standalone)
        Health-check(:0)  sla-pass selected alive
Neighbor(172.31.0.2): member(3 4)role(standalone)
        Health-check(:0)  sla-pass selected alive
Neighbor(172.31.0.129): member(5 6)role(standalone)
        Health-check(:0)  sla-pass selected alive
```

3. Verify the SD-WAN service rules status. Spoke-1 steers traffic to the H1_T22 overlay through Hub-1:

```
Spoke-1 (root) # diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x1c200 use-shortcut-sla use-shortcut sla-
stickiness
 Tie break: cfg
  Gen(2), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-
compare-order
Hold down time(30) seconds, Hold start at 364162 second, now 364162
  Service role: standalone
  Members(6):
    1: Seq_num(2 H1_T22 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(3 H2_T11 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
```

```
    3: Seq_num(4 H2_T22 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    4: Seq_num(5 H3_T11 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    5: Seq_num(6 H3_T22 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    6: Seq_num(1 H1_T11 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

4. Verify the BGP learned routes on Hub-1. The hubs continue to receive community 10:1 from the spoke and continue to route incoming traffic through Hub-1:

```
PoP1-Hub1 (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:1
      Last update: Mon Jul 17 15:16:57 2023
```

5. Send traffic from a host behind Spoke-1 to 172.31.200.200.
6. Run a sniffer trace on Spoke-1. Traffic leaves and returns on the H1_T22 overlay:

```
Spoke-1 (root) # diagnose sniffer packet any 'host 172.31.200.200' 4
interfaces=[any]
filters=[host 172.31.200.200]
25.299006 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
25.299080 H1_T22 out 10.0.3.2 -> 172.31.200.200: icmp: echo request
25.299323 H1_T22 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
25.299349 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
```

Test case 3: both SD-WAN members on Hub-1 are out of SLA

Other in SLA overlays in zone PoP1 though Hub-2 are still preferred over PoP2 in this scenario.

To verify the configuration:

1. Verify the health check status on Spoke-1. Both H1_T11 and H1_T22 overlays on Hub-1/PoP1 are out of SLA:

```
Spoke-1 (root) # diagnose sys sdwan health-check
Health Check(Hubs):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(220.220), jitter(0.018), mos
(4.103), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x0
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(220.174), jitter(0.007), mos
(4.104), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x0
Seq(3 H2_T11): state(alive), packet-loss(0.000%) latency(0.184), jitter(0.015), mos
(4.404), bandwidth-up(999998), bandwidth-dw(999997), bandwidth-bi(1999995) sla_map=0x1
Seq(4 H2_T22): state(alive), packet-loss(0.000%) latency(0.171), jitter(0.008), mos
(4.404), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(5 H3_T11): state(alive), packet-loss(0.000%) latency(0.173), jitter(0.011), mos
```

```
(4.404), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(6 H3_T22): state(alive), packet-loss(0.000%) latency(0.179), jitter(0.011), mos
(4.404), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
```

2. Verify the SD-WAN neighbor status. The SD-WAN neighbor displays Hub-1's zone status as failed. However, SD-WAN Hub-2 is pass/alive:

```
Spoke-1 (root) # diagnose sys sdwan neighbor
SD-WAN neighbor status: hold-down(disable), hold-down-time(0), hold_boot_time(0)
        Selected role(standalone) last_secondary_select_time/current_time in seconds
0/436535
Neighbor(172.31.0.1): member(1 2)role(standalone)
        Health-check(:0)  sla-fail alive
Neighbor(172.31.0.2): member(3 4)role(standalone)
        Health-check(:0)  sla-pass selected alive
Neighbor(172.31.0.129): member(5 6)role(standalone)
        Health-check(:0)  sla-pass selected alive
```

3. Verify the SD-WAN service rules status. Spoke-1 steers traffic to the H2_T11 overlay through Hub-2:

```
Spoke-1 (root) # diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x1c200 use-shortcut-sla use-shortcut sla-
stickiness
 Tie break: cfg
  Gen(3), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-
compare-order
Hold down time(30) seconds, Hold start at 364489 second, now 364490
  Service role: standalone
  Members(6):
    1: Seq_num(3 H2_T11 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(4 H2_T22 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    3: Seq_num(5 H3_T11 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    4: Seq_num(6 H3_T22 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    5: Seq_num(1 H1_T11 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
selected
    6: Seq_num(2 H1_T22 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

4. Verify the BGP learned routes on Hub-1 and Hub-2. Hub-2 and Hub-3 continue to receive community 10:1 from Spoke-1, but Hub-1 receives the out of SLA community of 10:2.
   a. On Hub-1:

   ```
   PoP1-Hub1 (root) # get router info bgp network 10.0.3.0/24
   VRF 0 BGP routing table entry for 10.0.3.0/24
   Paths: (1 available, best #1, table Default-IP-Routing-Table)
     Not advertised to any peer
     Original VRF 0
     Local, (Received from a RR-client)
       172.31.0.65 from 172.31.0.65 (172.31.0.65)
   ```

```
        Origin IGP metric 0, localpref 100, valid, internal, best
        Community: 10:2
        Last update: Mon Jul 17 18:08:58 2023
```

b. On Hub-2:

```
PoP1-Hub2 (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:1
      Last update: Mon Jul 17 15:31:43 2023
```

5. Send traffic from a host behind Spoke-1 to 172.31.200.200.
6. Run a sniffer trace on Spoke-1. Traffic leaves and returns on the H2_T11 overlay:

```
Spoke-1 (root) # diagnose sniffer packet any 'host 172.31.200.200' 4
interfaces=[any]
filters=[host 172.31.200.200]
13.726009 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
13.726075 H2_T11 out 10.0.3.2 -> 172.31.200.200: icmp: echo request

13.726354 H2_T11 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
13.726382 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
```

Test case 4: three SD-WAN members on PoP1 are out of SLA

The number of in SLA overlays in zone PoP1 is less than the `minimum-sla-meet-members` in zone PoP1. The SD-WAN service rule for Hub-2 is forcibly marked as `sla(0x0)` or out of SLA.

To verify the configuration:

1. Verify the health check status on Spoke-1. All three H1_T11, H1_T22, and H2_T11 overlays on PoP1 are out of SLA:

```
Spoke-1 (root) # diagnose sys sdwan health-check
Health Check(Hubs):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(220.219), jitter(0.019), mos
(4.103), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x0
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(220.184), jitter(0.008), mos
(4.104), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x0
Seq(3 H2_T11): state(alive), packet-loss(0.000%) latency(220.171), jitter(0.009), mos
(4.104), bandwidth-up(999998), bandwidth-dw(999997), bandwidth-bi(1999995) sla_map=0x0
Seq(4 H2_T22): state(alive), packet-loss(0.000%) latency(0.180), jitter(0.013), mos
(4.404), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(5 H3_T11): state(alive), packet-loss(0.000%) latency(0.174), jitter(0.014), mos
(4.404), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(6 H3_T22): state(alive), packet-loss(0.000%) latency(0.179), jitter(0.015), mos
(4.404), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
```

2. Verify the SD-WAN neighbor status. The SD-WAN neighbor displays Hub-1 and Hub-2's zone status as failed:

```
Spoke-1 (root) # diagnose sys sdwan neighbor
SD-WAN neighbor status: hold-down(disable), hold-down-time(0), hold_boot_time(0)
        Selected role(standalone) last_secondary_select_time/current_time in seconds
0/436605
Neighbor(172.31.0.1): member(1 2)role(standalone)
        Health-check(:0)  sla-fail alive
Neighbor(172.31.0.2): member(3 4)role(standalone)
        Health-check(:0)  sla-fail alive
Neighbor(172.31.0.129): member(5 6)role(standalone)
        Health-check(:0)  sla-pass selected alive
```

3. Verify the SD-WAN service rules status. Since the minimum SLA members is not met for the primary zone (PoP1), the remaining overlay in PoP1 associated with the SD-WAN service rule is forcibly set to out of SLA. Spoke-1 steers traffic to the H3_T11 overlay through Hub-3:

```
Spoke-1 (root) # diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x1c200 use-shortcut-sla use-shortcut sla-
stickiness
 Tie break: cfg
  Gen(6), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-
compare-order
Hold down time(30) seconds, Hold start at 365341 second, now 365341
  Service role: standalone
  Members(6):
    1: Seq_num(5 H3_T11 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    2: Seq_num(6 H3_T22 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    3: Seq_num(1 H1_T11 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
selected
    4: Seq_num(2 H1_T22 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
selected
    5: Seq_num(3 H2_T11 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
selected
    6: Seq_num(4 H2_T22 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

4. Verify the BGP learned routes on each hub. Hub-3 continues to receive community 10:1 from Spoke-1, but Hub-1 and Hub-2 receive the out of SLA community of 10:2.
   a. On Hub-1:

```
PoP1-Hub1 (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:2
      Last update: Mon Jul 17 18:22:14 2023
```

   b. On Hub-2:

```
PoP1-Hub2 (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:2
      Last update: Mon Jul 17 18:37:53 2023
```

c. On Hub-3:

```
PoP2-Hub3 (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:1
      Last update: Mon Jul 17 14:39:04 2023
```

5. Send traffic from a host behind Spoke-1 to 172.31.200.200.
6. Run a sniffer trace on Spoke-1. Traffic leaves and returns on the H3_T11 overlay:

```
Spoke-1 (root) # diagnose sniffer packet any 'host 172.31.200.200' 4
interfaces=[any]
filters=[host 172.31.200.200]
38.501449 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
38.501519 H3_T11 out 10.0.3.2 -> 172.31.200.200: icmp: echo request
38.501818 H3_T11 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
38.501845 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
```

Test case 5: an SD-WAN member on PoP1 recovers

SD-WAN member H2_T11 recovers and brings the number of overlays in SLA back to being above the `minimum-sla-meet-members` threshold in PoP1. After the hold down time duration (30 seconds), in SLA overlays in zone PoP1 are preferred over PoP2 again. With `sla-stickiness` enabled, existing traffic is kept on H3_T11, but new traffic is steered to H2_T11.

To verify the configuration:

1. Verify the SD-WAN service rules status on Spoke-1. The hold down timer has not yet passed, so H2_T11 is not yet preferred—even though the SLA status is pass/alive:

```
Spoke-1 (root) # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x1c200 use-shortcut-sla use-shortcut sla-
stickiness
 Tie break: cfg
  Gen(16), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-
compare-order
Hold down time(30) seconds, Hold start at 431972 second, now 432000
  Service role: standalone
```

```
  Members(6):
    1: Seq_num(5 H3_T11 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    2: Seq_num(6 H3_T22 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    3: Seq_num(1 H1_T11 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
selected
    4: Seq_num(2 H1_T22 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
selected
    5: Seq_num(3 H2_T11 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    6: Seq_num(4 H2_T22 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
```

2. Verify the SD-WAN service rules status again after the hold down timer passes. H2_T11 and H2_T22 from PoP1 are now preferred:

```
Spoke-1 (root) # diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x1c200 use-shortcut-sla use-shortcut sla-
stickiness
 Tie break: cfg
  Gen(17), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-
compare-order
Hold down time(30) seconds, Hold start at 432003 second, now 432003
  Service role: standalone
  Members(6):
    1: Seq_num(3 H2_T11 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(4 H2_T22 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    3: Seq_num(5 H3_T11 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    4: Seq_num(6 H3_T22 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    5: Seq_num(1 H1_T11 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
selected
    6: Seq_num(2 H1_T22 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
selected
```

3. Verify the BGP learned routes on Hub-2, which now receives community 10:1 from Spoke-1:

```
PoP1-Hub2 (root) #  get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:1
      Last update: Tue Jul 18 14:41:32 2023
```

4. Send traffic from a host behind Spoke-1 to 172.31.200.200.
5. Run a sniffer trace on Spoke-1. Because of `sla-stickiness`, the existing traffic is kept on H3_T11:

```
Spoke-1 (root) # diagnose sniffer packet any 'host 172.31.200.200' 4
interfaces=[any]
```

```
filters=[host 172.31.200.200]

0.202708 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
0.202724 H3_T11 out 10.0.3.2 -> 172.31.200.200: icmp: echo request
0.202911 H3_T11 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
0.202934 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
```

Test case 6: Hub-1 has an in SLA path to external peers

Since Hub-1 has an in SLA path to external peers, it will advertise the external route with destination 172.31.200.200/32 to Spoke-1.

To verify the configuration:

1. Verify the health check status on Hub-1. Note that port4 meets SLA, but port5 does not:

```
PoP1-Hub1 (root) # diagnose sys sdwan health-check
Health Check(external_peers):
Seq(1 port4): state(alive), packet-loss(0.000%) latency(0.161), jitter(0.009), mos
(4.404), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(2 port5): state(dead), packet-loss(100.000%) sla_map=0x0
```

2. Verify the SD-WAN neighbor status. The `minimum-sla-meet-members` threshold of 1 is still met:

```
PoP1-Hub1 (root) # diagnose sys sdwan neighbor
Neighbor(EDGE): member(1 2)role(standalone)
        Health-check(external_peers:1)  sla-pass selected alive
```

3. Verify the BGP learned routes. Hub-1 still advertises the external route to the Spoke-1 BGP neighbor:

```
PoP1-Hub1 (root) # get router info bgp neighbors 172.31.0.65 advertised-routes
VRF 0 BGP table version is 13, local router ID is 172.31.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network                 Next Hop      Metric     LocPrf  Weight  RouteTag Path
*>i172.31.200.200/32     172.31.0.1     100        32768        0           i <-/->
Total number of prefixes 1
```

Test case 7: all external peers on Hub-1 are out of SLA

In this case, Hub-1 will now advertise the default route map, which denies the advertisement of the external route. Spoke-1 will now route traffic to the next hub.

To verify the configuration:

1. Verify the health check status on Hub-1. Note that port4 and port5 do not meet SLA:

```
PoP1-Hub1 (root) # diagnose sys sdwan health-check
Health Check(external_peers):
Seq(1 port4): state(dead), packet-loss(100.000%) sla_map=0x0
Seq(2 port5): state(dead), packet-loss(100.000%) sla_map=0x0
```

2. Verify the SD-WAN neighbor status. The `minimum-sla-meet-members` threshold of 1 is not met:

```
PoP1-Hub1 (root) # diagnose sys sdwan neighbor
Neighbor(EDGE): member(1 2)role(standalone)
        Health-check(external_peers:1)  sla-fail dead
```

3. Verify the BGP learned routes. Hub-1 does not advertise any external routes to the Spoke-1 BGP neighbor:

```
PoP1-Hub1 (root) # get router info bgp neighbors 172.31.0.65 advertised-routes
% No prefix for neighbor 172.31.0.65
```

Active dynamic BGP neighbor triggered by ADVPN shortcut - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:
- Active dynamic BGP neighbor triggered by ADVPN shortcut

When a customer using SD-WAN with ADVPN has numerous IPv4 and IPv6 routes per spoke and there are many spokes in the topology, using ADVPN with a route reflector-based design poses the following challenges:

- The hub FortiGate will experience high CPU usage due to the amount of processing required to reflect the routes to the spoke FortiGates.
- Spoke FortiGates will learn many unnecessary routes.

For such cases, it is more suitable to deploy an IPv4- and IPv6-supported solution without a route-reflector that involves an active dynamic BGP neighbor triggered by an ADVPN shortcut. This solution allows a spoke FortiGate to form a BGP neighbor with another spoke FortiGate only after the shortcut tunnel between them has been established. As a result, the spoke only learns routes from its BGP neighbors.

How this solution differs from typical SD-WAN with ADVPN

In a topology where the Spoke 1 and Spoke 2 FortiGates are connected directly to the Hub FortiGate, route reflection will not be enabled. The Hub FortiGate is only configured with each spoke's summary route. An ADVPN shortcut tunnel is established between the Spoke 1 and Spoke 2 FortiGates. The valid routing between the Spoke 1 and Spoke 2 FortiGate is still through the Hub FortiGate at this point.

When a host behind Spoke 1 tries to connect to a host behind Spoke 2, Spoke 1 first reaches the Hub based on the valid routing table. The Hub determines that the destination is reachable, and the ADVPN shortcut tunnel between the spokes is established. Then, Spoke 1 and Spoke 2 will actively initiate a BGP connection to each other over the shortcut. Once established, they will exchange their routing information using BGP. On both spokes, BGP will resolve those routes on the shortcut and update the routing table accordingly.

For this solution, the following IPv4/IPv6 BGP configuration settings are required:

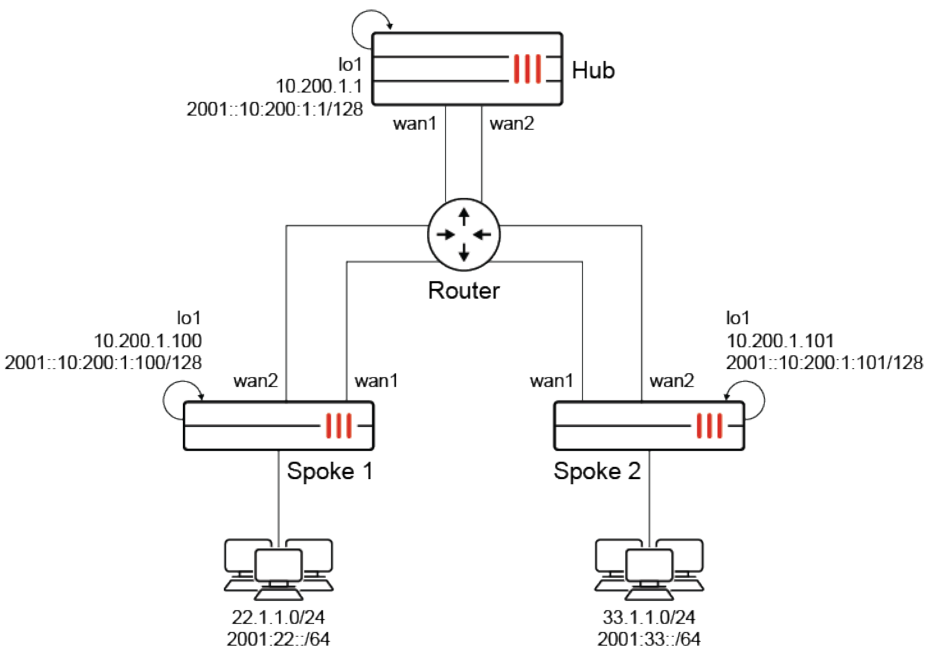
- The hub FortiGate should be configured with `neighbor-group` and `neighbor-range`/`neighbor-range6`.
- Each spoke FortiGate should be configured with `neighbor-group` and `neighbor-range`/`neighbor-range6` like the hub. More importantly, each spoke should be configured with `set passive disable` to ensure spokes are able to initiate dynamic BGP connections between each other.
- The hub FortiGate should have route reflection disabled (by default) where each `neighbor-group` setting should have `set route-reflector-client disable`.

In the configuration, each of the spokes will form a BGP neighbor relationship with the hub. This is unchanged from the typical SD-WAN with ADVPN configuration.

Example

This example configuration contains the following structure:

- Use SD-WAN member 1 (via ISP1) and its dynamic shortcuts for Financial Department traffic.
- Use SD-WAN member 2 (via ISP2) and its dynamic shortcuts for Engineering Department traffic.
- Internal subnets of Spoke 1:
  - IPv4: 22.1.1.0/24
  - IPv6: 2001:22::/64
- Internal subnets of Spoke 2:
  - IPv4: 33.1.1.0/24
    - Financial Department: 33.1.1.1 to 33.1.1.100
    - Engineering Department: 33.1.1.101 to 33.1.1.200
  - IPv6: 2001:33::/64
    - Financial Department: 2001:33::1 to 2001:33::100
    - Engineering Department: 2001:33::101 to 2001:33::200



To configure the Hub FortiGate:

1. Configure the BGP settings (neighbor group and ranges):

```
config router bgp
    set as 65100
    set router-id 10.200.1.1
    set ibgp-multipath enable
    config neighbor-group
        edit "EDGE"
            set activate6 disable
            set remote-as 65100
            set update-source "lo1"
```

```
            set route-reflector-client disable
        next
        edit "EDGEv6"
            set activate disable
            set remote-as 65100
            set update-source "lo1"
            set route-reflector-client disable
        next
    end
    config neighbor-range
        edit 2
            set prefix 10.200.1.0 255.255.255.0
            set neighbor-group "EDGE"
        next
    end
    config neighbor-range6
        edit 2
            set prefix6 2001::10:200:1:0/112
            set neighbor-group "EDGEv6"
        next
    end
    config network
        edit 2
            set prefix 10.200.1.0 255.255.255.0
        next
        edit 4
            set prefix 33.0.0.0 255.0.0.0
        next
        edit 5
            set prefix 22.0.0.0 255.0.0.0
        next
    end
    config network6
        edit 4
            set prefix6 2001:33::/32
        next
        edit 2
            set prefix6 2001:22::/32
        next
    end
end
```

2. Configure the static routes.
   a. For IPv4:

```
config router static
    edit 33
        set dst 33.0.0.0 255.0.0.0
        set blackhole enable
        set vrf 0
    next
    edit 22
        set dst 22.0.0.0 255.0.0.0
        set blackhole enable
        set vrf 0
```

```
    next
end
```

b. For IPv6:

```
config router static6
    edit 33
        set dst 2001:33::/32
        set blackhole enable
        set vrf 0
    next
    edit 22
        set dst 2001:22::/32
        set blackhole enable
        set vrf 0
    next
end
```

The following IPv4 summary routes are advertised:

- 33.0.0.0/8
- 22.0.0.0/8

The following IPv6 summary routes are advertised:

- 2001:33::/32
- 2001:22::/32

Because route reflection has been disabled in this example, initially, Spoke 1 will not know the local subnet of Spoke 2, and Spoke 2 will not know the local subnet of Spoke 1. Therefore, for traffic routing, summary routes are configured on the hub as blackhole routes and then advertised to the spokes using BGP.

For example, for traffic from the local subnet of Spoke 2 destined for the local subnet of Spoke 1:

- For the IPv4 case, the summary route 22.0.0.0/8, which includes the local subnet of Spoke 1 (22.1.1.0/24), is advertised to Spoke 2. When Spoke 2 sends traffic destined for 22.1.1.0/24 to the Hub, the Hub forwards this traffic to Spoke 1 since they are BGP neighbors.
- For the IPv6 case, the summary route 2001:22::/32, which includes the local subnet of Spoke 1 (2001:22::/64), is advertised to Spoke 2. When Spoke 2 sends traffic destined for 2001:22::/64 to the Hub, the Hub forwards this traffic to Spoke 1 since they are BGP neighbors.

Although traffic from spoke-to-spoke goes through the hub first, it is expected that the spoke will eventually go through the shortcut tunnel.

To configure the Spoke 1 FortiGate:

1. Configure the SD-WAN settings:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "spoke1-1"
            set cost 10
```

```
        next
        edit 2
            set interface "spoke-2"
            set cost 20
        next
    end
    config health-check
        edit "ping"
            set server "11.11.11.11"
            set source 10.200.1.100
            set members 1 2
            config sla
                edit 1
                    set latency-threshold 200
                    set jitter-threshold 50
                next
            end
        next
    end
    config service
        edit 1
            set dst "financial-department"
            set priority-members 1
        next
        edit 2
            set dst "engineering-department"
            set priority-members 2
        next
        edit 61
            set addr-mode ipv6
            set priority-members 1
            set dst6 "financial-department-IPv6"
        next
        edit 62
            set addr-mode ipv6
            set priority-members 2
            set dst6 "engineering-department-IPv6"
        next
    end
end
```

2. Configure the BGP settings (neighbor group and ranges):

```
config router bgp
    set as 65100
    set router-id 10.200.1.100
    set ibgp-multipath enable
    config neighbor
        edit "10.200.1.1"
            set activate6 disable
            set remote-as 65100
            set connect-timer 10
            set update-source "lo1"
        next
        edit "2001::10:200:1:1"
            set advertisement-interval 1
            set activate disable
```

```
                set remote-as 65100
                set update-source "lo1"
            next
        end
        config neighbor-group
            edit "spokes"
                set activate6 disable
                set passive disable
                set remote-as 65100
                set update-source "lo1"
            next
            edit "spokesv6"
                set activate disable
                set passive disable
                set remote-as 65100
                set update-source "lo1"
            next
        end
        config neighbor-range
            edit 1
                set prefix 10.200.1.0 255.255.255.0
                set neighbor-group "spokes"
            next
        end
        config neighbor-range6
            edit 1
                set prefix6 2001::10:200:1:0/112
                set neighbor-group "spokesv6"
            next
        end
        config network
            edit 3
                set prefix 22.1.1.0 255.255.255.0
            next
        end
        config network6
            edit 1
                set prefix6 2001:22::/64
            next
        end
    end
```

Verifying the configuration before a spoke-to-spoke shortcut VPN is established

IPv4 use case

To verify the status on Spoke 1:

1. Verify the BGP status:

```
# get router info bgp summary
VRF 0 BGP router identifier 10.200.1.100, local AS number 65100
BGP table version is 5
1 BGP AS-PATH entries
0 BGP community entries
```

```
Neighbor   V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.200.1.1 4      65100     222     225        3    0    0 00:15:14        3
Total number of neighbors 1
```

2. Verify the BGP routing table:

```
# get router info routing-table bgp
Routing table for VRF=0
B       11.11.11.11/32 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11),
00:15:19
                                              (recursive via spoke1-2 tunnel
111.1.1.11), 00:15:19, [1/0]
B       22.0.0.0/8 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11),
00:15:19
                                          (recursive via spoke1-2 tunnel 111.1.1.11),
00:15:19, [1/0]
B       33.0.0.0/8 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11),
00:15:19
                                          (recursive via spoke1-2 tunnel 111.1.1.11),
00:15:19, [1/0]
```

IPv6 use case

To verify the status on Spoke 1:

1. Verify the BGP status:

```
# get router info6 bgp summary
VRF 0 BGP router identifier 10.200.1.100, local AS number 65100
BGP table version is 6
1 BGP AS-PATH entries
0 BGP community entries
Neighbor         V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001::10:200:1:1 4      65100     223     224        4    0    0 00:15:21        3
Total number of neighbors 1
```

2. Verify the BGP routing table:

```
# get router info6 routing-table bgp
Routing table for VRF=0
B       2001::11:11:11:11/128 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1
tunnel ::11.1.1.11), 00:15:29
                                                           (recursive via spoke1-2
tunnel ::111.1.1.11), 00:15:29, [1024/0]
B       2001:22::/32 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1 tunnel
::11.1.1.11), 00:15:29
                                                  (recursive via spoke1-2 tunnel
::111.1.1.11), 00:15:29, [1024/0]
B       2001:33::/32 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1 tunnel
::11.1.1.11), 00:15:29
                                                  (recursive via spoke1-2 tunnel
::111.1.1.11), 00:15:29, [1024/0]
```

Verifying the configuration after a single spoke-to-spoke shortcut VPN is established

IPv4 use case

To trigger a single spoke-to-spoke shortcut VPN, on host 22.1.1.22, ping the host 33.1.1.33 in the Financial Department. Because of the SD-WAN rule, use SD-WAN member 1 (via ISP1) and its dynamic shortcuts to reach hosts in the Financial Department.

To verify the status on Spoke 1:

1. Verify the BGP status:

```
# get router info bgp summary
VRF 0 BGP router identifier 10.200.1.100, local AS number 65100
BGP table version is 6
1 BGP AS-PATH entries
0 BGP community entries
Neighbor      V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.200.1.1    4      65100     252     254        3    0    0 00:17:22        3
10.200.1.101 4      65100       6       6        5    0    0 00:00:14        1
Total number of neighbors 2
```

   Spoke 1 has as its BGP neighbors:
   - Hub FortiGate at 10.200.1.1
   - Spoke 2 FortiGate at 10.200.1.101

2. Verify the BGP routing table:

```
# get router info routing-table bgp
Routing table for VRF=0
B       11.11.11.11/32 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11),
00:17:26
                                              (recursive via spoke1-2 tunnel
111.1.1.11), 00:17:26, [1/0]
B       22.0.0.0/8 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11),
00:17:26
                                          (recursive via spoke1-2 tunnel 111.1.1.11),
00:17:26, [1/0]
B       33.0.0.0/8 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11),
00:17:26
                                          (recursive via spoke1-2 tunnel 111.1.1.11),
00:17:26, [1/0]
B       33.1.1.0/24 [200/0] via 10.200.1.101 (recursive via spoke1-1_0 tunnel 13.1.1.3),
00:00:18, [1/0]
```

   The remote route learned from Spoke 2 through the spoke1_1_0 tunnel and using BGP is 33.1.1.0/24.

IPv6 use case

To trigger a single spoke-to-spoke shortcut VPN over IPv6, on host 2001:22::22/64, ping the host 2001:33::33/64 in the Financial Department. Because of the SD-WAN rule, use SD-WAN member 1 (via ISP1) and its dynamic shortcuts to reach hosts in the Financial Department.

To verify the status on Spoke 1:

1. Verify the BGP status:

```
# get router info6 bgp summary
VRF 0 BGP router identifier 10.200.1.100, local AS number 65100
BGP table version is 7
1 BGP AS-PATH entries
0 BGP community entries
Neighbor           V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001::10:200:1:1   4      65100     253     254        4    0    0 00:17:28        3
2001::10:200:1:101 4      65100       7       7        6    0    0 00:00:21        1
Total number of neighbors 2
```

Spoke 1 has as its BGP neighbors:
   - Hub FortiGate at 2001::10:200:1:1
   - Spoke 2 FortiGate at 2001::10:200:1:101

2. Verify the BGP routing table:

```
# get router info6 routing-table bgp
Routing table for VRF=0
B       2001::11:11:11:11/128 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1
tunnel ::11.1.1.11), 00:17:30
                                                           (recursive via spoke1-2
tunnel ::111.1.1.11), 00:17:30, [1024/0]
B       2001:22::/32 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1 tunnel
::11.1.1.11), 00:17:30
                                                  (recursive via spoke1-2 tunnel
::111.1.1.11), 00:17:30, [1024/0]
B       2001:33::/32 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1 tunnel
::11.1.1.11), 00:17:30
                                                  (recursive via spoke1-2 tunnel
::111.1.1.11), 00:17:30, [1024/0]
B       2001:33::/64 [200/0] via 2001::10:200:1:101 (recursive via spoke1-1_0 tunnel
::13.1.1.3), 00:00:24, [1024/0]
```

The remote route learned from Spoke 2 through the spoke1-1_0 tunnel and using BGP is 2001:33::/64.

Verifying the configuration after a second spoke-to-spoke shortcut VPN is established

IPv4 use case

To trigger a second spoke-to-spoke shortcut VPN, on host 22.1.1.22, ping the host 33.1.1.133 in the Engineering Department. Because of the SD-WAN rule, use SD-WAN member 2 (via ISP2) and its dynamic shortcuts to reach hosts in the Engineering Department.

To verify the status on Spoke 1:

1. Verify the BGP status:

```
# get router info bgp summary
VRF 0 BGP router identifier 10.200.1.100, local AS number 65100
BGP table version is 6
1 BGP AS-PATH entries
0 BGP community entries
```

```
Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.200.1.1   4      65100     263     265        3    0    0 00:18:12        3
10.200.1.101 4      65100      17      17        5    0    0 00:01:04        1
Total number of neighbors
```

Spoke 1 continues to have its BGP neighbors:

- Hub FortiGate at 10.200.1.1
- Spoke 2 FortiGate at 10.200.1.101

2. Verify the BGP routing table:

```
# get router info routing-table bgp
Routing table for VRF=0
B       11.11.11.11/32 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11),
00:18:17
                                               (recursive via spoke1-2 tunnel
111.1.1.11), 00:18:17, [1/0]
B       22.0.0.0/8 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11),
00:18:17
                                           (recursive via spoke1-2 tunnel 111.1.1.11),
00:18:17, [1/0]
B       33.0.0.0/8 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11),
00:18:17
                                           (recursive via spoke1-2 tunnel 111.1.1.11),
00:18:17, [1/0]
B       33.1.1.0/24 [200/0] via 10.200.1.101 (recursive via spoke1-1_0 tunnel 13.1.1.3),
00:01:09
                                              (recursive via spoke1-2_0 tunnel
113.1.1.3), 00:01:09, [1/0]
```

The remote route learned from Spoke 2 through the spoke1-2_0 tunnel and using BGP is 33.1.1.0/24.

IPv6 use case

To trigger a second spoke-to-spoke shortcut VPN over IPv6, on host 2001:22::22/64, ping the host 2001:33::133/64 in the Engineering Department. Because of the SD-WAN rule, use SD-WAN member 2 (via ISP2) and its dynamic shortcuts to reach hosts in the Engineering Department.

To verify the status on Spoke 1:

1. Verify the BGP status:

```
# get router info6 bgp summary
VRF 0 BGP router identifier 10.200.1.100, local AS number 65100
BGP table version is 7
1 BGP AS-PATH entries
0 BGP community entries
Neighbor           V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001::10:200:1:1   4      65100     264     265        4    0    0 00:18:18        3
2001::10:200:1:101 4      65100      19      19        6    0    0 00:01:11        1
Total number of neighbors 2
```

Spoke 1 continues to have its BGP neighbors:

- Hub FortiGate at 2001::10:200:1:1
- Spoke 2 FortiGate at 2001::10:200:1:101

2. Verify the BGP routing table:

```
# get router info6 routing-table bgp
Routing table for VRF=0
B       2001::11:11:11:11/128 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1
tunnel ::11.1.1.11), 00:18:20
                                                           (recursive via spoke1-2
tunnel ::111.1.1.11), 00:18:20, [1024/0]
B       2001:22::/32 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1 tunnel
::11.1.1.11), 00:18:20
                                                  (recursive via spoke1-2 tunnel
::111.1.1.11), 00:18:20, [1024/0]
B       2001:33::/32 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1 tunnel
::11.1.1.11), 00:18:20
                                                  (recursive via spoke1-2 tunnel
::111.1.1.11), 00:18:20, [1024/0]
B       2001:33::/64 [200/0] via 2001::10:200:1:101 (recursive via spoke1-1_0 tunnel
::13.1.1.3), 00:01:14
                                                    (recursive via spoke1-2_0 tunnel
::113.1.1.3), 00:01:14, [1024/0]
```

The remote route learned from Spoke 2 through the spoke1-2_0 tunnel and using BGP is 2001:33::/64.

Performance SLA

This section includes information about performance SLA related new features:

- Logging FortiMonitor-detected performance metrics on page 246
- Classifying SLA probes for traffic prioritization on page 249
- VRF-aware SD-WAN IPv6 health checks on page 254
- Support maximize bandwidth (SLA) to load balance spoke-to-spoke traffic between multiple ADVPN shortcuts on page 255
- Support HTTPS performance SLA health checks 7.4.1 on page 263

Logging FortiMonitor-detected performance metrics

This information is also available in the FortiOS 7.4 Administration Guide:
- SD-WAN application monitor using FortiMonitor

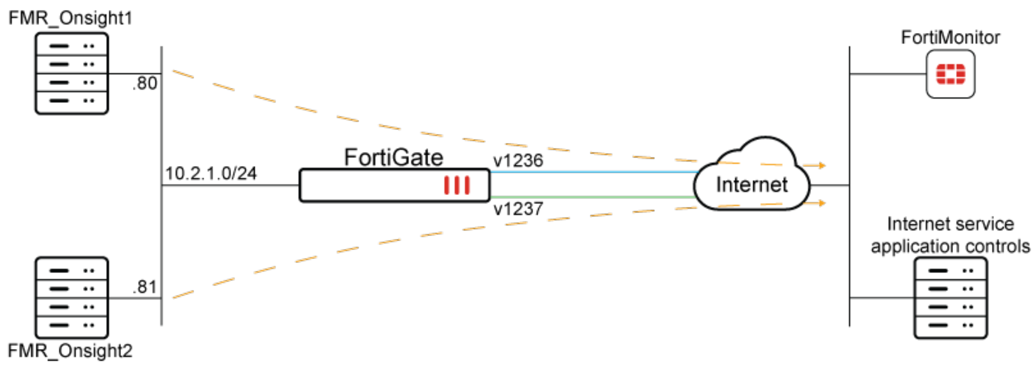
FortiGate can log statistics when using FortiMonitor to detect advanced SD-WAN application performance metrics. These logs may also be sent to FortiAnalyzer and FortiManager for review and reporting.

You can control the logging frequency using the `app-perf-log-period` command:

```
config system sdwan
    set app-perf-log-period <time in seconds>
end
```

Example

This example is based on the following topology:



To configure logging of FortiMonitor-detected performance metrics:

1. Configure the address objects for each FortiMonitor client:

```
config firewall address
    edit "FMR_OnSight1"
        set subnet 10.2.1.80 255.255.255.255
    next
    edit "FMR_OnSight2"
        set subnet 10.2.1.81 255.255.255.255
    next
end
```

2. Set the logging frequency:

```
config system sdwan
    set status enable
    set app-perf-log-period 60
end
```

3. Configure the SD-WAN zone and members:

```
config system sdwan
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "v1236"
            set gateway 10.12.36.2
        next
        edit 2
            set interface "v1237"
            set gateway 10.12.37.20
        next
    end
end
```

4. Configure the SD-WAN rules:

```
config system sdwan
    config service
        edit 1
            set dst "all"
            set src "FMR_OnSight1"
            set priority-members 2
            set agent-exclusive enable
        next
        edit 2
            set dst "all"
            set src "FMR_OnSight2"
            set priority-members 1
            set agent-exclusive enable
        next
    end
end
```

5. Configure the SD-WAN health check:

```
config system sdwan
    config health-check
        edit "FMR"
            set detect-mode agent-based
            set probe-timeout 60000
            set recoverytime 1
            set members 1 2
            config sla
                edit 1
                next
            end
        next
    end
end
```

To verify SD-WAN member performance and review logs:

1. Verify the health check diagnostics:

```
# diagnose sys sdwan health-check
    Health Check(FMR):
    Seq(1 v1236): state(alive), packet-loss(0.000%) latency(200.099), jitter(0.201), mos
(4.171), bandwidth-up(999989), bandwidth-dw(999983), bandwidth-bi(1999972) sla_map=0x0
    Seq(2 v1237): state(alive), packet-loss(0.000%) latency(200.103), jitter(0.391), mos
(4.169), bandwidth-up(999994), bandwidth-dw(999981), bandwidth-bi(1999975) sla_map=0x0
```

2. Review the SD-WAN logs:

```
# execute log filter category event
# execute log filter field subtype sdwan
# execute log display

1: date=2023-01-27 time=16:32:15 eventtime=1674865935918381398 tz="-0800"
logid="0113022937" type="event" subtype="sdwan" level="information" vd="root"
logdesc="Virtuan WAN Link application performance metrics via FortiMonitor"
eventtype="Application Performance Metrics" app="fortinet.com" appid=0 interface="v1237"
latency="200.2" jitter="0.6" packetloss="0.0" serverresponsetime="827.7"
networktransfertime="107.7" apperror="0.0" timestamp="01-28 00:31:59" msg="Application
Performance Metrics via FortiMonitor"
```

```
2: date=2023-01-27 time=16:32:15 eventtime=1674865935918367770 tz="-0800"
logid="0113022937" type="event" subtype="sdwan" level="information" vd="root"
logdesc="Virtuan WAN Link application performance metrics via FortiMonitor"
eventtype="Application Performance Metrics" app="fortinet.com" appid=0 interface="v1236"
latency="200.0" jitter="0.3" packetloss="0.0" serverresponsetime="870.6"
networktransfertime="130.4" apperror="0.0" timestamp="01-28 00:31:59" msg="Application
Performance Metrics via FortiMonitor"

3: date=2023-01-27 time=16:31:15 eventtime=1674865875917685437 tz="-0800"
logid="0113022937" type="event" subtype="sdwan" level="information" vd="root"
logdesc="Virtuan WAN Link application performance metrics via FortiMonitor"
eventtype="Application Performance Metrics" app="fortinet.com" appid=0 interface="v1237"
latency="200.5" jitter="0.7" packetloss="0.0" serverresponsetime="1008.9"
networktransfertime="129.8" apperror="0.0" timestamp="01-28 00:31:02" msg="Application
Performance Metrics via FortiMonitor"

4: date=2023-01-27 time=16:31:15 eventtime=1674865875917672824 tz="-0800"
logid="0113022937" type="event" subtype="sdwan" level="information" vd="root"
logdesc="Virtuan WAN Link application performance metrics via FortiMonitor"
eventtype="Application Performance Metrics" app="fortinet.com" appid=0 interface="v1236"
latency="200.3" jitter="0.8" packetloss="0.0" serverresponsetime="825.4"
networktransfertime="106.4" apperror="0.0" timestamp="01-28 00:31:02" msg="Application
Performance Metrics via FortiMonitor"

5: date=2023-01-27 time=16:30:15 eventtime=1674865815912801725 tz="-0800"
logid="0113022937" type="event" subtype="sdwan" level="information" vd="root"
logdesc="Virtuan WAN Link application performance metrics via FortiMonitor"
eventtype="Application Performance Metrics" app="fortinet.com" appid=0 interface="v1237"
latency="200.1" jitter="0.4" packetloss="0.0" serverresponsetime="845.4"
networktransfertime="116.0" apperror="0.0" timestamp="01-28 00:30:01" msg="Application
Performance Metrics via FortiMonitor"

6: date=2023-01-27 time=16:30:15 eventtime=1674865815912786458 tz="-0800"
logid="0113022937" type="event" subtype="sdwan" level="information" vd="root"
logdesc="Virtuan WAN Link application performance metrics via FortiMonitor"
eventtype="Application Performance Metrics" app="fortinet.com" appid=0 interface="v1236"
latency="200.0" jitter="0.3" packetloss="0.0" serverresponsetime="1032.0"
networktransfertime="138.9" apperror="0.0" timestamp="01-28 00:30:01" msg="Application
Performance Metrics via FortiMonitor"
```

Classifying SLA probes for traffic prioritization

This information is also available in the FortiOS 7.4 Administration Guide:

- Classifying SLA probes for traffic prioritization

Support for traffic classification on SLA probes has been implemented to ensure they are prioritized in times of congestion. This prevents SD-WAN link flapping and unexpected routing behaviors, and stabilizes SD-WAN from unnecessary failovers.

SLA probes can now be classified into a specific class ID so that SLA probes assigned to a class ID with higher priority are prioritized over other traffic. SLA probes are assigned using the `class-id` command:

```
config system sdwan
    config health-check
        edit <health-check name>
            set class-id <class name>
        next
    end
end
```

Example

In this example, SLA probes are assigned into different class ID. The interfaces dmz and vd1-01 both have outbandwidth of 1000000 Kbps (1 Gbps) configured. Three traffic shaping classes are defined:

Class ID	Name	Definition
2	sla_probe	High priority with a guaranteed 10% of bandwidth (100 Mbps)
3	default	Low priority with a guaranteed 80% of bandwidth (800 Mbps)
4	sla_probe_2	Medium priority with a guaranteed 10% of bandwidth (100 Mbps)

Under this scheme, when congestion occurs, traffic in each class will have their guaranteed bandwidth honored. If there is remaining bandwidth, higher priority traffic will get the bandwidth. On the SD-WAN health check, probes to server 2.2.2.2 are assigned to class 2 (sla_probe). This means it has a guaranteed bandwidth and has the highest priority to use unused bandwidth. This allows SD-WAN health check to function properly even during times of congestion.

To classify SLA probes for traffic prioritization:

1. Configure the firewall traffic class:

```
config firewall traffic-class
    edit 2
        set class-name "sla_probe"
    next
    edit 3
        set class-name "default"
    next
    edit 4
        set class-name "sla_probe_2"
    next
end
```

2. Configure the class ID priority and guaranteed bandwidth:

```
config firewall shaping-profile
    edit "profile-1"
        set default-class-id 3
        config shaping-entries
            edit 2
                set class-id 2
                set priority high
                set guaranteed-bandwidth-percentage 10
```

```
                set maximum-bandwidth-percentage 100
            next
            edit 3
                set class-id 3
                set priority low
                set guaranteed-bandwidth-percentage 80
                set maximum-bandwidth-percentage 100
            next
            edit 4
                set class-id 4
                set priority medium
                set guaranteed-bandwidth-percentage 10
                set maximum-bandwidth-percentage 100
            next
        end
    next
end
```

3. Configure the interfaces:

```
config system interface
    edit "dmz"
        set outbandwidth 1000000
        set egress-shaping-profile "profile-1"
        ...
    next
    edit "vd1-p1"
        set outbandwidth 1000000
        set egress-shaping-profile "profile-1"
        ...
    next
end
```

4. Configure the SD-WAN health check and assign the SLA probes into class 2:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "dmz"
            set gateway 172.16.208.2
        next
        edit 2
            set interface "vd1-p1"
        next
    end
    config health-check
        edit "1"
            set server "2.2.2.2"
            set members 1 2
            set class-id 2
            config sla
                edit 1
```

```
                next
            end
        next
    end
end
```

To verify the SLA probe assignment:

1. Verify the health check diagnostics:

```
diagnose sys sdwan health-check
    Health Check(1):
    Seq(1 dmz): state(alive), packet-loss(0.000%) latency(0.247), jitter(0.022), mos
(4.404), bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
    Seq(2 vd1-p1): state(alive), packet-loss(0.000%) latency(0.247), jitter(0.018), mos
(4.404), bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x1
```

2. Verify the SLA probes are assigned into class 2:

```
# diagnose netlink interface list dmz
    if=dmz family=00 type=1 index=5 mtu=1500 link=0 master=0
    ref=36 state=start present fw_flags=10018000 flags=up broadcast run multicast
    Qdisc=mq hw_addr=e0:23:ff:9d:f9:9e broadcast_addr=ff:ff:ff:ff:ff:ff
    egress traffic control:
            bandwidth=1000000(kbps) lock_hit=0 default_class=3 n_active_class=3
            class-id=3      allocated-bandwidth=800000(kbps)        guaranteed-
bandwidth=800000(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=1(kbps)
                            priority=low    forwarded_bytes=1446
                            dropped_packets=0       dropped_bytes=0
            class-id=4      allocated-bandwidth=100000(kbps)        guaranteed-
bandwidth=100000(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=0(kbps)
                            priority=medium         forwarded_bytes=0
                            dropped_packets=0       dropped_bytes=0
            class-id=2      allocated-bandwidth=100000(kbps)        guaranteed-
bandwidth=100000(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=1(kbps)
                            priority=high   forwarded_bytes=1404
                            dropped_packets=0       dropped_bytes=0
    stat: rxp=19502 txp=14844 rxb=2233923 txb=802522 rxe=0 txe=0 rxd=0 txd=0 mc=0
collision=0 @ time=1675121675
    re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
    te: txa=0 txc=0 txfi=0 txh=0 txw=0
    misc rxc=0 txc=0
    input_type=0 state=3 arp_entry=0 refcnt=36
# diagnose netlink interface list vd1-p1
    if=vd1-p1 family=00 type=768 index=99 mtu=1420 link=0 master=0
    ref=20 state=start present fw_flags=10010000 flags=up p2p run noarp multicast
    Qdisc=noqueue
    egress traffic control:
            bandwidth=1000000(kbps) lock_hit=0 default_class=3 n_active_class=3
            class-id=3      allocated-bandwidth=800000(kbps)        guaranteed-
bandwidth=800000(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=0(kbps)
                            priority=low    forwarded_bytes=0
                            dropped_packets=0       dropped_bytes=0
```

```
            class-id=4      allocated-bandwidth=100000(kbps)        guaranteed-
bandwidth=100000(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=0(kbps)
                            priority=medium         forwarded_bytes=0
                            dropped_packets=0       dropped_bytes=0
            class-id=2      allocated-bandwidth=100000(kbps)        guaranteed-
bandwidth=100000(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=1(kbps)
                            priority=high   forwarded_bytes=1120
                            dropped_packets=0       dropped_bytes=0
    stat: rxp=4097 txp=4586 rxb=540622 txb=221500 rxe=0 txe=19 rxd=0 txd=0 mc=0
collision=0 @ time=1675121742
    re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
    te: txa=0 txc=0 txfi=0 txh=0 txw=0
    misc rxc=0 txc=0
    input_type=0 state=3 arp_entry=0 refcnt=20
```

When verifying the class assignment, the counter value should increase.

The example also demonstrates assigning SLA probes to class 4 (sla_probe_2), in which case the probes get medium priority.

To assign the SLA probe to medium priority:

1. Assign SLA probes into class 4:

```
config sys sdwan
    config health-check
        edit 1
            set class-id 4
        next
    end
    set status disable
end
config sys sdwan
    set status enable
end
```

2. Verify the SLA probes are assigned into class 4.

```
# diagnose netlink interface list dmz
    if=dmz family=00 type=1 index=5 mtu=1500 link=0 master=0
    ref=34 state=start present fw_flags=10018000 flags=up broadcast run multicast
    Qdisc=mq hw_addr=e0:23:ff:9d:f9:9e broadcast_addr=ff:ff:ff:ff:ff:ff
    egress traffic control:
            bandwidth=1000000(kbps) lock_hit=0 default_class=3 n_active_class=3
            class-id=3      allocated-bandwidth=800000(kbps)        guaranteed-
bandwidth=800000(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=1(kbps)
                            priority=low    forwarded_bytes=24K
                            dropped_packets=0       dropped_bytes=0
            class-id=4      allocated-bandwidth=100000(kbps)        guaranteed-
bandwidth=100000(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=1(kbps)
```

```
                                priority=medium         forwarded_bytes=1674
                                dropped_packets=0       dropped_bytes=0
                class-id=2      allocated-bandwidth=100000(kbps)        guaranteed-
bandwidth=100000(kbps)
                                max-bandwidth=1000000(kbps)     current-bandwidth=0(kbps)
                                priority=high   forwarded_bytes=0
                                dropped_packets=0       dropped_bytes=0
    stat: rxp=20818 txp=15874 rxb=2382789 txb=857674 rxe=0 txe=0 rxd=0 txd=0 mc=0
collision=0 @ time=1675122057
    re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
    te: txa=0 txc=0 txfi=0 txh=0 txw=0
    misc rxc=0 txc=0
    input_type=0 state=3 arp_entry=0 refcnt=34
# diagnose netlink interface list vd1-p1
    if=vd1-p1 family=00 type=768 index=99 mtu=1420 link=0 master=0
    ref=20 state=start present fw_flags=10010000 flags=up p2p run noarp multicast
    Qdisc=noqueue
    egress traffic control:
            bandwidth=1000000(kbps) lock_hit=0 default_class=3 n_active_class=3
            class-id=3      allocated-bandwidth=800000(kbps)        guaranteed-
bandwidth=800000(kbps)
                                max-bandwidth=1000000(kbps)     current-bandwidth=0(kbps)
                                priority=low    forwarded_bytes=0
                                dropped_packets=0       dropped_bytes=0
            class-id=4      allocated-bandwidth=100000(kbps)        guaranteed-
bandwidth=100000(kbps)
                                max-bandwidth=1000000(kbps)     current-bandwidth=1(kbps)
                                priority=medium         forwarded_bytes=1280
                                dropped_packets=0       dropped_bytes=0
            class-id=2      allocated-bandwidth=100000(kbps)        guaranteed-
bandwidth=100000(kbps)
                                max-bandwidth=1000000(kbps)     current-bandwidth=0(kbps)
                                priority=high   forwarded_bytes=0
                                dropped_packets=0       dropped_bytes=0
    stat: rxp=4097 txp=4703 rxb=540622 txb=226180 rxe=0 txe=19 rxd=0 txd=0 mc=0
collision=0 @ time=1675122058
    re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
    te: txa=0 txc=0 txfi=0 txh=0 txw=0
    misc rxc=0 txc=0
    input_type=0 state=3 arp_entry=0 refcnt=20
```

VRF-aware SD-WAN IPv6 health checks

VRF and source can be configured in SD-WAN IPv6 health checks.

```
config system sdwan
   config health-check
      edit <name>
         set addr-mode ipv6
         set vrf <vrf id>
         set source6 <IPv6 address>
      next
   end
end
```

This example shows how to configure VRF and source for SD-WAN IPv6 health check on a standalone FortiGate.

To configure the VRF and source for SD-WAN IPv6 health check:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "R150"
            set gateway 10.100.1.1
            set gateway6 2000:10:100:1::1
        next
        edit 2
            set interface "R160"
            set gateway 10.100.1.5
            set gateway6 2000:10:100:1::5
        next
    end
    config health-check
        edit "ping6"
            set addr-mode ipv6
            set server "2000:10:100:2::22"
            set vrf 10
            set source6 2000:10:100:1::2
            set members 1 2
        next
    end
end
```

If an SD-WAN member can reach the server, but not on VRF 10, then it is dead:

```
# diagnose sys sdwan health-check
Health Check(ping6):
Seq(1 R150): state(alive), packet-loss(0.000%) latency(0.042), jitter(0.022), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x0
Seq(2 R160): state(dead), packet-loss(100.000%) sla_map=0x0
```

Only the SD-WAN member with the proper VRF route can have the protocol 17 route, so the VRF is functioning correctly:

```
# diagnose ipv6 route list | grep protocol=17
vf=0 tbl=10 type=01(unicast) protocol=17(fortios) flag=00000000 prio=1024
src:2000:10:100:1::2/128-> dst:2000:10:100:2::22/128 gwy:2000:10:100:1::1 dev=48(R150)
pmtu=1500
```

Support maximize bandwidth (SLA) to load balance spoke-to-spoke traffic between multiple ADVPN shortcuts

This information is also available in the FortiOS 7.4 Administration Guide:

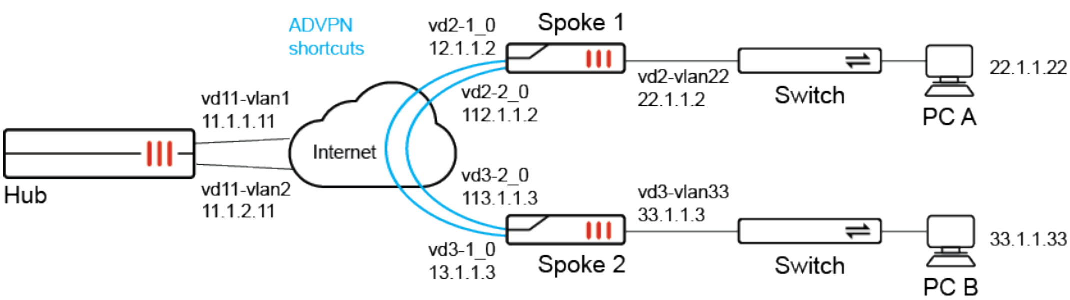
- Use maximize bandwidth to load balance traffic between ADVPN shortcuts

When ADVPN is configured on a FortiGate spoke along with an SD-WAN rule set to *Maximize Bandwidth SLA* (GUI) or load balance mode (CLI) as well as `tie-break` set to `fib-best-match`, then spoke-to-spoke traffic is load balanced between multiple ADVPN shortcuts when the shortcuts are within the configured SLA conditions.

Following is an example configuration with `set mode load-balance` and `set tie-break fib-best-match` enabled:

```
config system sdwan
    config service
        edit 3
            set mode load-balance
            set dst "all"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 1 2
            set tie-break fib-best-match
        next
    end
end
```

Example



In this example SD-WAN is configured between one hub and multiple spokes, and the SD-WAN configuration shows SD-WAN rule 3 with the following required settings to enable spoke-to-spoke traffic between multiple ADVPN shortcuts:

- set mode load-balance
- set tie-break fib-best-match

```
show system sdwan
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "zon2"
        next
    end
    config members
        edit 1
            set interface "vd2-1"
            set cost 10
        next
```

```
        edit 2
            set interface "vd2-2"
            set cost 20
        next
    end
    config health-check
        edit "ping"
            set server "11.11.11.11"
            set members 1 2
            config sla
                edit 1
                    set latency-threshold 200
                    set jitter-threshold 50
                next
                edit 2
                next
            end
        next
        edit "1"
        next
    end
    config service
        edit 1
            set dst "033"
            set priority-members 1
        next
        edit 2
            set dst "133"
            set priority-members 2
        next
        edit 3
            set mode load-balance
            set dst "all"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 1 2
            set tie-break fib-best-match
        next
    end
end
```

To trigger spoke-to-spoke communication, run an ICMP ping on PC A with IP address 22.1.1.22 behind spoke 1 that is destined for PC B with IP address 33.1.1.33 behind spoke 2. The spoke-to-spoke traffic will be used to demonstrate load balancing between shortcuts in the CLI output of this topic.

To verify the configuration:

1. Confirm the ADVPN shortcuts are within the SLA conditions:

```
# diagnose system sdwan health-check
Health Check(ping):
Seq(1 vd2-1): state(alive), packet-loss(0.000%) latency(0.029), jitter(0.002), mos
(4.404), bandwidth-up(1999), bandwidth-dw(0), bandwidth-bi(1999) sla_map=0x3
Seq(1 vd2-1_0): state(alive), packet-loss(0.000%) latency(0.026), jitter(0.001), mos
```

```
(4.404), bandwidth-up(2000), bandwidth-dw(0), bandwidth-bi(2000) sla_map=0x3
Seq(2 vd2-2): state(alive), packet-loss(0.000%) latency(0.055), jitter(0.064), mos
(4.404), bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x3
Seq(2 vd2-2_0): state(alive), packet-loss(0.000%) latency(0.060), jitter(0.058), mos
(4.404), bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x3
```

2. Confirm the settings for SD-WAN rule 3:

```
# diagnose system sdwan service 3

Service(3): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: fib
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance  hash-mode=round-robin)
  Member sub interface(4):
    1: seq_num(1), interface(vd2-1):
       1: vd2-1_0(125)
    3: seq_num(2), interface(vd2-2):
       1: vd2-2_0(127)
  Members(4):
    1: Seq_num(1 vd2-1), alive, sla(0x1), gid(2), num of pass(1), selected
    2: Seq_num(1 vd2-1_0), alive, sla(0x1), gid(2), num of pass(1), selected
    3: Seq_num(2 vd2-2), alive, sla(0x1), gid(2), num of pass(1), selected
    4: Seq_num(2 vd2-2_0), alive, sla(0x1), gid(2), num of pass(1), selected
  Dst address(1):
        0.0.0.0-255.255.255.255
```

3. Confirm firewall policing routing list:

```
# diagnose firewall proute list 2131230723
list route policy info(vf=vd2):

id=2131230723(0x7f080003) vwl_service=3 vwl_mbr_seq=1 1 2 2 dscp_tag=0xfc 0xfc
flags=0x90 load-balance hash-mode=round-robin  fib-best-match tos=0x00 tos_mask=0x00
protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(4) oif=116(vd2-1) num_pass=1
oif=125(vd2-1_0) num_pass=1 oif=117(vd2-2) num_pass=1 oif=127(vd2-2_0) num_pass=1
destination(1): 0.0.0.0-255.255.255.255
source wildcard(1): 0.0.0.0/0.0.0.0
hit_count=117 last_used=2023-04-21 15:49:59
```

4. Confirm the routing table:

```
# get router info routing-table bgp
Routing table for VRF=0
B*      0.0.0.0/0 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11),
01:26:14, [1/0]
                  [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11),
01:26:14, [1/0]
B       1.1.1.1/32 [200/0] via 11.1.1.1 [2] (recursive via 12.1.1.1, vd2-vlan12),
01:26:14, [1/0]
B       11.11.11.11/32 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11),
01:26:14, [1/0]
                       [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11),
01:26:14, [1/0]
B       33.1.1.0/24 [200/0] via 10.10.100.3 [2] (recursive is directly connected, vd2-1_
0), 01:19:41, [1/0]
                    [200/0] via 10.10.200.3 [2] (recursive is directly connected, vd2-2_
0), 01:19:41, [1/0]
B       100.1.1.0/24 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11),
```

```
01:26:14, [1/0]
                     [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11),
01:26:14, [1/0]
```

5. Check the packet sniffer output for the default setting.

   This step demonstrates routing for the default setting of `set tie-break zone`. The following packet sniffer output of ICMP pings demonstrates how spoke-to-spoke traffic (ping from 22.1.1.22 to 33.1.1.13) is load balanced between all parent tunnels and shortcuts, and is not limited to shortcuts within SLA.

```
# diagnose sniffer packet any "host 33.1.1.13" 4
interfaces=[any]
filters=[host 33.1.1.13]
14.665232 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665234 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665240 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665262 vd2-1_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665274 vd3-1_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665284 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665285 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665289 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665299 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665300 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665306 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665314 vd3-1_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665326 vd2-1_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665331 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665332 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665337 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

24.190955 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.190957 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.190963 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.190982 vd2-2 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.190993 p2 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191002 p2 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191020 vd3-2 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191031 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191032 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191036 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191046 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191047 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191053 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191063 vd3-2 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191074 p2 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191079 p2 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191090 vd2-2 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191094 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191095 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191100 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

51.064984 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.064985 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.064991 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065011 vd2-2_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065022 vd3-2_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065031 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
```

```
51.065032 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065036 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065046 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065047 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065054 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065063 vd3-2_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065075 vd2-2_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065082 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065082 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065087 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

67.257123 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257125 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257131 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257150 vd2-1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257162 p1 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257170 p1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257189 vd3-1 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257199 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257200 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257205 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257216 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257217 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257223 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257234 vd3-1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257245 p1 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257250 p1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257261 vd2-1 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257266 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257267 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257272 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

^C
84 packets received by filter
0 packets dropped by kernel
```

6. Check the sniffer packet output after changing the setting to `set tie-break fib-best-match`.

   The following packet sniffer output of ICMP pings demonstrates how load balancing of spoke-to-spoke is limited and only occurs between shortcuts vd2-1_0 and vd2-2_0, which are within SLA.

```
# diagnose sniffer packet any "host 33.1.1.13" 4

interfaces=[any]
filters=[host 33.1.1.13]
2.592392 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592394 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592400 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592420 vd2-1_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592432 vd3-1_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592441 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592442 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592447 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592484 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592485 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592491 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592498 vd3-1_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592510 vd2-1_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
```

```
2.592515 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592516 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592520 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

8.808792 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808793 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808799 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808816 vd2-2_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808827 vd3-2_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808838 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808838 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808842 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808852 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808853 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808858 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808866 vd3-2_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808877 vd2-2_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808882 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808883 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808887 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

18.024377 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024379 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024385 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024400 vd2-1_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024411 vd3-1_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024421 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024422 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024427 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024436 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024437 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024443 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024449 vd3-1_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024459 vd2-1_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024463 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024464 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024468 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

24.216469 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216470 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216477 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216493 vd2-2_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216506 vd3-2_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216518 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216519 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216525 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216535 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216536 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216542 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216548 vd3-2_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216559 vd2-2_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216563 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216564 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216568 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
^C
70 packets received by filter
```

```
0 packets dropped by kernel
```

7. Check SD-WAN heath.

   When an ADVPN shortcut is out of SLA, traffic does not run on it. Shortcut vd2-1_0 is out of SLA.

```
# diagnose system sdwan health-check
Health Check(ping):
Seq(1 vd2-1): state(alive), packet-loss(6.000%) latency(0.026), jitter(0.001), mos
(4.401), bandwidth-up(1999), bandwidth-dw(0), bandwidth-bi(1999) sla_map=0x0
Seq(1 vd2-1_0): state(alive), packet-loss(18.182%) latency(0.033), jitter(0.003), mos
(4.395), bandwidth-up(2000), bandwidth-dw(0), bandwidth-bi(2000) sla_map=0x0
Seq(2 vd2-2): state(alive), packet-loss(0.000%) latency(0.024), jitter(0.001), mos
(4.404), bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x3
Seq(2 vd2-2_0): state(alive), packet-loss(0.000%) latency(0.033), jitter(0.005), mos
(4.404), bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x3
```

8. Check the sniffer packet:

   No traffic runs on Shortcut vd2-1_0 because it is out of SLA.

```
# diagnose sniffer packet any "host 33.1.1.13" 4
interfaces=[any]
filters=[host 33.1.1.13]
8.723075 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723077 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723084 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723103 vd2-2_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723115 vd3-2_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723148 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723149 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723154 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723166 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723166 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723171 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723179 vd3-2_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723190 vd2-2_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723195 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723195 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723199 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

17.202681 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202683 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202688 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202704 vd2-2_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202716 vd3-2_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202727 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202728 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202733 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202742 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202743 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202749 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202755 vd3-2_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202767 vd2-2_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202771 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202772 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202777 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
```

Support HTTPS performance SLA health checks - 7.4.1

HTTPS is supported for SD-WAN performance SLA health checks. All default HTTP-based health checks have been updated to use HTTPS instead. This includes:

- Default_AWS
- Default_FortiGuard
- Default_Google Search
- Default_Office_365

After upgrading, the default profiles using HTTP are changed to use HTTPS. Non-default performance SLA health check profiles are not affected after upgrading.

Example 1: applying a default HTTPS health check:

In this example, the Default_AWS health check is applied to an SD-WAN member in the default virtual-wan-link zone.

To apply the Default_AWS health check in an SD-WAN configuration:

1. Configure SD-WAN:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "port1"
            set gateway 172.16.200.254
            set gateway6 2000:172:16:200::254
        next
    end
    config health-check
        edit "Default_AWS"
            set server "aws.amazon.com"
            set protocol https
            set interval 1000
            set probe-timeout 1000
            set recoverytime 10
            set update-static-route disable
            set members 1
            config sla
                edit 1
                    set latency-threshold 250
                    set jitter-threshold 50
                    set packetloss-threshold 5
                next
            end
        next
```

```
    end
end
```

2. Verify the health check status:

```
# diagnose sys sdwan health-check status Default_AWS
Health Check(Default_AWS):
Seq(1 port1): state(alive), packet-loss(0.000%) latency(107.732), jitter(10.425), mos
(4.332), bandwidth-up(999920), bandwidth-dw(997555), bandwidth-bi(1997475) sla_map=0x1
```

Example 2: configuring an IPv6 health check with HTTPS

To configure an IPv6 health check with HTTPS:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "port1"
            set gateway 172.16.200.254
            set gateway6 2000:172:16:200::254
        next
    end
    config health-check
        edit "ipv6"
            set addr-mode ipv6
            set server "ipv6.google.com"
            set protocol https
            set members 1
            config sla
                edit 1
                    set latency-threshold 250
                    set jitter-threshold 50
                    set packetloss-threshold 5
                next
            end
        next
    end
end
```

Service rules

This section includes information about service rule related new features:

- Support IPv6 application based steering in SD-WAN on page 265
- Allow multicast traffic to be steered by SD-WAN on page 269
- Using load balancing in a manual SD-WAN rule without configuring an SLA target 7.4.1 on page 283

Support IPv6 application based steering in SD-WAN

This information is also available in the FortiOS 7.4 Administration Guide:

- Internet service and application control steering

IPv6 based SD-WAN rules allow matching of applications and application categories. The following options are available with `set addr-mode ipv6`:

```
config system sdwan
   config service
     edit
       set addr-mode ipv6
       set internet-service enable
       set internet-service-app-ctrl
       set internet-service-app-ctrl-group
       set internet-service-app-ctrl-category
     next
   end
end
```

Example

In this example, SD-WAN is configured to use an IPv6 service rule to steer traffic from FGT_A to FGT_B based on the following application control options:

- Application Telnet
- An application group for ping
- An application category that includes SSH

When the rule is matched, traffic is steered based on the lowest cost SLA strategy. In this example, vlan100 is the preferred interface, and traffic is routed to vlan100 on FGT_B.

To view the configuration:

1. View the SD-WAN configuration on FGT_A:
   SD-WAN has four members in the default virtual-wan-link zone, each with an IPv4 and IPv6 gateway. The SD-WAN service rule includes `internet-service-app-ctrl 16091` for the Telnet, `internet-service-app-ctrl-group "network-Ping"` for ping , and `internet-service-app-ctrl-category 15` for SSH applications.

```
(sdwan) # show
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "dmz"
            set gateway 172.16.208.2
            set gateway6 2000:172:16:208::2
```

```
        next
        edit 2
            set interface "IPSec-1"
        next
        edit 3
            set interface "agg1"
            set gateway 172.16.203.2
            set gateway6 2000:172:16:203::2
        next
        edit 4
            set interface "vlan100"
            set gateway 172.16.206.2
            set gateway6 2000:172:16:206::2
        next
    end
    config health-check
        edit "1"
            set addr-mode ipv6
            set server "2000::2:2:2:2"
            set members 0
            config sla
                edit 1
                next
            end
        next
    end
    config service
        edit 1
            set name "1"
            set addr-mode ipv6
            set mode sla
            set internet-service enable
            set internet-service-app-ctrl 16091
            set internet-service-app-ctrl-group "network-Ping"
            set internet-service-app-ctrl-category 15
            config sla
                edit "1"
                    set id 1
                next
            end
            set priority-members 4 1 2 3
        next
    end
end
```

2. View the default route for FGT_A:

```
config router static
    edit 5
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
end
```

3. View the firewall policy for FGT_A:

   The `utm-status` option is enabled to learn application 3T (3 tuple) information, and the default application profile of `g-default` is selected.

```
config firewall policy
    edit 1
        set uuid f09bddc4-def3-51ed-8517-0d8b6bc18f35
        set srcintf "any"
        set dstintf "any"
        set action accept
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set application-list "g-default"
    next
end
```

To verify the configuration:

1. On FGT_A, check the routing table:

   The routing table has ECMP applied to default gateways for each SD-WAN member.

   ```
   # get router info routing-table  static
   Routing table for VRF=0
   S*      0.0.0.0/0 [1/0] via 172.16.203.2, agg1, [1/0]
                            [1/0] via 172.16.206.2, vlan100, [1/0]
                            [1/0] via 172.16.208.2, dmz, [1/0]
                            [1/0] via IPSec-1 tunnel 172.16.209.2, [1/0]
   ```

2. Check the SD-WAN service:

   Based on the service rule, member 4 named vlan100 is preferred. Traffic must also match the highlighted internet services.

   ```
   # diagnose system sdwan service

   Service(1): Address Mode(IPV6) flags=0x4200 use-shortcut-sla use-shortcut
    Tie break: cfg
     Gen(2), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
     Members(4):
       1: Seq_num(4 vlan100), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), 
   selected
       2: Seq_num(1 dmz), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
       3: Seq_num(2 IPSec-1), alive, sla(0x1), gid(0), cfg_order(2), local cost(0), 
   selected
       4: Seq_num(3 agg1), alive, sla(0x1), gid(0), cfg_order(3), local cost(0), selected
     Internet Service(3): Telnet(4294837974,0,0,0,0 16091) IPv6.ICMP(4294837087,0,0,0,0 
   16321) Network.Service(0,15,0,0,0)
   ```

3. Initiate traffic for ping, Telnet, and SSH to FGT_B, then FGT_A will learn 3T information for these applications, and use the SD-WAN rule to route traffic for the applications to the preferred interface of vlan100.
   - Following is the sniffer traffic for ping application. The ping traffic flows out of DMZ before 3T information is recognized, then out from vlan100 after T3 traffic is recognized:

     ```
     # diagnose sniffer packet any 'host 2000::2:0:0:4' 4
     interfaces=[any]
     filters=[host 2000::2:0:0:4]
     16.952138 port5 in 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 1
     ```

```
[flowlabel 0x5080d]
16.954571 dmz out 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 1
[flowlabel 0x5080d]
16.954920 dmz in 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 1
16.955086 port5 out 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 1
17.953277 port5 in 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 2
[flowlabel 0x5080d]
17.953455 dmz out 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 2
[flowlabel 0x5080d]
17.953622 dmz in 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 2
17.953722 port5 out 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 2
18.959823 port5 in 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 3
[flowlabel 0x5080d]
18.960005 vlan100 out 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq
3 [flowlabel 0x5080d]
18.960015 agg1 out 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 3
[flowlabel 0x5080d]
18.960024 port4 out 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 3
[flowlabel 0x5080d]
18.960295 vlan100 in 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 3
18.960449 port5 out 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 3
19.983802 port5 in 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 4
[flowlabel 0x5080d]
```

- Following is the sniffer traffic for Telnet application group. The Telnet traffic flows out of agg1 before 3T information is recognized, then out from vlan100 after T3 traffic is recognized:

```
# diagnose sniffer packet any 'host 2000::2:0:0:4 and dst port 23' 4
interfaces=[any]
filters=[host 2000::2:0:0:4 and dst port 23]
4.096393 port5 in 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: syn 2723132265
[flowlabel 0xd4e65]
4.096739 agg1 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: syn 2723132265
[flowlabel 0xd4e65]
4.096752 port4 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: syn 2723132265
[flowlabel 0xd4e65]
.........
5.503679 port5 in 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: psh 2723132345 ack
544895389  [flowlabel 0xd4e65]
5.503894 vlan100 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: psh 2723132345
ack 544895389  [flowlabel 0xd4e65]
5.503907 agg1 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: psh 2723132345 ack
544895389  [flowlabel 0xd4e65]
5.503918 port4 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: psh 2723132345 ack
544895389  [flowlabel 0xd4e65]
5.504641 port5 in 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: ack 544895390
[flowlabel 0xd4e65]
5.504713 vlan100 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: ack 544895390
[flowlabel 0xd4e65]
5.504721 agg1 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: ack 544895390
[flowlabel 0xd4e65]
5.504728 port4 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: ack 544895390
[flowlabel 0xd4e65]
```

- Following is the sniffer traffic for SSH application category. The SSH traffic flows out of dmz before 3T information is recognized, then out from vlan100 after T3 traffic is recognized:

```
# diagnose sniffer packet any 'host 2000::2:0:0:4 and dst port 22' 4
interfaces=[any]
filters=[host 2000::2:0:0:4 and dst port 22]
5.910752 port5 in 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: syn 980547187
[flowlabel 0xf1403]
5.911002 dmz out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: syn 980547187
[flowlabel 0xf1403]
5.914550 port5 in 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583860244
[flowlabel 0xf1403]
5.914651 dmz out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583860244
[flowlabel 0xf1403]
.....
8.116507 port5 in 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: psh 980549261 ack
583862554  [class 0x10] [flowlabel 0xf1403]
8.116663 vlan100 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: psh 980549261
ack 583862554  [class 0x10] [flowlabel 0xf1403]
8.116674 agg1 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: psh 980549261 ack
583862554  [class 0x10] [flowlabel 0xf1403]
8.116685 port4 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: psh 980549261 ack
583862554  [class 0x10] [flowlabel 0xf1403]
8.118135 port5 in 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583862598
[class 0x10] [flowlabel 0xf1403]
8.118171 vlan100 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583862598
[class 0x10] [flowlabel 0xf1403]
8.118179 agg1 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583862598
[class 0x10] [flowlabel 0xf1403]
8.118189 port4 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583862598
[class 0x10] [flowlabel 0xf1403]
```

4. View the IPv6 application control internet service ID list:

```
# diagnose system sdwan internet-service-app-ctrl6-list

Telnet(16091 4294837974): 2000::2:0:0:4 6 23 Thu Apr 20 17:43:00 2023
IPv6.ICMP(16321 4294837087): 2000::2:0:0:4 58 0 Thu Apr 20 17:43:00 2023
```

5. View the IPv6 application control internet service ID list by category:

```
# diagnose system sdwan internet-service-app-ctrl6-category-list

SSH(16060 4294837772): 2000::2:0:0:4 6 22 Thu Apr 20 17:43:00 2023
```

Allow multicast traffic to be steered by SD-WAN

This information is also available in the FortiOS 7.4 Administration Guide:

- Use SD-WAN rules to steer multicast traffic

SD-WAN rules can now steer multicast traffic. When an SD-WAN member is out of SLA, multicast traffic can fail over to another SD-WAN member, and switch back when SLA recovers.

The new `pim-use-sdwan` option enables or disables the use of SD-WAN for PIM (Protocol Independent Multicast) when checking RP (Rendezvous Point) neighbors and sending packets.

```
config router multicast
    config pim-sm-global
        set pim-use-sdwan {enable | disable}
    end
end
```

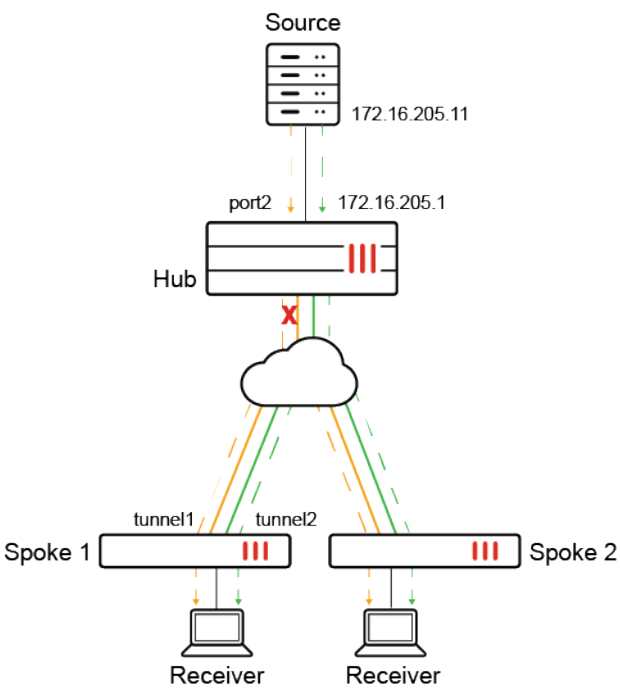
When SD-WAN steers multicast traffic, ADVPN is not supported. Use the `set shortcut` option to disable shortcuts for the service:

```
config system sdwan
    config service
        edit <id>
            set shortcut {enable | disable}
        next
    end
end
```

Example 1

In this hub and spoke example, the PIM source is behind the hub FortiGate, and the RP is set to internal port (port2) of the hub firewall. Each spoke connects to the two WAN interfaces on the hub by using an overlay tunnel. The overlay tunnels are members of SD-WAN.

Receivers behind the spoke FortiGates request a stream from the source to receive traffic on tunnel1 by default. When the overlay tunnel goes out of SLA, the multicast traffic fails over to tunnel2 and continues to flow.



Following is an overview of how to configure the topology:

1. Configure the hub FortiGate in front of the PIM source. The RP is configured on internal port (port2) of the hub FortiGate.

2. Configure the spoke FortiGates.
3. Verify traffic failover.

To configure the hub:

1. On the hub, enable multicast routing, configure the multicast RP, and enable PIM sparse mode on each interface:

```
config router multicast
    set multicast-routing enable
    config pim-sm-global
        config rp-address
            edit 1
                set ip-address 172.16.205.1
            next
        end
    end
    config interface
        edit "tport1"
            set pim-mode sparse-mode
        next
        edit "tagg1"
            set pim-mode sparse-mode
        next
        edit "port2"
            set pim-mode sparse-mode
        next
    end
end
```

To configure each spoke:

1. Enable SD-WAN with the following settings:
   - Configure the overlay tunnels as member of the SD-WAN zone.
   - Configure a performance SLA health-check using ping.
   - Configure a service rule for the PIM protocol with the following settings:
     - Use the lowest cost (SLA) strategy.
     - Monitor with the ping health-check.
   - Disable ADVPN shortcut.

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "tunnel1"
        next
        edit 2
            set interface "tunnel2"
        next
    end
    config health-check
```

```
        edit "ping"
            set server "172.16.205.1"
            set update-static-route disable
            set members 0
            config sla
                edit 1
                next
            end
        next
    end
    config service
        edit 1
            set mode sla
            set protocol 103
            set dst "all"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 1 2
            set use-shortcut-sla disable
            set shortcut disable
        next
        edit 2
            set mode sla
            set dst "all"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
```

2. Enable multicast routing and configure the multicast RP. Enable PIM sparse-mode on each interface:

```
config router multicast
    set multicast-routing enable
    config pim-sm-global
        set spt-threshold disable
        set pim-use-sdwan enable
        config rp-address
            edit 1
                set ip-address 172.16.205.1
            next
        end
    end
    config interface
        edit "tunnel1"
            set pim-mode sparse-mode
        next
        edit "tunnel2"
            set pim-mode sparse-mode
        next
```

```
        edit "port4"
            set pim-mode sparse-mode
        next
    end
end
```

To verify traffic failover:

With this configuration, multicast traffic starts on tunnel1. When tunnel1 becomes out of SLA, traffic switches to tunnel2. When tunnel1 is in SLA again, the traffic switches back to tunnel1.

The following health-check capture on the spokes shows tunnel1 in SLA with packet-loss (1.000%):

```
# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel1): state(alive), packet-loss(0.000%) latency(0.056), jitter(0.002), mos(4.404),
bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x1
Seq(2 tunnel2): state(alive), packet-loss(0.000%) latency(0.100), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1

# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel1): state(alive), packet-loss(1.000%) latency(0.056), jitter(0.002), mos(4.404),
bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x1
Seq(2 tunnel2): state(alive), packet-loss(0.000%) latency(0.100), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1
```

The following example shows tunnel1 out of SLA with packet-loss (3.000%):

```
# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel1): state(alive), packet-loss(3.000%) latency(0.057), jitter(0.003), mos(4.403),
bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x0
Seq(2 tunnel2): state(alive), packet-loss(0.000%) latency(0.101), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1
```

The following example shows tunnel1 back in SLA again:

```
# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel1): state(alive), packet-loss(1.000%) latency(0.061), jitter(0.004), mos(4.404),
bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x0
Seq(2 tunnel2): state(alive), packet-loss(0.000%) latency(0.102), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1

# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel1): state(alive), packet-loss(0.000%) latency(0.061), jitter(0.004), mos(4.404),
bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x0
Seq(2 tunnel2): state(alive), packet-loss(0.000%) latency(0.102), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1
```

The following example how traffic switches to tunnel2 while tunnel1 health-check is out of SLA. Source (172.16.205.11) sends traffic to the multicast group. Later the traffic switches back to tunnel1 once SLA returns to normal:

```
195.060797 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
195.060805 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
```

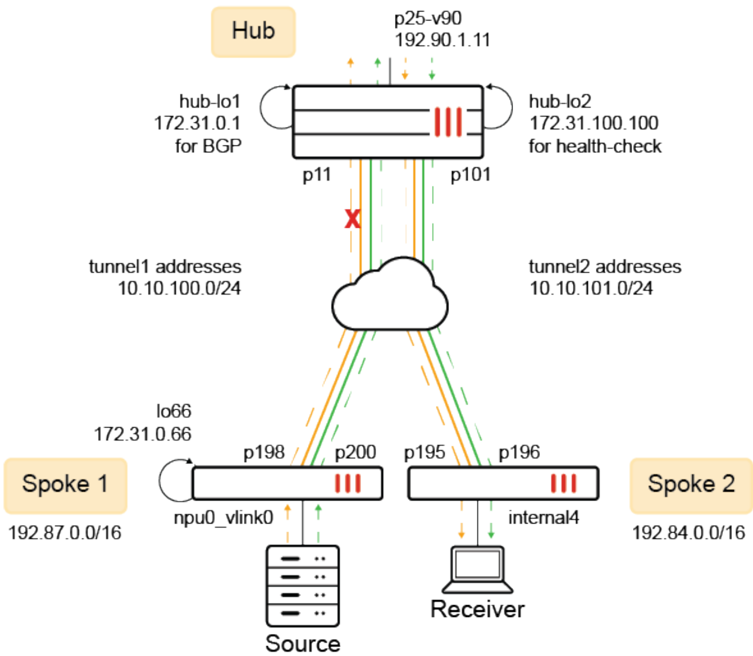
```
196.060744 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
196.060752 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
197.060728 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
197.060740 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
198.060720 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
198.060736 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
199.060647 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
199.060655 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
200.060598 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
200.060604 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
... ...
... ...
264.060974 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
265.060950 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
265.060958 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
266.060867 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
266.060877 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
267.060828 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
267.060835 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
268.060836 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
268.060854 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
269.060757 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
269.060767 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
270.060645 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
270.060653 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
```

Example 2

In this hub and spoke example, the PIM source is behind spoke 1, and the RP is configured on the hub FortiGate. BGP is used for routing. The hub uses embedded SLA in ICMP probes to determine the health of each tunnel, allowing it to prioritize healthy IKE routes.

The receiver is on another spoke. Upon requesting a stream, source passes the traffic to the RP on the hub FortiGate, and routes the traffic to the receiver over tunnel1. If a tunnel falls out of SLA, the multicast traffic fails over to the other tunnel.

In this configuration, SD-WAN steers multicast traffic by using embedded SLA information in ICMP probes. See also Embedded SD-WAN SLA information in ICMP probes. With this feature, the hub FortiGate can use the SLA information of the spoke's health-check to control BGP and IKE routes over tunnels.



Following is an overview of how to configure the topology:

1. Configure the hub FortiGate. The RP is configured on the hub FortiGate.
2. Configure the spoke FortiGate in front of the traffic receiver.
3. Configure the spoke FortiGate in front of the PIM source.

To configure the hub:

1. Configure loopbacks hub-lo1 172.31.0.1 for BGP and hub-lo100 172.31.100.100 for health-check:

```
config system interface
    edit "hub-lo1"
        set vdom "hub"
        set ip 172.31.0.1 255.255.255.255
        set allowaccess ping
        set type loopback
        set snmp-index 82
    next
    edit "hub-lo100"
        set vdom "hub"
        set ip 172.31.100.100 255.255.255.255
        set allowaccess ping
        set type loopback
        set snmp-index 81
    next
end
```

2. Enable multicast routing with the following settings:
   - Configure internal interface p25-v90 as RP.
   - Enable interfaces for PIM sparse-mode.

```
config router multicast
    set multicast-routing enable
```

```
    config pim-sm-global
        config rp-address
            edit 1
                set ip-address 192.90.1.11
            next
        end
    end
    config interface
        edit "p11"
            set pim-mode sparse-mode
        next
        edit "p101"
            set pim-mode sparse-mode
        next
        edit "p25-v90"
            set pim-mode sparse-mode
        next
    end
end
```

3. Enable SD-WAN with the following settings:
   - Add interfaces p11 and p101 as members.
   - Configure embedded SLA health-checks to detect ICMP probes from each overlay tunnel. Prioritize based on the health of each tunnel.

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "p11"
        next
        edit 2
            set interface "p101"
        next
    end
    config health-check
        edit "1"
            set detect-mode remote
            set probe-timeout 60000
            set recoverytime 1
            set sla-id-redistribute 1
            set members 1
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                    set priority-in-sla 10
                    set priority-out-sla 20
                next
            end
        next
        edit "2"
```

```
            set detect-mode remote
            set probe-timeout 60000
            set recoverytime 1
            set sla-id-redistribute 1
            set members 2
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                    set priority-in-sla 15
                    set priority-out-sla 25
                next
            end
        next
    end
end
```

4. Configure BGP to peer with neighbors. Neighbor group is configured for tunnel interface IP addresses:

```
config router bgp
    set as 65505
    set router-id 172.31.0.1
    set ibgp-multipath enable
    set additional-path enable
    set recursive-inherit-priority enable
    config neighbor-group
        edit "gr1"
            set remote-as 65505
            set update-source "hub-lo1"
            set additional-path both
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 1
            set prefix 10.10.0.0 255.255.0.0
            set neighbor-group "gr1"
        next
        edit 66
            set prefix 172.31.0.66 255.255.255.255
            set neighbor-group "gr1"
        next
    end
    config network
        ....
        edit 90
            set prefix 192.90.0.0 255.255.0.0
        next
    end
end
```

To configure the spoke (in front of the receiver):

1. Enable multicast routing to use SD-WAN. Configure the RP address. Enable interfaces for PIM sparse-mode.

```
config router multicast
    set multicast-routing enable
    config pim-sm-global
        set spt-threshold disable
        set pim-use-sdwan enable
        config rp-address
            edit 1
                set ip-address 192.90.1.11
            next
        end
    end
    config interface
        edit "p195"
            set pim-mode sparse-mode
        next
        edit "p196"
            set pim-mode sparse-mode
        next
        edit "internal4"
            set pim-mode sparse-mode
            set static-group "225-1-1-122"
        next
    end
end
```

2. Configure SD-WAN with the following settings:
   - Add overlay tunnel interfaces as members.
   - Configure a performance SLA health-check to send ping probes to the hub.
   - Configure a service rule for the PIM protocol. Use the lowest cost (SLA) strategy, and monitor with the ping health-check.
   - Disable ADVPN shortcuts.

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 6
            set interface "p196"
        next
        edit 5
            set interface "p195"
        next
    end
```

```
    config health-check
        edit "ping"
            set server "172.31.100.100"
            set update-static-route disable
            set members 0
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
    config service
        edit 1
            set mode sla
            set protocol 103
            set dst "all"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 5 6
            set use-shortcut-sla disable
            set shortcut disable
        next
        edit 2
            set mode sla
            set dst "all"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 5 6
        next
    end
end
```

3. Configure BGP and set neighbors to the overlay gateway IP address on the hub:

```
config router bgp
    set as 65505
    set router-id 122.1.1.122
    set ibgp-multipath enable
    set additional-path enable
    config neighbor
        edit "10.10.100.254"
            set soft-reconfiguration enable
```

```
            set remote-as 65505
            set connect-timer 10
            set additional-path both
        next
        edit "10.10.101.254"
            set soft-reconfiguration enable
            set remote-as 65505
            set connect-timer 10
            set additional-path both
        next
    end
    config network
        edit 3
            set prefix 192.84.0.0 255.255.0.0
        next
    end
end
```

4. Configure the default gateway to use the SD-WAN zone. Other routes are for the underlay to route traffic to the hub's WAN interfaces:

```
config router static
    edit 10
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
    ....
    next
end
```

To configure the spoke (in front of the source):

1. Enable multicast routing to use SD-WAN. Configure the RP address. Enable interfaces for PIM sparse-mode:

```
config router multicast
    set multicast-routing enable
    config pim-sm-global
        set pim-use-sdwan enable
        config rp-address
            edit 1
                set ip-address 192.90.1.11
            next
        end
    end
    config interface
        edit "p198"
            set pim-mode sparse-mode
        next
        edit "p200"
            set pim-mode sparse-mode
        next
        edit "npu0_vlink0"
            set pim-mode sparse-mode
```

```
        next
    end
end
```

2. Configure loopback interface lo66 for BGP and sourcing SD-WAN traffic:

```
config system interface
    edit "lo66"
        set vdom "root"
        set ip 172.31.0.66 255.255.255.255
        set allowaccess ping
        set type loopback
        set snmp-index 21
    next
end
```

3. Configure SD-WAN:
   - Add overlay tunnel interfaces as members.
   - Configure a performance SLA health-check to send ping probes to the hub.
   - Configure a service rule for the PIM protocol. Use the lowest cost (SLA) strategy, and monitor with the ping health-check.
   - Disable the use of an ADVPN shortcut.

   In the following example, 11.11.11.11 is the underlay address for one of the WAN links on the hub, and 172.31.100.100 is the loopback address on the server.

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "overlay"
        next
    end
    config members
        edit 1
            set interface "p198"
            set zone "overlay"
            set source 172.31.0.66
        next
        edit 2
            set interface "p200"
            set zone "overlay"
            set source 172.31.0.66
        next
    end
    config health-check
        edit "ping"
            set server "11.11.11.11"
            set members 0
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
```

```
        edit "HUB"
            set server "172.31.100.100"
            set embed-measured-health enable
            set members 0
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
     config service
        edit 1
            set mode sla
            set protocol 103
            set dst "all"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 1 2
            set use-shortcut-sla disable
            set shortcut disable
        next
        edit 2
            set mode sla
            set dst "all"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
```

4. Configure BGP:

```
config router bgp
    set as 65505
    set router-id 123.1.1.123
    set ibgp-multipath enable
    set additional-path enable
    config neighbor
        edit "172.31.0.1"
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65505
            set update-source "lo66"
        next
    end
    config network
        edit 3
            set prefix 192.87.0.0 255.255.0.0
```

```
        next
    end
end
```

5. Configure the default gateway to use the SD-WAN zone. Other routes are for the underlay to route to the hub's WAN interfaces:

```
config router static
    edit 10
        set distance 1
        set sdwan-zone "virtual-wan-link" "overlay"
    next
    ...
    next
end
```

Using load balancing in a manual SD-WAN rule without configuring an SLA target - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Load balancing strategy
- Load balancing strategy without SLA targets
- Load balancing strategy with SLA targets

The maximize bandwidth (`load-balance`) strategy used prior to FortiOS 7.4.1 is now known as the load balancing strategy. This strategy can be configured under the manual mode and the lowest cost (SLA) strategies.

- When the load balancing strategy is configured under the manual mode strategy, SLA targets are not used.
- When the load balancing strategy is configured under the lowest cost (SLA) strategy, SLA targets are used.

Policy and objects

This section includes information about policy and object related new features:

- NGFW on page 284
- Policies on page 287
- Objects on page 315
- Traffic shaping on page 319
- Protocol options on page 328

NGFW

This section includes information about NGFW policy mode related new features:

- Add scanunit support for learning mode on page 284
- Support dynamic Fabric address in security policies 7.4.1 on page 285

Add scanunit support for learning mode

The scanunit provides a more powerful file detection mechanism through full-scanning in learning mode. This improves the accuracy of the IPS engine in detecting malicious files.

The following use cases demonstrate using the scanunit to process full-scanning in learning mode with the corresponding UTM logs.

Use case 1: antivirus

In this example, the scanunit detects an infected ZIP file through HTTPS GET in learning mode.

Sample log

```
1: date=2023-03-28 time=17:25:20 eventtime=1680049519342670162 tz="-0700" logid="0211008193"
type="utm" subtype="virus" eventtype="infected" level="notice" vd="vdom1" policyid=1
poluuid="4b848b70-cd95-51ed-c0ca-25e725a61062" policytype="security-policy"
policymode="learn" msg="File is infected." action="monitored" service="HTTPS" sessionid=5204
srcip=10.1.100.161 dstip=172.16.200.164 srcport=54106 dstport=443 srccountry="Reserved"
dstcountry="Reserved" srcintf="port2" srcintfrole="undefined" dstintf="port1"
dstintfrole="undefined" srcuuid="9bfd47cc-cd31-51ed-759b-ee3ad82f9d8c" dstuuid="9bfd47cc-
cd31-51ed-759b-ee3ad82f9d8c" proto=6 direction="incoming" filename="eicar.zip"
quarskip="Quarantine-disabled" virus="EICAR_TEST_FILE" viruscat="Virus" dtype="av-engine"
ref="http://www.fortinet.com/ve?vn=EICAR_TEST_FILE" virusid=2172
url="https://172.16.200.164/sample/eicar.zip" profile="learn-av" agent="curl/7.68.0"
httpmethod="GET"
analyticscksum="ed6ff9fb7388ccbd23e767ad38187e856f6810a1b74bb4945020a046e4ed9f09"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical" rawdata="Response-Content-
Type=application/zip"
```

Use case 2: DLP

In this example, the scanunit detects a *.doc file pattern filter through SMTP in learning mode.

Sample log

```
1: date=2023-03-28 time=17:22:55 eventtime=1680049375198948937 tz="-0700" logid="0954024577"
type="utm" subtype="dlp" eventtype="dlp" level="notice" vd="vdom1" filteridx=0
filtertype="none" filtercat="file" severity="info" policyid=1 poluuid="4b848b70-cd95-51ed-
c0ca-25e725a61062" policytype="security-policy" policymode="learn" sessionid=5157 epoch=0
eventid=0 srcip=10.1.100.161 srcport=57252 srccountry="Reserved" srcintf="port2"
srcintfrole="undefined" srcuuid="9bfd47cc-cd31-51ed-759b-ee3ad82f9d8c" dstip=172.16.200.164
dstport=25 dstcountry="Reserved" dstintf="port1" dstintfrole="undefined" dstuuid="9bfd47cc-
cd31-51ed-759b-ee3ad82f9d8c" proto=6 service="SMTP" filetype="msoffice" direction="outgoing"
action="log-only" filename="test.doc" profile="learn-dlp"
```

Use case 3: file filter

In this example, the scanunit detects an HTA file type through CIFS upload in learning mode.

Sample log

```
1: date=2023-03-28 time=17:20:45 eventtime=1680049244473571348 tz="-0700" logid="1900064001"
type="utm" subtype="file-filter" eventtype="file-filter" level="notice" vd="vdom1"
policyid=1 poluuid="4b848b70-cd95-51ed-c0ca-25e725a61062" policytype="security-policy"
policymode="learn" sessionid=5120 srcip=10.1.100.161 srcport=50706 srccountry="Reserved"
srcintf="port2" srcintfrole="undefined" srcuuid="9bfd47cc-cd31-51ed-759b-ee3ad82f9d8c"
dstip=172.16.200.164 dstport=445 dstcountry="Reserved" dstintf="port1"
dstintfrole="undefined" dstuuid="9bfd47cc-cd31-51ed-759b-ee3ad82f9d8c" proto=6 service="SMB"
profile="learn-filef" direction="outgoing" action="log-only" filename="upload\\hta_
sample.hta" filesize=290 filetype="hta" msg="File was detected by file filter."
```

Support dynamic Fabric address in security policies - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:
- Dynamic address tags

The FABRIC_DEVICE object, which is a dynamic address consisting of several types of Fabric devices (including FortiManager, FortiAnalyzer, FortiClient EMS, FortiMail, FortiAP, and FortiSwitch), can be used as the source or destination address in security policies. The `diagnose ips pme fabric-address list` command can be used to check what device address is set in the security policy after FABRIC_DEVICE is applied.

Tags for dynamic addresses, including EMS (normal and local EMS tags), FortiPolicy, FortiVoice, and FortiNAC can be used as the source or destination address in security policies. Once these tags are used in security policies, run `diagnose ips pme dynamic-address list` to show the addresses that are used in the policy.

Example 1: FABRIC_DEVICE object

To apply the FABRIC_DEVICE object to a security policy in the GUI:

1. Go to *Policy & Objects > Security Policy*.
2. Click *Create new* or edit an existing policy.
3. In the *Source* field, click the + and select *FABRIC_DEVICE*.
4. Configure the other settings as needed.
5. Click *OK*.

To apply the FABRIC_DEVICE object to a security policy in the CLI:

1. Configure the policy:

```
config firewall security-policy
    edit 1
        set name "ddd"
        set srcintf "port8"
        set dstintf "port7"
        set srcaddr "FABRIC_DEVICE"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set logtraffic all
    next
end
```

2. Verify which IP addresses are used in the policy:

```
# diagnose ips pme fabric-address list
VDOM 0:
- builtin [mask=0x1e]:
  - type=4: 172.18.62.213
  - type=4: 172.18.62.219
  - type=2: 172.18.70.82
- query:
  - 168.254.1.2
  - 0.0.0.0
  - 168.254.1.2
```

Example 2: EMS tag

To apply an EMS tag object to a security policy in the GUI:

1. Go to *Policy & Objects > Security Policy*.
2. Click *Create new* or edit an existing policy.
3. In the *Source* field, click the + and select *EMS1_ZTNA_ZT_OS_WIN*.
4. Configure the other settings as needed.
5. Click *OK*.

To apply an EMS tag object to a security policy in the CLI:

1. Configure the policy:

```
config firewall security-policy
    edit 1
        set name "ddd"
        set srcintf "port8"
        set dstintf "port7"
        set srcaddr "EMS1_ZTNA_ZT_OS_WIN"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set logtraffic all
    next
end
```

2. Verify which IP addresses are used in the policy:

```
# diagnose ips pme dynamic-address list
EMS1_ZTNA_ZT_OS_WIN [vdom=0 type=IP]:
  172.16.200.136-172.16.200.136
```

Policies

This section includes information about policy related new features:

Support destination port matching of central SNAT rules

This information is also available in the FortiOS 7.4 Administration Guide:

- Central SNAT

Central SNAT rules now include the destination port for traffic matching when the protocols are TCP, UDP, or SCTP. When configuring central SNAT rules in the CLI, the `set dst-port` command can be used to specify the destination port range.

Example

In the following example, two central SNAT rules will be created:

- Rule 3 will have a destination port set and IP pool `test-ippool4-3` applied.
- Rule 5 will have IP pool `test-ippool4-1` applied but will not set the destination port.

Example traffic will then be passed to see how the rule is matched.

To test central SNAT rule destination port support:

1. Configure central SNAT rule 3 with the destination port range specified:

```
config firewall ippool
    edit "test-ippool4-3"
        set startip 172.16.200.150
        set endip 172.16.200.150
    next
end
config firewall central-snat-map
    edit 3
        set srcintf "port24"
        set dstintf "port17"
        set orig-addr "all"
        set dst-addr "all"
        set protocol 6
        set nat-ippool "test-ippool4-3"
        set dst-port 80-443
    next
end
```

2. Configure central SNAT rule 5:

```
config firewall ippool
    edit "test-ippool4-1"
        set startip 172.16.200.151
        set endip 172.16.200.151
    next
end
config firewall central-snat-map
    edit 5
        set srcintf "port24"
        set dstintf "port17"
        set orig-addr "all"
        set dst-addr "all"
        set nat-ippool "test-ippool4-1"
    next
end
```

3. Send HTTP traffic to pass through the FortiGate that is expected to match central SNAT rule 3. IP pool `test-ippool4-3` will perform source NAT.
4. Check the session to review for expected behavior:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=2 expire=3599 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
```

```
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=1800/31/1 reply=77304/60/1 tuples=2
tx speed(Bps/kbps): 602/4 rx speed(Bps/kbps): 25854/206
orgin->sink: org pre->post, reply pre->post dev=24->17/17->24
gwy=172.16.200.55/10.1.100.42
hook=post dir=org act=snat 10.1.100.42:46731->172.16.200.55:80(172.16.200.150:46731)
hook=pre dir=reply act=dnat 172.16.200.55:80->172.16.200.150:46731(10.1.100.42:46731)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=99 pol_uuid_idx=15864 auth_info=0 chk_client_info=0 vd=0
serial=00003c37 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

5. Send PING traffic to pass through the FortiGate that is expected to match central SNAT rule 5. IP pool `test-ippool4-1` will perform source NAT.
6. Check the session to review for expected behavior:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=2 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=252/3/1 reply=252/3/1 tuples=2
tx speed(Bps/kbps): 99/0 rx speed(Bps/kbps): 99/0
orgin->sink: org pre->post, reply pre->post dev=24->17/17->24
gwy=172.16.200.55/10.1.100.42
hook=post dir=org act=snat 10.1.100.42:36732->172.16.200.55:8(172.16.200.151:36732)
hook=pre dir=reply act=dnat 172.16.200.55:36732->172.16.200.151:0(10.1.100.42:36732)
misc=0 policy_id=99 pol_uuid_idx=15864 auth_info=0 chk_client_info=0 vd=0
serial=00003f62 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

Support the Port Control Protocol

This information is also available in the FortiOS 7.4 Administration Guide:

- Configuring PCP port mapping with SNAT and DNAT

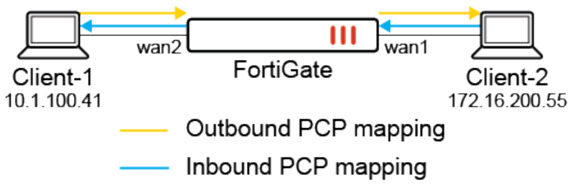
FortiOS supports the Port Control Protocol (PCP) by allowing the FortiGate to act as a PCP server, and dynamically manage network addresses and port translations for PCP clients. The PCP server must be enabled with a pool (`config system pcp-server`). In the firewall policy, enable either `pcp-outbound` or `pcp-inbound` mode and assign the pool.

```
config system pcp-server
    set status {enable | disable}
    config pools
        edit <name>
            set client-subnet <ip_address/subnet>
            set ext-intf <string>
            set extip ip>[-<ip>]
            set extport <port>[-<port>]
            set minimal-lifetime <integer>
            set maximal-lifetime <integer>
            set client-mapping-limit <integer>
            set mapping-filter-limit <integer>
            set allow-opcode {map peer announce}
            set third-party {allow | disallow}
            set multicast-announcement {enable | disable}
            set announcement-count <integer>
            set intl-intf <string>
            set recycle-delay <integer>
        next
    end
end
```

`client-subnet <ip_address/subnet>`	Enter the IP address with subnet from which PCP requests are accepted.
`ext-intf <string>`	Enter the external interface name.
`extip <ip>[-<ip>]`	Enter the IP address or address range on the external interface to map to an address on the internal network.
`extport <port>[-<port>]`	Enter the incoming port number or port range to map to a port number on the internal network.
`minimal-lifetime <integer>`	Set the minimal lifetime of a PCP mapping, in seconds (60 - 300, default = 120).
`maximal-lifetime <integer>`	Set the maximal lifetime of a PCP mapping, in seconds (3600 - 604800, default = 86400).
`client-mapping-limit <integer>`	Mapping limit per client (0 - 65535, default = 0, 0 = unlimited).
`mapping-filter-limit <integer>`	Filter limit per mapping (0 - 5, default = 1).
`allow-opcode {map peer announce}`	Set the allowed PCP OpCode: • `map`: allow MAP OpCode • `peer`: allow PEER OpCode • `announce`: allow ANNOUNCE OpCode
`third-party {allow | disallow}`	Allow/disallow the third-party option.
`multicast-announcement {enable | disable}`	Enable/disable multicast announcements.
`announcement-count <integer>`	Set the number of multicast announcements (3 - 10, default = 3).

`intl-intf <string>`	Enter the internal interface name.
`recycle-delay <integer>`	Set the minimum delay the PCP server will wait before recycling mappings that have expired, in seconds (0 - 3600, default = 0).

The following topology is used to demonstrate two use cases of PCP mapping: with SNAT and DNAT.



Example 1: PCP mapping with SNAT

This example demonstrates how PCP mapping works with SNAT. In the FortiGate PCP server settings, the pcp-pool1 pool is applied in the firewall policy with `pcp-outbound` mode. A PCP request is sent from Client-1 to the FortiGate to create PCP outbound mapping. When traffic is sent from Client-1 to Client-2, SNAT is performed by the PCP outbound mapping.

To configure the FortiGate as a PCP server:

1. Configure the PCP server settings:

```
config system pcp-server
    set status enable
    config pools
        edit "pcp-pool1"
            set client-subnet "10.1.100.41/32"
            set ext-intf "wan1"
            set extip 172.16.200.231
            set extport 50000-51000
            set intl-intf "wan2"
        next
    end
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 999
        set name "Outbound-pcp-policy999"
        set srcintf "wan2"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
```

```
        set pcp-outbound enable
        set pcp-poolname "pcp-pool1"
    next
end
```

To verify the configuration:

1. Generate a PCP peer request from Client-1 (10.1.100.41) to the FortiGate.
2. Verify the client's PCP request to the PCP server. In this example, an PCP client was installed on Ubuntu:
   ```
   root@pc41:~# pcp -i 10.1.100.41:41111 -p 172.16.200.55:80 -s 10.1.100.8
   ```
3. On the FortiGate, verify the PCP outbound mappings list:
   ```
   # diagnose firewall pcp-mapping list outbound
   PCP outbound mappings (vdom=root):
   pool:1 nonce:04307eb4037e0448317dc8b7 protocol:6 duration:8 lifetime:900 expiry:893
   intl:10.1.100.41:41111 ext:172.16.200.231:50000 remote:172.16.200.55:80
   ```
4. Send HTTP traffic that passes through the FortiGate and access Client-2 (172.16.200.55:80) from Client-1.
5. On the FortiGate, verify the session list. The source IP address of Client-1 is translated to 172.16.200.231:50000, which follows the PCP outbound mapping:
   ```
   # diagnose sys session list
   session info: proto=6 proto_state=01 duration=8 expire=3599 timeout=3600 flags=00000000
   socktype=0 sockport=0 av_idx=0 use=3
   origin-shaper=
   reply-shaper=
   per_ip_shaper=
   class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
   state=log may_dirty f00 pcp_outbound
   statistic(bytes/packets/allow_err): org=1812/33/1 reply=124168/92/1 tuples=2
   tx speed(Bps/kbps): 204/1 rx speed(Bps/kbps): 13998/111
   orgin->sink: org pre->post, reply pre->post dev=8->7/7->8 gwy=172.16.200.55/10.1.100.41
   hook=post dir=org act=snat 10.1.100.41:41111->172.16.200.55:80(172.16.200.231:50000)
   hook=pre dir=reply act=dnat 172.16.200.55:80->172.16.200.231:50000(10.1.100.41:41111)
   pos/(before,after) 0/(0,0), 0/(0,0)
   misc=0 policy_id=999 pol_uuid_idx=677 auth_info=0 chk_client_info=0 vd=0
   serial=0000b4f8 tos=ff/ff app_list=0 app=0 url_cat=0
   rpdb_link_id=00000000 ngfwid=n/a
   npu_state=0x4000001 no_offload
   no_ofld_reason:  disabled-by-policy
   total session 1
   ```
6. Send HTTP traffic that passes through the FortiGate and access another server from Client-1.
7. On the FortiGate, verify the session list. This time, the source IP address of Client-1 is not translated to 172.16.200.231:50000, since the traffic does not match the existing PCP outbound mapping:
   ```
   # diagnose sys session list
   session info: proto=6 proto_state=01 duration=6 expire=3596 timeout=3600 flags=00000000
   socktype=0 sockport=0 av_idx=0 use=3
   origin-shaper=
   reply-shaper=
   per_ip_shaper=
   class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
   state=log may_dirty f00
   statistic(bytes/packets/allow_err): org=1449/26/1 reply=98808/72/1 tuples=2
   ```

```
tx speed(Bps/kbps): 215/1 rx speed(Bps/kbps): 14703/117
orgin->sink: org pre->post, reply pre->post dev=8->7/7->8 gwy=172.16.200.155/10.1.100.41
hook=post dir=org act=snat 10.1.100.41:41111->172.16.200.155:80(172.16.200.8:41111)
hook=pre dir=reply act=dnat 172.16.200.155:80->172.16.200.8:41111(10.1.100.41:41111)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=999 pol_uuid_idx=677 auth_info=0 chk_client_info=0 vd=0
serial=0000b596 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

Example 2: PCP mapping with DNAT

This example demonstrates how PCP mapping works with DNAT. In the FortiGate PCP server settings, the pcp-pool1 pool is applied in the firewall policy with `pcp-inbound` mode. A PCP request is sent from Client-1 to the FortiGate to create PCP inbound mapping. When traffic is sent from Client-2 to access the external IP of Client-1 (172.16.200.231:50000), traffic passes by due to the PCP inbound mapping.

To configure the FortiGate as a PCP server:

1. Configure the PCP server settings:

```
config system pcp-server
    set status enable
    config pools
        edit "pcp-pool1"
            set client-subnet "10.1.100.41/32"
            set ext-intf "wan1"
            set extip 172.16.200.231
            set extport 50000-51000
            set intl-intf "wan2"
        next
    end
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 998
        set name "Inbound-pcp-policy998"
        set srcintf "wan1"
        set dstintf "wan2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
        set pcp-inbound enable
        set pcp-poolname "pcp-pool1"
```

```
    next
end
```

To verify the configuration:

1. Generate a PCP peer request from Client-1 (10.1.100.41) to the FortiGate.
2. Verify the client's PCP request to the PCP server. In this example, an PCP client was installed on Ubuntu:

```
root@pc41:~# pcp -i 10.1.100.41:80 -s 10.1.100.8
```

3. On the FortiGate, verify the PCP inbound mappings list:

```
# diagnose firewall pcp-mapping list inbound
PCP inbound mappings (vdom=root):
pool:1 nonce:35e2ff035b959f7a4e669791 protocol:6 duration:3 lifetime:900 expiry:900
intl:10.1.100.41:80 ext:172.16.200.231:50000
```

4. From Client-2 (172.16.200.55:80), send traffic that passes through the FortiGate and access the external IP of Client-1 (172.16.200.231:50000).
5. On the FortiGate, run a sniffer trace. The traffic is allowed through policy 998, and the destination IP:port is translated from 172.16.200.231:50000 to 10.1.100.41:80, which follows the PCP inbound mapping:

```
# diagnose sniffer packet any 'tcp and port 50000 or port 80' 4
interfaces=[any]
filters=[tcp and port 50000 or port 80]
2.959915 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: syn 3480016601
2.960051 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: syn 3480016601
2.960390 wan2 in 10.1.100.41.80 -> 10.1.100.8.43284: syn 2813145613 ack 3480016602
2.960447 wan1 out 172.16.200.231.50000 -> 172.16.200.55.43284: syn 2813145613 ack
3480016602
2.960644 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: ack 2813145614
2.960664 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: ack 2813145614
2.961194 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: psh 3480016602 ack
2813145614
2.961209 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: psh 3480016602 ack 2813145614
2.961516 wan2 in 10.1.100.41.80 -> 10.1.100.8.43284: ack 3480016686
2.961533 wan1 out 172.16.200.231.50000 -> 172.16.200.55.43284: ack 3480016686
2.993623 wan2 in 10.1.100.41.80 -> 10.1.100.8.43284: psh 2813145614 ack 3480016686
2.993637 wan1 out 172.16.200.231.50000 -> 172.16.200.55.43284: psh 2813145614 ack
3480016686
2.993947 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: ack 2813145875
2.993962 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: ack 2813145875
2.995677 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: fin 3480016686 ack
2813145875
2.995691 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: fin 3480016686 ack 2813145875
2.996059 wan2 in 10.1.100.41.80 -> 10.1.100.8.43284: fin 2813145875 ack 3480016687
2.996075 wan1 out 172.16.200.231.50000 -> 172.16.200.55.43284: fin 2813145875 ack
3480016687
2.996230 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: ack 2813145876
2.996245 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: ack 2813145876
```

Only traffic matching the PCP inbound mapping will be forwarded by policy 998. Any other traffic is dropped.

Improve the performance of the GUI policy list

This information is also available in the FortiOS 7.4 Administration Guide:

- Policy views

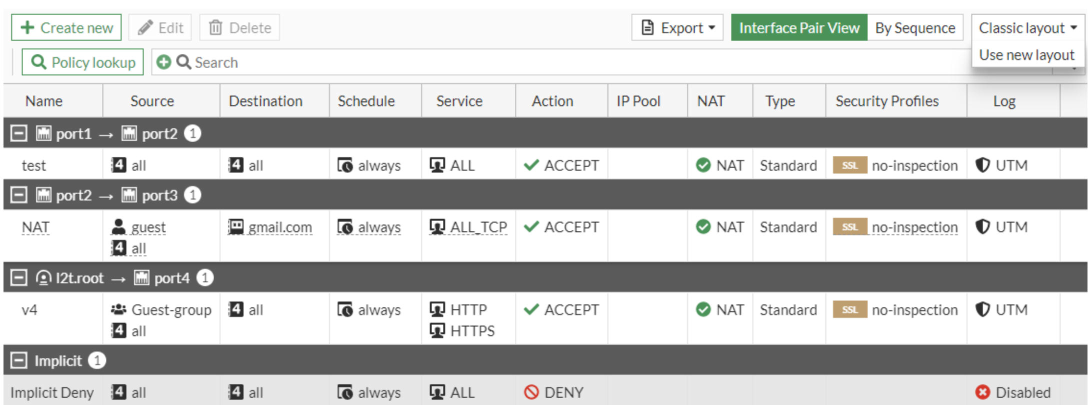
Improvements to the FortiOS GUI backend have been implemented to speed up the loading of a large number of policies. This is achieved by only loading the necessary data when needed, rather than loading all the data at once. This can significantly improve performance and reduce the time it takes to load a large number of policies.

A new layout has also been introduced for the policy list with the option to choose between the new layout and the old layout. To switch between the classic and new policy list layout, select the style from the dropdown menu.

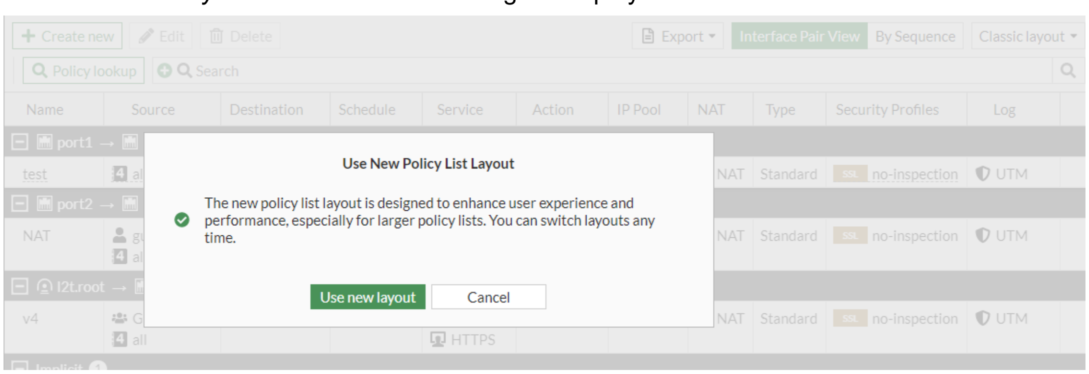
In addition, the *Interface Pair View* is now available when a policy is configured with multiple interfaces. Previously the *Interface Pair View* was grayed out when multiple interfaces were set for a policy, and the *By Sequence* view was displayed.

To change from the classic layout to the new layout:

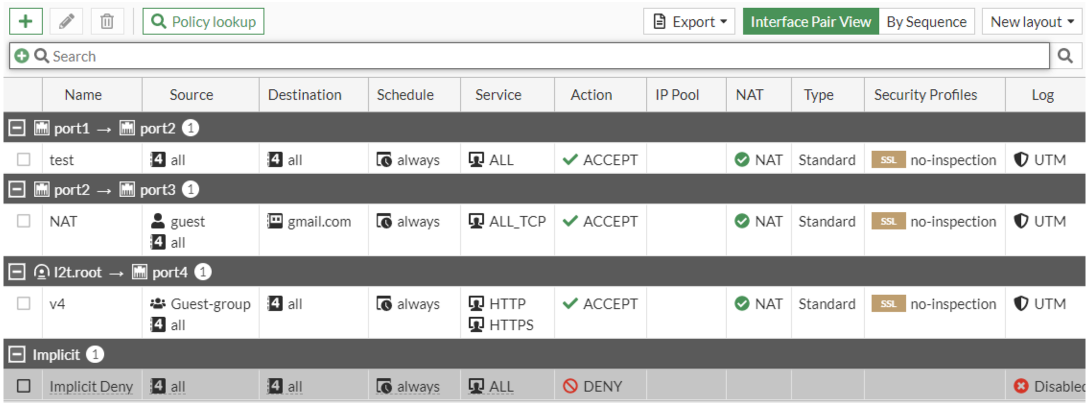
1. Go to *Policy & Objects > Firewall Policy*.
2. Select the *Classic layout* dropdown menu.



3. Select *Use new layout*. A confirmation message is displayed.

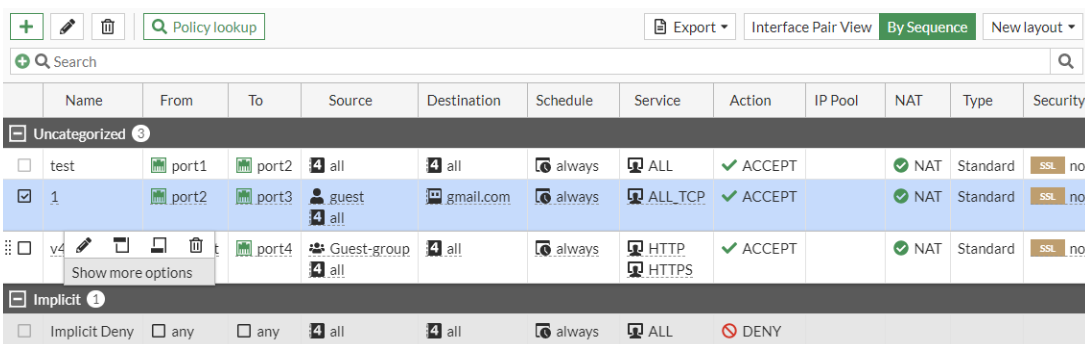


4. Click *Use new layout*. The new layout is displayed.



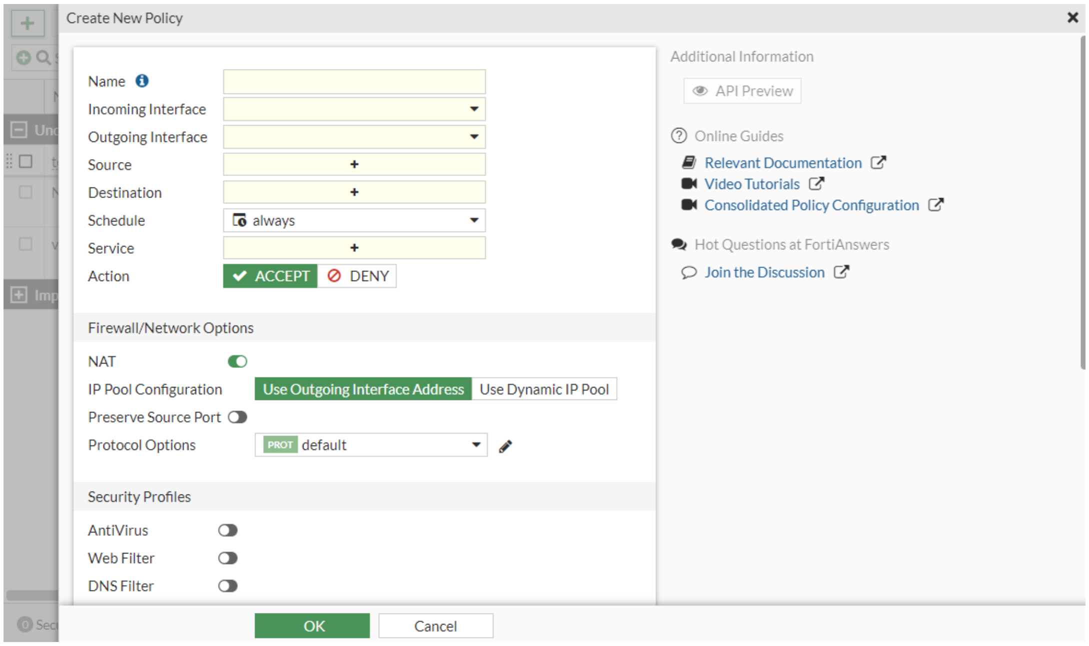
The new layout includes several features to enhance user experience when using the *Policy & Objects > Firewall Policy* page:

- The create, edit, and delete buttons are identified through icons instead of words. Selecting a policy also displays an inline menu with options to edit, delete, and insert policies, with the option to *Show more options* when hovered over.

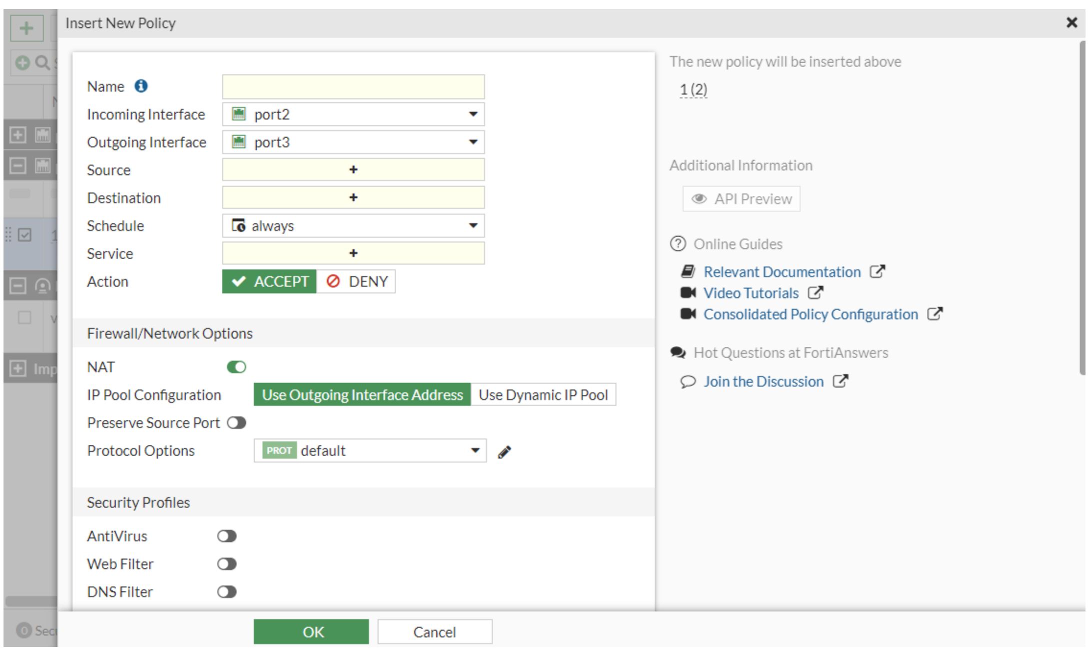


- Right-click in *Interface Pair View* to *Expand All* or *Collapse All* sections.

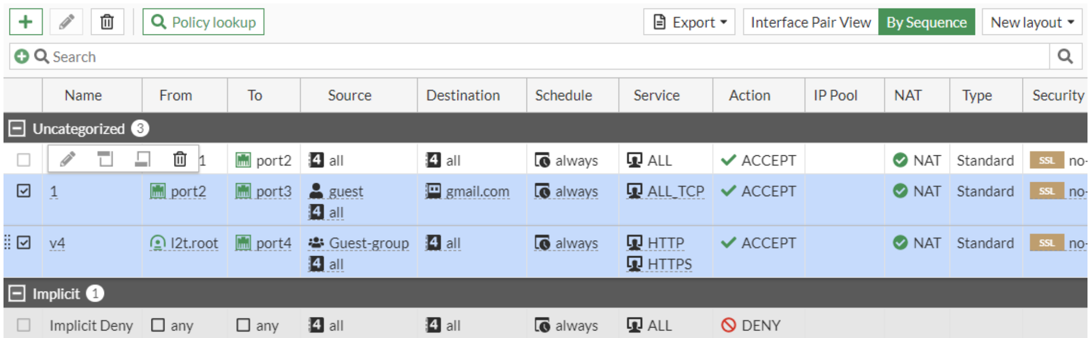
- A pane is used to create, edit, and insert policies instead of a separate page.



- When a policy is inserted in *Interface Pair View*, the *Incoming Interface* and *Destination Interface* fields will be automatically filled. You can confirm the location of the new policy in the right-side gutter before clicking *OK* to insert the policy.



- Multiple policies can be selected at once to efficiently work with a large number of policies.



Process Ethernet frames with Cisco Security Group Tag and VLAN tag

This information is also available in the FortiOS 7.4 Administration Guide:

- Processing Ethernet frames with a Cisco Security Group Tag and VLAN tag

The FortiGate has the ability to process Ethernet frames with both the Cisco Security Group Tag and VLAN tag.

The FortiGate can read the Cisco Security Group Tag (SGT) in Ethernet frames with wildcard VLANs, and use them as matching criteria in firewall policies. A policy can match based on the presence of an SGT with wildcard VLAN, or the detection of a specific ID or IDs.

When a packet with an SGT passes through and a session is established, the `ext_header_type=0xc5:0xc5` flag is included in the session table.

This feature is available in flow mode policies for virtual wire pair policies or policies in transparent mode VDOMs.

Example

In this example, wan1 and wan2 are in a virtual wire pair. An Ethernet frame is sent from PC01 with an SGT tag (ID 20) and VLAN ID (2), which can pass through to PC05 based on the firewall policy because `sgt-check` is enabled, and `sgt` is set to `20`.

To configure the FortiGate:

1. Configure the virtual wire pair:

```
config system virtual-wire-pair
    edit "test-vwp-1"
        set member "wan2" "wan1"
        set wildcard-vlan enable
```

```
    next
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "wan2"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set auto-asic-offload disable
        set sgt-check enable
        set sgt 20
    next
end
```

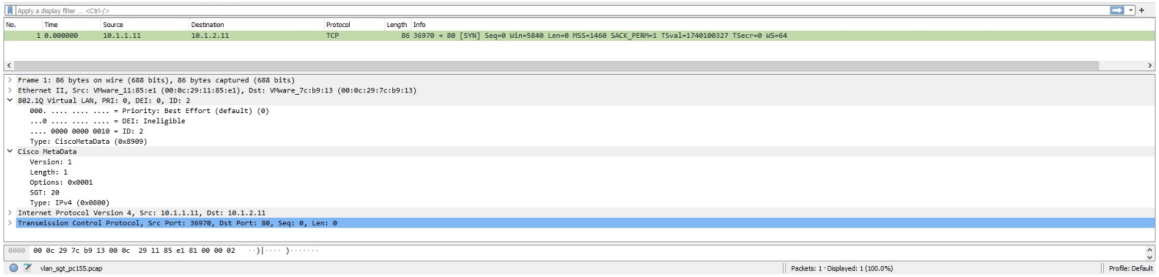
To verify the configuration:

1. Check the session list:

```
# diagnose sys session list

session info: proto=6 proto_state=01 duration=2007 expire=3482 timeout=3600
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=may_dirty br
statistic(bytes/packets/allow_err): org=164/3/1 reply=120/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=8->7/7->8 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.1.1.11:36970->10.1.2.11:80(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.2.11:80->10.1.1.11:36970(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=572 auth_info=0 chk_client_info=0 vd=0
serial=0432fb8f tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
vlanid=2
npu_state=0x4000001 no_offload
no_ofld_reason:  disabled-by-policy
           ext_header_type=0xc5:0xc5
```

2. Perform a packet capture on PC05 (Wireshark is used in this example) and check that the packet includes the

VLAN ID and Cisco SGT fields.



Support port block allocation for NAT64

This information is also available in the FortiOS 7.4 Administration Guide:

- Port block allocation with NAT64

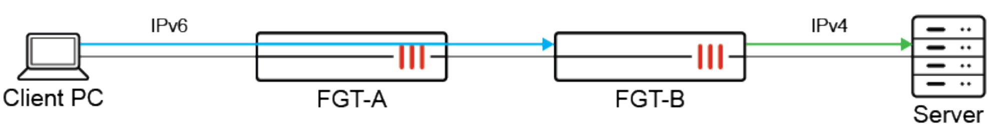
Port block allocation (PBA) support for NAT64 is supported for FortiGates with a hyperscale firewall license. This feature has been added to mainstream FortiOS to make it available to non-hyperscale customers, including customers running a VM version of FortiOS. Hyperscale firewall logging is designed for optimal performance and does not have the same detailed logging features as are available for non-hyperscale traffic.

```
config firewall ippool
    edit <name>
        set type port-block-allocation
        set nat64 enable
    next
end
```

In FortiOS 7.4.2 and later, the *IP Pool* dialog page (*Policy & Objects* > *IP Pools*) includes an option to enable *NAT64* in PBA configurations.

Example

In this example, a NAT64 virtual IPv6 address and PBA IP pool are configured on FGT-B. IPv6 traffic from the client PC is able to access the IPv4 server.



The IPv6 addresses used in this example are for demonstrative purposes only and should not be used in your environment.

The 2001:db8::/32 prefix is a special IPv6 prefix designated for use in documentation examples. See RFC 3849 for more information.

To configure PBA for NAT64 on FGT-B:

1. Configure the IP pools and enable NAT 64:

```
(vdom1) config firewall ippool
    edit "ippool4-1072390-1"
        set type port-block-allocation
        set startip 172.16.164.164
        set endip 172.16.164.164
        set block-size 64
        set num-blocks-per-user 1
        set pba-timeout 60
        set nat64 enable
    next
    edit "ippool4-1072390-2"
        set type port-block-allocation
        set startip 172.16.164.165
        set endip 172.16.164.165
        set block-size 64
        set num-blocks-per-user 1
        set pba-timeout 60
        set nat64 enable
    next
end
```

2. Configure the virtual IP for IPv6:

```
(vdom1) config firewall vip6
    edit "vip64-1072390"
        set extip 64:ff9b::-64:ff9b::ffff:ffff
        set nat66 disable
        set nat64 enable
        set embedded-ipv4-address enable
    next
end
```

3. Configure the firewall policy:

```
(vdom1) config firewall policy
    edit 1072390
        set srcintf "port7"
        set dstintf "port1"
        set action accept
        set nat64 enable
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "vip64-1072390"
        set schedule "always"
        set service "ALL"
        set auto-asic-offload disable
        set ippool enable
        set poolname "ippool4-1072390-1" "ippool4-1072390-2"
    next
end
```

4. Send IPv6 packets from the client to access the IPv4 server.
5. Verify the NAT64 sessions:

```
(vdom1) # diagnose sys session6 stat
misc info:  session_count=128 setup_rate=0 exp_count=0 reflect_count=0 clash=0
      memory_tension_drop=0 ephemeral=0/0 removeable=0 extreme_low_mem=0
      npu_session_count=0
      nturbo_session_count=0
delete=0, flush=3, dev_down=0/0 ses_walkers=0
```

There are 128 sessions allocated to the two PBA IP pools.

6. Verify the PBA IP pools status:

```
(vdom1) # diagnose firewall ippool list
list ippool info:(vf=vdom1)
ippool ippool4-1072390-1: id=1, block-sz=64, num-block=1, fixed-port=no, use=5
      nat ip-range=172.16.164.164-172.16.164.164 start-port=5117, num-pba-per-ip=944
      clients=2, inuse-NAT-IPs=1
      total-PBAs=944, inuse-PBAs=1, expiring-PBAs=1, free-PBAs=99.89%
      allocate-PBA-times=2, reuse-PBA-times=0
ippool ippool4-1072390-2: id=2, block-sz=64, num-block=1, fixed-port=no, use=4
      nat ip-range=172.16.164.165-172.16.164.165 start-port=5117, num-pba-per-ip=944
      clients=1, inuse-NAT-IPs=1
      total-PBAs=944, inuse-PBAs=1, expiring-PBAs=0, free-PBAs=99.89%
      allocate-PBA-times=1, reuse-PBA-times=0
```

Each IP pool uses one IPv4 address and one block (64 ports) for SNAT.

7. Verify the PBAs in the IP pools in the current VDOM:

```
(vdom1) # diagnose firewall ippool list pba
user 2001:db8:d0c:1::1, 172.16.164.164, 5181-5244, idx=1, use=66
user 2001:db8:d0c:1::1, 172.16.164.165, 5117-5180, idx=0, use=66
```

This output includes the client IP, NAT IP, NAT port range, port block index, and a kernel reference counter.

8. Verify the NAT IPs in use in the current VDOM:

```
(vdom1) # diagnose firewall ippool list nat-ip
NAT-IP 172.16.164.164, pba=1, use=3
NAT-IP 172.16.164.165, pba=1, use=3
```

This output includes the number of PBAs allocated for the NAT IP and the number of PBAs in use.

9. Verify the number of PBAs assigned to the user IP and the number of PBAs being used:

```
(vdom1) # diagnose firewall ippool list user
User-IP 2001:db8:d0c:1::1, pba=1, use=3
User-IP 2001:db8:d0c:1::1, pba=1, use=3
```

Support refreshing active sessions for specific protocols and port ranges per VDOM in a specified direction - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Refreshing active sessions for specific protocols and port ranges per VDOM in a specified direction

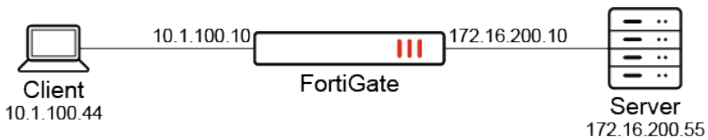
Active sessions can be refreshed for specific protocols and port ranges per VDOM in a specified direction. This option can help prevent potential denial of service (DoS) attacks by controlling the direction of traffic that refreshes existing sessions.

```
config system session-ttl
    config port
        edit <id>
            set protocol <integer>
            set timeout <timeout_value>
            set refresh-direction {both | outgoing | incoming}
        next
    end
end
```

Setting the `refresh-direction` to `outgoing` will use the original direction, while `incoming` will use the reply direction. To refresh in both directions, select `both`.

Example

In this example, active sessions for UDP port 5001 will be refreshed in the incoming direction.



To refresh active sessions for UDP port 5001 in the incoming direction:

1. Configure the global session TTL timer:

```
config system session-ttl
    set default 3600
    config port
        edit 5001
            set protocol 17
            set timeout 5001
            set refresh-direction incoming
            set start-port 5001
            set end-port 5001
        next
    end
end
```

2. Send UDP 5001 traffic from the client to the server.
3. Verify the session table:

```
# diagnose sys session list
session info: proto=17 proto_state=00 duration=77 expire=4923 timeout=5001 refresh_
dir=reply flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=58/2/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=18->17/17->18 gwy=172.16.200.55/0.0.0.0
hook=post dir=org act=snat 10.1.100.41:2041->172.16.200.55:5001(172.16.200.10:62458)
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.10:62458(10.1.100.41:2041)
```

```
src_mac=00:0c:29:b6:e8:be  dst_mac=00:0c:29:92:89:96
misc=0 policy_id=99 pol_uuid_idx=1501 auth_info=0 chk_client_info=0 vd=0
serial=00005071 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

The timeout and refresh for the reply direction are attached to the session.

4. Send UDP 5001 traffic again from the client to the server.
5. Verify the diagnostics.
   a. Run the sniffer trace:

```
# diagnose sniffer packet any 'udp and port 5001' 4
interfaces=[any]
filters=[udp and port 5001]
3.387747 wan2 in 10.1.100.41.2041 -> 172.16.200.55.5001: udp 1
3.387757 wan1 out 172.16.200.10.62458 -> 172.16.200.55.5001: udp 1
^C
2 packets received by filter
0 packets dropped by kernel
```

   b. Verify the session table:

```
# diagnose sys session list
session info: proto=17 proto_state=00 duration=119 expire=4881 timeout=5001 refresh_
dir=reply flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=116/4/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 1/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=18->17/17->18
gwy=172.16.200.55/0.0.0.0
hook=post dir=org act=snat 10.1.100.41:2041->172.16.200.55:5001(172.16.200.10:62458)
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.10:62458(10.1.100.41:2041)
src_mac=00:0c:29:b6:e8:be  dst_mac=00:0c:29:92:89:96
misc=0 policy_id=99 pol_uuid_idx=1501 auth_info=0 chk_client_info=0 vd=0
serial=00005071 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

As the traffic flows from the client to the server (outgoing), the expiration timer continues to count down and is not refreshed.

6. Send reverse UDP 5001 traffic from the server to the client.
7. Verify the diagnostics again.
   a. Run the sniffer trace:

```
# diagnose sniffer packet any 'udp and port 62458 or port 2041' 4
interfaces=[any]
filters=[udp and port 62458 or port 2041]
3.237328 wan1 in 172.16.200.55.5001 -> 172.16.200.10.62458: udp 1
```

```
3.237339 wan2 out 172.16.200.55.5001 -> 10.1.100.41.2041: udp 1
^C
2 packets received by filter
0 packets dropped by kernel
```

b. Verify the session table:

```
# diagnose sys session list
session info: proto=17 proto_state=01 duration=1710 expire=4995 timeout=5001 refresh_
dir=reply flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=116/4/1 reply=116/4/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=18->17/17->18
gwy=172.16.200.55/10.1.100.41
hook=post dir=org act=snat 10.1.100.41:2041->172.16.200.55:5001(172.16.200.10:62458)
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.10:62458(10.1.100.41:2041)
src_mac=00:0c:29:b6:e8:be  dst_mac=00:0c:29:92:89:96
misc=0 policy_id=99 pol_uuid_idx=1501 auth_info=0 chk_client_info=0 vd=0
serial=00005071 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

As the traffic flows from the server to the client (incoming), the expiration timer is refreshed.

Update policy lookup tool with policy match tool - 7.4.1

The enhanced *Policy match* tool retains all the functionality of its predecessor (*Policy lookup*) and adds the ability to return a new policy match results page based on the provided parameters. Policy match results now include web filter profile information (if a web filter is applied) and the ability to use identity-based policy matching. From the *Matched Policy* section in the match results, administrators can redirect to the policy list or edit the policy. The gutter area in the *Policy Match Tool* pane displays the top 10 recent matches. This feature provides a more comprehensive and user-friendly way to diagnose and manage policies.

The `diagnose firewall iprope lookup` command has been updated to specify additional parameters, including policy type (policy or proxy), and a new parameter for identity-based policy matching. The policy match feature will be activated if more than six parameters are specified in the existing diagnose command.

```
# diagnose firewall iprope lookup <source_ip> <source_port> <destination_ip> <destination_
port> <protocol> <device> <policy_type> [<auth_type>] [<user/group>] [<server>]
```

On entry-level FortiGates, the *Policy lookup* tool is renamed to *Policy match*. The web filter action tracing and user matching functionalities are not available, and `diagnose firewall iprope lookup` can only be used for basic policy lookups.

Example

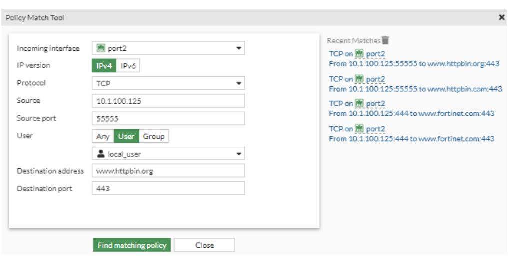
In this example, a local user (local_user) belongs to the local_group user group (see User definition and groups for more information). A web filter profile (WF) is configured where category 52 (Information Technology) is blocked (see FortiGuard filter for more information). A firewall policy (WF) is configured where the local_group is used as a source, and the WF web filter is applied (see Firewall policy for more information).

The administrator uses the *Policy match* tool to search for matches based on a URL belonging to Information Technology category and the local user.

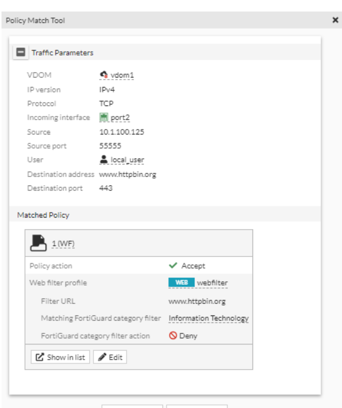
To use the policy match tool in the GUI:

1. Go to *Policy & Objects > Firewall Policy* and click *Policy match*. The *Policy Match Tool* pane opens.
2. Enter the search parameters:

Incoming interface	*port2*
Protocol	*TCP*
Source	*10.1.100.125*
Source port	*55555*
User	Select *User* and choose the *local_user*.
Destination address	*www.httpbin.org*
Destination port	*443*



3. Click *Find Matching policy*.
4. The results are returned.



There is one policy match, and the corresponding web filter profile information is included in the results (*Deny* action).

5. Optionally, click *Show in list* or *Edit* to view or edit the policy respectively.

To use the policy match tool in the CLI:

```
# diagnose firewall iprope lookup 10.1.100.125 55555 www.httpbin.org 443 6 port2 policy user
local_user
firewall policy id: 1
firewall proxy-policy id: 0
matched policy_type: policy
policy_action: accept
webf_profile: webfilter
webf_action: deny
webf_cate: 52
urlf_entry_id: 0
```

To perform the REST API request:

1. Open the web browser.
2. In the address bar, enter `https://172.18.200.63:443/api/v2/monitor/firewall/policy-lookup/?access_token=<token>&srcintf=port2&sourceport=55555&sourceip=10.1.100.125&protocol=6&dest=www.httpbin.org&destport=443&policy_type=policy&auth_type=user&user_group=local_user&ipv6=false&vdom=vdom1`.
3. The browser displays the output similar to the following:

```
{
  "http_method":"GET",
  "results":{
    "match":true,
       "policy_id":1,
       "matched_policy_type":"policy",
       "srcaddr":"",
       "dstaddr":"",
       "user_group":"",
       "webfilter_profile":"webfilter",
       "webfilter_action":"deny",
       "webfilter_category":52,
       "urlf_entry_id":0,
       "success":true
  },
  "vdom":"vdom1",
  "path":"firewall",
  "name":"policy-lookup",
  "sction":"",
  "status":"success",
  "serial":"FG10E1TB20900000",
  "version":"v7.4.1",
       "build":"2463",
}
```

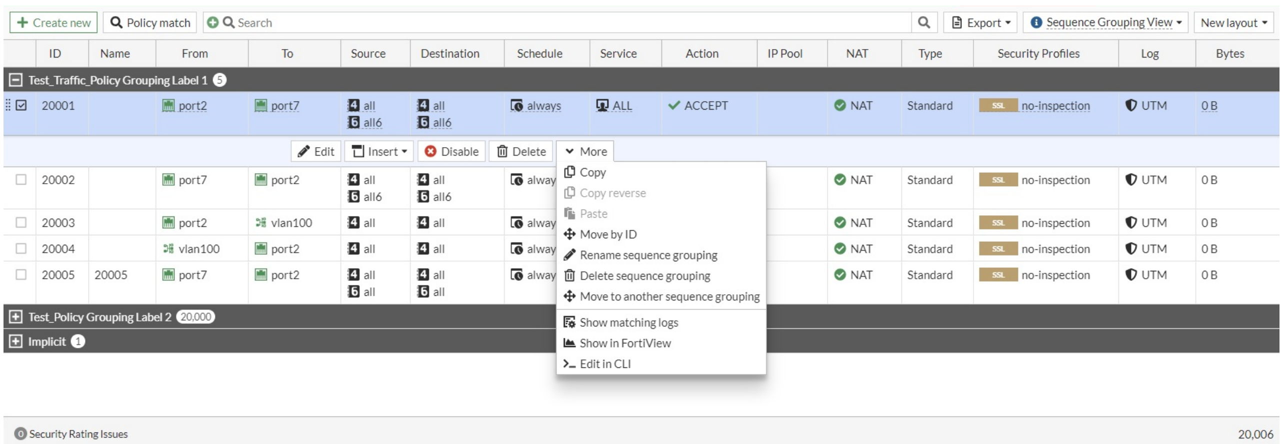
Policy list enhancements - 7.4.1

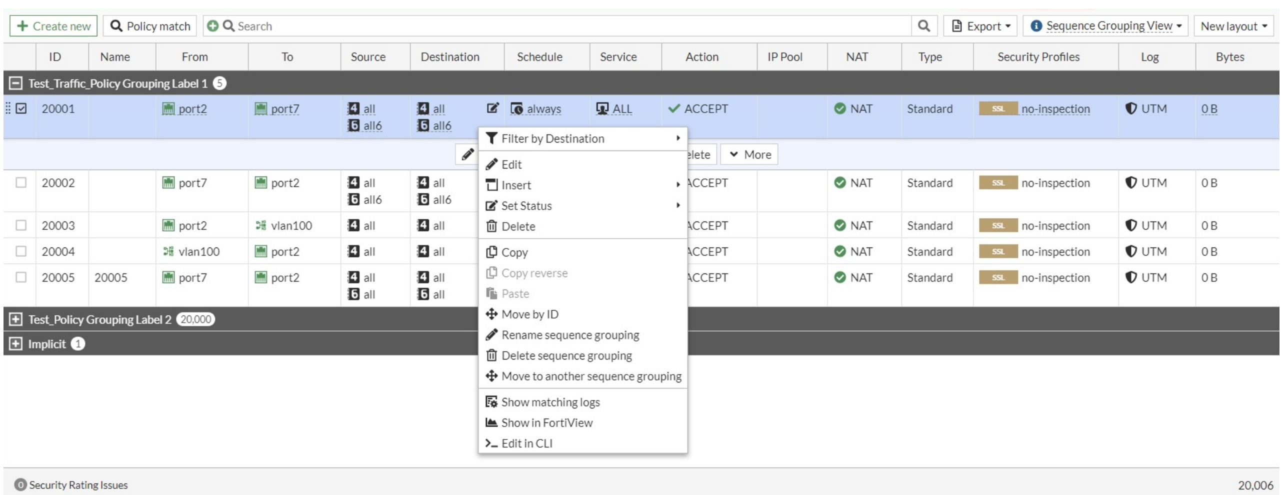
This information is also available in the FortiOS 7.4 Administration Guide:

- Firewall policy

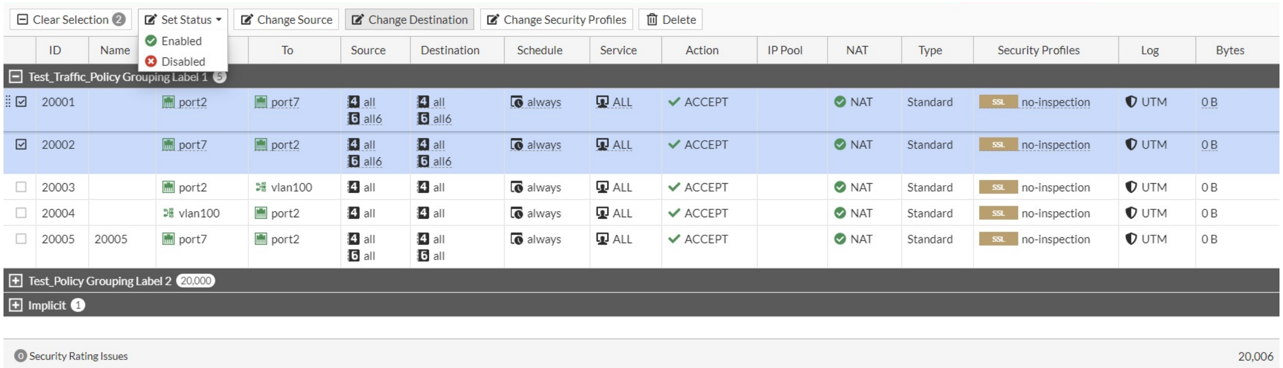
The policy list has been enhanced.

- When a single policy is selected, an inline menu opens below the row. The *More* dropdown menu includes the same expanded list of options that are available in the right-click menu.

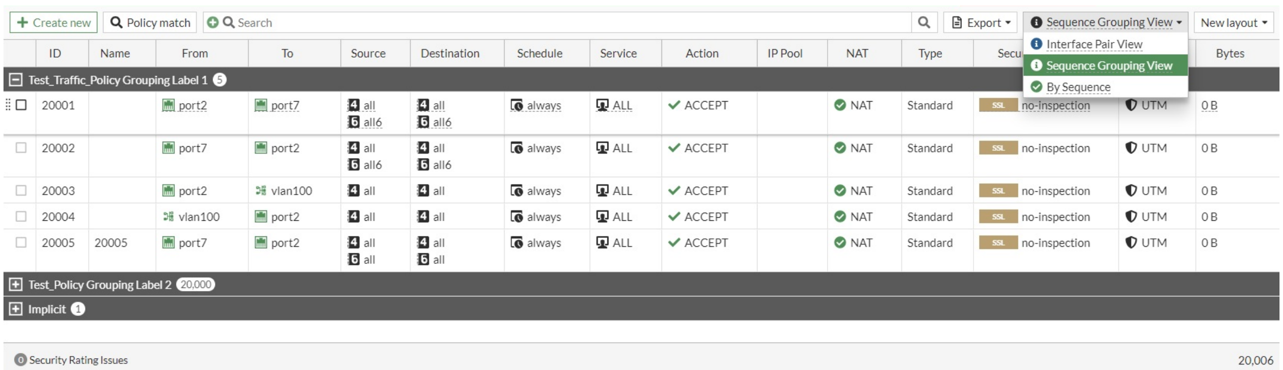




- When multiple policies are selected, the top menu bar changes to show buttons that are applicable to the multiple selections.



- The view selector drop-down includes three options: *Interface Pair View*, *Sequence Grouping View*, and *By Sequence*. For large policy tables (thousands of policies), a tooltip will specify that the *By Sequence* view will load the fastest.



Support IPS inspection for multicast UDP traffic - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

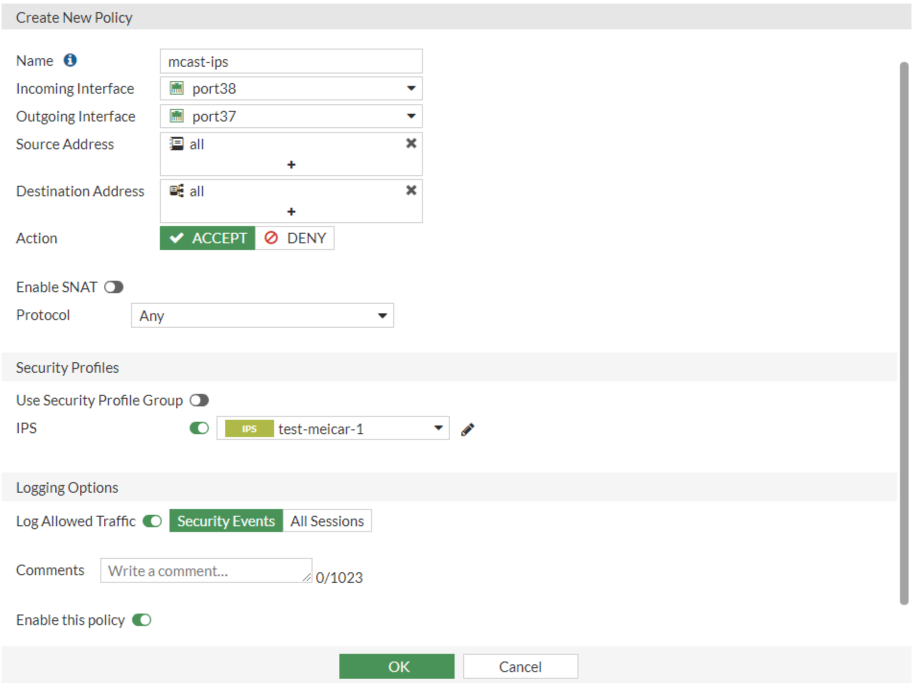
- Using IPS inspection for multicast UDP traffic

IPS inspection can be applied for multicast UDP traffic in multicast firewall policies.

```
config firewall {multicast-policy | multicast-policy6}
    edit <id>
        set utm-status {enable | disable}
        set ips-sensor <name>
        set logtraffic {all | utm | disable}
    next
end
```

IPv4 and IPv6 multicast policies can be configured in the GUI. Go to *System* > *Feature Visibility*, and enable *Multicast Policy* and *IPv6*.

The multicast policy dialog page (*Policy & Objects* > *Multicast Policy*) includes a *Security Profiles* section where you can enable *IPS* and apply an IPS profile.



Example

In this example, an IPv4 multicast policy is configured with IPS inspection enabled. Multicast UDP traffic that contains IPS attacks is detected and blocked. A custom IPS signature is created with an infected EICAR pattern for the UDP protocol.

To use IPS inspection for multicast UDP traffic:

1. Configure the IPS custom signature:

```
config ips custom
    edit "meicar"
        set signature "F-SBID( --name \"meicar\"; --attack_id 9999; --protocol udp; --
severity medium; --default_action clear_session; --pattern \"$EICAR-STANDARD-ANTIVIRUS-
TEST-FILE\";)"
        set protocol UDP
        set log disable
        set action block
    next
end
```

2. Configure the IPS sensor:

```
config ips sensor
    edit "test-meicar-1"
        config entries
            edit 1
                set rule 9999
                set status enable
                set action block
            next
```

```
        end
    next
end
```

3. Configure the multicast policy:

```
config firewall multicast-policy
    edit 1
        set srcintf "port38"
        set dstintf "port37"
        set srcaddr "all"
        set dstaddr "all"
        set utm-status enable
        set ips-sensor "test-meicar-1"
    next
end
```

4. Add the server to the multicast group 239.1.1.10 and join it using a terminal:

```
fosqa@ips_pc5:~$ iperf -s -u -B 239.1.1.10 -i 1
------------------------------------------------------------
Server listening on UDP port 5001
Binding to local address 239.1.1.10
Joining multicast group  239.1.1.10
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 239.1.1.10 port 5001 connected with 10.1.100.11 port 52972
```

5. From a terminal on the client, send multicast UDP traffic with the EICAR file:

```
root@PC01:~# iperf -c 239.1.1.10 -u -T 3 -t 20 -i 1 -F eicar
------------------------------------------------------------
Client connecting to 239.1.1.10, UDP port 5001
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
Setting multicast TTL to 3
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  4] local 10.1.100.11 port 33383 connected with 239.1.1.10 port 5001
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0- 0.0 sec  1.44 KBytes  1.03 Mbits/sec
[  4] Sent 1 datagrams
```

The traffic will be blocked, and the server will not be able to receive the packets.

6. Verify that the traffic is blocked.

   a. Verify the IPS event log:

```
# execute log filter category 4
# execute log display
1 logs found.
1 logs returned.

1: date=2023-11-01 time=17:01:43 eventtime=1698883303178500916 tz="-0700"
logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert"
vd="vd1" severity="medium" srcip=10.1.100.11 srccountry="Reserved" dstip=239.1.1.10
dstcountry="Reserved" srcintf="port38" srcintfrole="undefined" dstintf="port37"
dstintfrole="undefined" sessionid=18 action="dropped" proto=17 service="udp/5001"
policyid=1 poluuid="09bdd086-78e2-51ee-1d61-0955f9046b53" policytype="multicast-
```

```
policy" attack="meicar" srcport=52673 dstport=5001 direction="outgoing" attackid=9999
profile="test-meicar-1" incidentserialno=245366798 msg="custom: meicar" crscore=10
craction=16384 crlevel="medium"
```

b. Verify the IPS traffic log:

```
# execute log filter category 0
# execute log display
1 logs found.
1 logs returned.

1: date=2023-11-01 time=17:04:39 eventtime=1698883474200006380 tz="-0700"
logid="0002000012" type="traffic" subtype="multicast" level="notice" vd="vd1"
srcip=10.1.100.11 srcport=52673 srcintf="port38" srcintfrole="undefined"
dstip=239.1.1.10 dstport=5001 dstintf="port37" dstintfrole="undefined"
srccountry="Reserved" dstcountry="Reserved" sessionid=18 proto=17 action="accept"
policyid=1 policytype="multicast-policy" poluuid="09bdd086-78e2-51ee-1d61-
0955f9046b53" policyname="mcast-ips" service="udp/5001" trandisp="noop" duration=180
sentbyte=2996 rcvdbyte=0 sentpkt=2 rcvdpkt=0 appcat="unscanned" utmref=0-266
```

c. Verify the multicast session list:

```
# diagnose sys mcast-session list

session info: id=19 vf=1 proto=17 10.1.100.11.56538->239.1.1.10.5001
used=2 path=1 duration=2 expire=177 indev=10
state=00000000:
session-npu-info: ipid/vlifid=0/0 vlanid/vtag_in=0/0 in_npuid=0 tae_index=0 qid=0
fwd_map=0x00000000
path: log ndr policy=1, outdev=9, tos=0xff
Total 1 sessions
```

Optimize virtual patching on the local-in interface - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- Virtual patching on the local-in management interface

Virtual patching is a method of mitigating vulnerability exploits by using the FortiGate's IPS engine to block known vulnerabilities. Virtual patching can be applied to traffic destined to the FortiGate by applying IPS signatures to the local-in interface using local-in policies.

When virtual patching is enabled in a local-in policy, the IPS engine queries the FortiGuard API server to:

- Obtain a list of vulnerabilities targeting the FortiGate on a particular version
- Determine whether the session destined to the local-in interface on the FortiGate requires a scan by identifying and tagging services in the session. The session's port number and protocol are used to identify the services. Currently only SSL VPN and web GUI services are tagged in a session.

If a tagged session lacks vulnerability signatures for the FortiOS version, then the IPS engine bypasses the session. This optimizes performance by only scanning and dropping sessions that are exploiting a vulnerability.

Example

In this example, virtual patching is enabled for the local-in policy and the following scenarios are described:

- FortiGate with an SSL VPN vulnerability
- FortiGate with a web GUI vulnerability
- FortiGate with both an SSL VPN and web GUI vulnerability

To enable virtual patching:

1. Enable virtual patching in the local-in policy:

```
config firewall local-in-policy
    edit 1
        set intf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set service "ALL"
        set schedule "always"
        set virtual-patch enable
    next
end
```

Because the IPS engine can currently only tag services related to SSL VPN and web GUI signatures, all other protocols are scanned when `service` is set to `ALL`. However, you can bypass scanning of other protocols, such as SSH and FTP, by setting `service` to only `HTTPS`.

2. Observe the outcome of the following scenarios:
   - In this example, FortiOS has an SSL VPN vulnerability. The IPS engine drops SSL VPN traffic to the local-in interface on the FortiGate and bypasses web GUI traffic. Traffic for other services is scanned and passed to the interface.

     Following is a log of the SSL VPN traffic that was dropped because of the vulnerability. Bypassed web GUI traffic did not generate any logs.

```
# diagnose ips vpatch fmwp-status
Enabled FMWP signatures: 3

  10002887 FortiOS.SSL-VPN.Heap.Buffer.Overflow.

1: date=2023-11-07 time=14:53:44 eventtime=1699325624346021995 tz="+1200"
logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert"
vd="root" severity="critical" srcip=10.1.100.22 srccountry="Reserved"
dstip=10.1.100.1 dstcountry="Reserved" srcintf="port2" srcintfrole="undefined"
dstintf="root" dstintfrole="undefined" sessionid=284 action="dropped" proto=6
service="HTTPS" policyid=1 attack="FortiOS.SSL-VPN.Heap.Buffer.Overflow."
srcport=53250 dstport=11443 hostname="myfortigate.example" url="/error"
httpmethod="POST" direction="outgoing" attackid=10002887
ref="http://www.fortinet.com/ids/VID10002887" incidentserialno=99614721 msg="vPatch:
FortiOS.SSL-VPN.Heap.Buffer.Overflow." crscore=50 craction=4096 crlevel="critical"
```

   - In this example, FortiOS has a web GUI vulnerability. The IPS engine drops web GUI traffic to the local-in interface on the FortiGate and bypasses SSL VPN traffic. Traffic for other services is scanned and passed to the interface.

Following is a log of the web GUI traffic that was dropped because of the vulnerability. Bypassed SSL VPN traffic did not generate any logs.

```
# diagnose ips vpatch fmwp-status
Enabled FMWP signatures: 2

  10002156 FortiOS.NodeJS.Proxy.Authentication.Bypass.
  10002890 FortiOS.HTTPD.Content-Length.Memory.Corruption.

1: date=2023-11-07 time=14:55:15 eventtime=1699325715311370215 tz="+1200"
logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert"
vd="root" severity="critical" srcip=10.1.100.22 srccountry="Reserved"
dstip=10.1.100.1 dstcountry="Reserved" srcintf="port2" srcintfrole="undefined"
dstintf="root" dstintfrole="undefined" sessionid=304 action="dropped" proto=6
service="HTTPS" policyid=1 attack="FortiOS.NodeJS.Proxy.Authentication.Bypass."
srcport=53622 dstport=443 hostname="127.0.0.1:9980" url="/api/v2/cmdb/system/admin"
agent="Node.js" httpmethod="GET" direction="outgoing" attackid=10002156
ref="http://www.fortinet.com/ids/VID10002156" incidentserialno=99614722 msg="vPatch:
FortiOS.NodeJS.Proxy.Authentication.Bypass." crscore=50 craction=4096
crlevel="critical"
```

- In this example, FortiOS has an SSL VPN and a web GUI vulnerability. The IPS engine drops both SSL VPN and web GUI traffic to the local-in interface on the FortiGate. Traffic for other services is scanned and passed to the interface.

  Following is a log of the SSL VPN and web GUI traffic that was dropped because of the vulnerability.

```
# diagnose ips vpatch fmwp-status
Enabled FMWP signatures: 3

  10002156 FortiOS.NodeJS.Proxy.Authentication.Bypass.
  10002887 FortiOS.SSL-VPN.Heap.Buffer.Overflow.
  10002890 FortiOS.HTTPD.Content-Length.Memory.Corruption.

1: date=2023-11-07 time=06:42:44 eventtime=1699296164649894963 tz="+1200"
logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert"
vd="root" severity="critical" srcip=10.1.100.22 srccountry="Reserved"
dstip=10.1.100.1 dstcountry="Reserved" srcintf="port2" srcintfrole="undefined"
dstintf="root" dstintfrole="undefined" sessionid=1094 action="dropped" proto=6
service="HTTPS" policyid=1 attack="FortiOS.SSL-VPN.Heap.Buffer.Overflow."
srcport=44164 dstport=10443 hostname="myfortigate.example" url="/error"
httpmethod="POST" direction="outgoing" attackid=10002887
ref="http://www.fortinet.com/ids/VID10002887" incidentserialno=116392250 msg="vPatch:
FortiOS.SSL-VPN.Heap.Buffer.Overflow." crscore=50 craction=4096 crlevel="critical"

2: date=2023-11-07 time=06:42:09 eventtime=1699296129458704870 tz="+1200"
logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert"
vd="root" severity="critical" srcip=10.1.100.22 srccountry="Reserved"
dstip=10.1.100.1 dstcountry="Reserved" srcintf="port2" srcintfrole="undefined"
dstintf="root" dstintfrole="undefined" sessionid=1066 action="dropped" proto=6
service="HTTPS" policyid=1 attack="FortiOS.NodeJS.Proxy.Authentication.Bypass."
srcport=42352 dstport=443 hostname="127.0.0.1:9980" url="/api/v2/cmdb/system/admin"
agent="Node.js" httpmethod="GET" direction="outgoing" attackid=10002156
ref="http://www.fortinet.com/ids/VID10002156" incidentserialno=116392236 msg="vPatch:
FortiOS.NodeJS.Proxy.Authentication.Bypass." crscore=50 craction=4096
crlevel="critical"
```

Objects

This section includes information about object related new features:

- Increase the number of supported dynamic FSSO IP addresses on page 315
- Internet service as source addresses in the local-in policy 7.4.4 on page 317

Increase the number of supported dynamic FSSO IP addresses

Increase the number of supported dynamic FSSO IP addresses from 100 to 3000 per dynamic FSSO group. The dynamic FSSO type addresses can be pointed to FortiManager's Universal Connector, which imports the addresses from Cisco ACI or Guardicore Centra.

For more information about the FortiManager Universal Connector, see Universal Connector MEA, Cisco ACI Fabric connectors, and Using the imported EPGs in the FortiManager documentation.

Example

In this example, FSSO user logon events are used to populate a dynamic FSSO address object (fsso-dyn-37).

To configure the FSSO dynamic address object:

1. From the diagnostics, collect the list of FSSO dynamic addresses:

```
# diagnose debug authd fsso show-address

FSSO Dynamic Addresses(master=1):
ad-fsso-1, ref 1
ADGRP: FORTINET-FSSO/GROUP1
ADDR(LI): 10.1.100.188
fsso-dyn-1, ref 1
ADGRP: CN=FSSOB20,OU=FSSO-BULK,DC=FORTINET-FSSO,DC=COM
ADDR(LI): 10.0.0.2
ADDR(LI): 10.0.0.3
ADDR(LI): 10.0.0.4
...
ADDR(LI): 10.0.179.175
ADDR(LI): 10.0.179.176
ADDR(LI): 10.0.179.177
fsso-dyn-18, ref 1
ADGRP: CN=FSSOB37,OU=FSSO-BULK,DC=FORTINET-FSSO,DC=COM
ADDR(LI): 10.0.203.34
ADDR(LI): 10.0.203.35
ADDR(LI): 10.0.203.36
...
ADDR(LI): 10.0.214.214
ADDR(LI): 10.0.214.215
ADDR(LI): 10.0.214.216
```

```
fsso-dyn-19, ref 1
ADGRP: CN=FSSOB36,OU=FSSO-BULK,DC=FORTINET-FSSO,DC=COM
ADDR(LI): 10.0.191.106
```

The range of the CN=FSSOB37,OU=FSSO-BULK,DC=FORTINET-FSSO,DC=COM group is 10.0.203.34 to 10.0.214.216.

2. Create the dynamic address object:

```
config firewall address
    edit "fsso-dyn-37"
        set type dynamic
        set sub-type fsso
        set fsso-group "CN=FSSOB37,OU=FSSO-BULK,DC=FORTINET-FSSO,DC=COM"
    next
end
```

3. Add the dynamic address object to a firewall policy:

```
config firewall policy
    edit 3
        set name "pol1"
        set srcintf "port10"
        set dstintf "port9"
        set action accept
        set srcaddr "ad-fsso-1" "fsso-dyn-37"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set av-profile "default"
        set logtraffic all
        set nat enable
        set groups "ad-fsso-grp1"
    next
end
```

4. Verify the policy traffic:

```
# diagnose firewall iprope list 100004
policy index=3 uuid_idx=561 action=accept
flag (8052129): log redir auth nat nids_raw master use_src pol_stats
flag2 (6004): fsso log_fail resolve_sso
flag3 (b0): !sp link-local best-route
schedule(always)
cos_fwd=255  cos_rev=255
group=00100004 av=00004e20 au=00000003 split=00000000
host=0 chk_client_info=0x1 app_list=0 ips_view=1
misc=0
zone(1): 18 -> zone(1): 17
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=542,
source dynamic address (2):  uuid_idx=582
        fsso-dyn-37 ID(37)
        RANGE(10.0.203.34-10.0.214.216)
 uuid_idx=548
        ad-fsso-1 ID(237)
        ADDR(10.1.100.188)
```

```
user group(1): 2
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto
```

Internet service as source addresses in the local-in policy - 7.4.4

An internet service can be used as the source address in a local-in policy. This allows for more flexibility and control when managing local traffic, enhancing network security and efficiency.

```
config firewall local-in-policy
    edit <id>
        set internet-service-src {enable | disable}
        set internet-service-src-name <string>
        set internet-service-src-group <string>
        set internet-service-src-custom <string>
        set internet-service-src-custom-group <string>
        set internet-service-src-negate {enable | disable}
    next
end
```

internet-service-src {enable | disable}	Enable/disable use of Internet Services in source for this local-in policy. If enabled, the source address is not used.
internet-service-src-name <string>	Internet Service source name.
internet-service-src-group <string>	Internet Service source group name.
internet-service-src-custom <string>	Custom Internet Service source name.
internet-service-src-custom-group <string>	Custom Internet Service source group name.
internet-service-src-negate {enable | disable}	When enabled, `internet-service-src` specifies what the service must NOT be.

In this example, the internet service Malicious-Malicious.Server is applied in local-in policy. Packets then sent to the FortiGate from a client with an IP address that belongs to that internet service. The local-in policy should block the packet.

To configure the local-in policy, send a packet, and then check the results:

1. Apply the Malicious-Malicious.Server internet service in the local-in policy:

```
config firewall local-in-policy
    edit 1
        set intf "port3"
        set dstaddr "all"
        set internet-service-src enable
        set internet-service-src-name "Malicious-Malicious.Server"
        set service "ALL_ICMP" "ALL_TCP"
        set schedule "always"
```

```
    next
end
```

2. Configure the interface used in the local-in policy to allow ping, HTTPS, and SSH access:

```
config system interface
    edit "port3"
        set vdom "vdom1"
        set ip 10.2.2.2 255.255.255.0
        set allowaccess ping https ssh
        set type physical
        set device-identification enable
        set snmp-index 5
    next
end
```

3. Enable local-in-deny-unicast logging so that the policy blocking results can be checked:

```
config log setting
    set local-in-deny-unicast enable
end
```

4. Send packets from the client IP address 1.0.1.21, which belongs to the Malicious-Malicious.Server internet service. The packet will hit the local-in policy and the FortiGate will not respond for incoming ICMP or SSH packets.

```
# diagnose sniffer packet any icmp 4
interfaces=[any]
filters=[icmp]
34.814391 port3 in 1.0.1.21 -> 10.2.2.2: icmp: echo request
35.814252 port3 in 1.0.1.21 -> 10.2.2.2: icmp: echo request
36.814121 port3 in 1.0.1.21 -> 10.2.2.2: icmp: echo request
37.813983 port3 in 1.0.1.21 -> 10.2.2.2: icmp: echo request
38.813847 port3 in 1.0.1.21 -> 10.2.2.2: icmp: echo request
^C
5 packets received by filter
0 packets dropped by kernel
```

```
# diagnose sniffer packet any 'tcp and port 22' 4
interfaces=[any]
filters=[tcp and port 22]
5.988037 port3 in 1.0.1.21.21102 -> 10.2.2.2.22: syn 2964400061
6.985778 port3 in 1.0.1.21.21102 -> 10.2.2.2.22: syn 2964400061
8.986481 port3 in 1.0.1.21.21102 -> 10.2.2.2.22: syn 2964400061
12.997883 port3 in 1.0.1.21.21102 -> 10.2.2.2.22: syn 2964400061
^C
4 packets received by filter
0 packets dropped by kernel
```

5. Check the local-in traffic log to confirm that the ICMP and SSH packets were blocked:

```
1: date=2024-04-08 time=15:14:38 eventtime=1712643278466511132 tz="-0700"
logid="0001000014" type="traffic" subtype="local" level="notice" vd="vdom1"
srcip=1.0.1.21 identifier=1 srcintf="port3" srcintfrole="undefined" dstip=10.2.2.2
dstintf="vdom1" dstintfrole="undefined" srcinetsvc="Malicious-Malicious.Server"
srccountry="China" srcregion="Fujian" srccity="Sanming" dstcountry="Reserved"
sessionid=29356 proto=1 action="deny" policyid=1 policytype="local-in-policy"
poluuid="dd003848-f633-51ee-7dad-cc8be11d188e" service="icmp" trandisp="noop" app="icmp"
duration=0 sentbyte=0 rcvdbyte=0 sentpkt=0 rcvdpkt=0 appcat="unscanned" crscore=5
craction=262144 crlevel="low" msg="Connection Failed" srchwvendor="Fortinet"
```

```
devtype="Unknown" osname="Unknown" mastersrcmac="70:4c:a5:97:d9:26"
srcmac="70:4c:a5:97:d9:26" srcserver=0

6: date=2024-04-08 time=15:09:30 eventtime=1712642970682804537 tz="-0700"
logid="0001000014" type="traffic" subtype="local" level="notice" vd="vdom1"
srcip=1.0.1.21 srcport=21102 srcintf="port3" srcintfrole="undefined" dstip=10.2.2.2
dstport=22 dstintf="vdom1" dstintfrole="undefined" srcinetsvc="Malicious-
Malicious.Server" srccountry="China" dstcountry="Reserved" sessionid=29240 proto=6
action="deny" policyid=1 policytype="local-in-policy" poluuid="dd003848-f633-51ee-7dad-
cc8be11d188e" service="SSH" trandisp="noop" app="Console Management(SSH)" duration=0
sentbyte=0 rcvdbyte=0 sentpkt=0 rcvdpkt=0 appcat="unscanned" crscore=5 craction=262144
crlevel="low" msg="Connection Failed" srchwvendor="Fortinet" devtype="Unknown"
osname="Unknown" mastersrcmac="70:4c:a5:97:d9:26" srcmac="70:4c:a5:97:d9:26" srcserver=0
```

Traffic shaping

This section includes information about traffic shaping related new features:

- Traffic shaping extensions on page 319

Traffic shaping extensions

This information is also available in the FortiOS 7.4 Administration Guide:
- Local-in and local-out traffic matching
- VLAN CoS matching on a traffic shaping policy
- Multi-stage VLAN CoS marking

Traffic shaping now supports the following.

- Local-in and local-out traffic matching: the FortiGate can apply shaping policies to local traffic entering or leaving the firewall interface based on source and destination IP addresses, ports, protocols, and applications.
- VLAN CoS matching on a shaping policy: the FortiGate can use the class of service (CoS) value of VLAN packets as a matching criterion for shaping policies. This enables the FortiGate to prioritize traffic based on the CoS value assigned by the switch or router.
- Multi-stage VLAN CoS marking: the FortiGate can configure the traffic shaper to dynamically change the CoS value of outgoing VLAN packets based on the shaper profile. This allows the FortiGate to mark traffic with different CoS values at different stages of the shaping process.

```
config firewall shaping-policy
    edit <id>
        set traffic-type {forwarding | local-in | local-out}
        set cos-mask <3-bit_binary>
        set cos <3-bit_binary>
    next
end
```

`traffic-type {forwarding | local-in | local-out}`	Set the traffic type. • `forwarding`: use forwarding traffic (default) • `local-in`: local-in traffic

	• `local-out`: local-out traffic
`cos-mask <3-bit_binary>`	Set the VLAN CoS evaluated bits, 3-bit binary (000 - 111). This setting is only available for forwarding traffic.
`cos <3-bit_binary>`	Set the VLAN CoS bit pattern, 3-bit binary (000 - 111). This setting is available once `cos-mask` is configured.

```
config firewall shaper traffic-shaper
    edit <name>
        set bandwidth-unit {kbps | mbps | gbps}
        set guaranteed-bandwidth <integer>
        set maximum-bandwidth <integer>
        set cos-marking {enable | disable}
        set cos-marking-method {static | multi-stage}
        set cos <3-bit_binary>
        set exceed-cos <3-bit_binary>
        set maximum-cos <3-bit_binary>
        set exceed-bandwidth <integer>
    next
end
```

`cos-marking {enable | disable}`	Enable/disable VLAN CoS marking (default = `disable`).
`cos-marking-method {static | multi-stage}`	Set the VLAN CoS marking method. • `static`: use static VLAN CoS marking (default) • `multi-stage`: multi-stage VLAN CoS marking
`cos <3-bit_binary>`	Set the VLAN CoS mark, 3-bit binary (000 - 111).
`exceed-cos <3-bit_binary>`	Set the VLAN CoS mark for traffic in `guaranteed-bandwidth` and `exceed-bandwidth`, 3-bit binary (000 - 111).
`maximum-cos <3-bit_binary>`	Set the VLAN CoS mark for traffic in `exceed-bandwidth` and `maximum-bandwidth`, 3-bit binary (000 - 111).
`exceed-bandwidth <integer>`	Set the exceed bandwidth used for DSCP or VLAN CoS multi-stage marking. The integer value range depends on the `bandwidth-unit` setting. This setting is only available for CoS multi-stage marking.

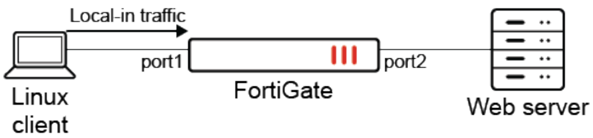
Examples

This topic contains the following examples:

- Example 1: local-in traffic shaping
- Example 2: local-out traffic shaping
- Example 3: VLAN CoS marking on a firewall policy and matching on a shaping policy
- Example 4: mutli-stage VLAN CoS marking on a traffic shaper

Example 1: local-in traffic shaping

In this example, the traffic shaping policy applies to local-in traffic. The local-in traffic originates from the Linux client and is destined to port1 on the FortiGate.



To configure the traffic shaping policy:

```
config firewall shaping-policy
    edit 2
        set traffic-type local-in
        set service "ALL"
        set schedule "always"
        set class-id 3
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

To verify the configuration:

1. Check the shaping policy information for local-in traffic to verify that the correct class ID (3) is applied:

```
# diagnose firewall iprope list 100018
policy index=2 uuid_idx=1300 action=accept
flag (0):
schedule(always)
cos_fwd=0  cos_rev=0
group=00100018 av=00000000 au=00000000 split=00000000
host=1 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 0 -> zone(1): 0
source(1): 0.0.0.0-255.255.255.255, uuid_idx=1106,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=1106,
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto
class_id: 3
```

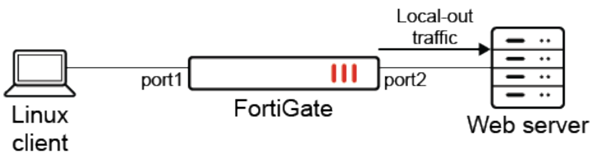
2. Check the session list to verify that the class ID (3) matches the shaping policy ID (2):

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=1195 expire=3574 timeout=3600
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=3 shaping_policy_id=2 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log local may_dirty
statistic(bytes/packets/allow_err): org=18274/350/1 reply=826037/603/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 1/0
orgin->sink: org pre->in, reply out->post dev=17->34/34->17 gwy=172.16.200.2/0.0.0.0
hook=pre dir=org act=noop 172.16.200.254:55432->172.16.200.2:443(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.2:443->172.16.200.254:55432(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=08:5b:0e:7d:42:db
misc=0 policy_id=4294967295 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=1
serial=0000009d tos=ff/ff app_list=0 app=0 url_cat=0
```

```
rpdb_link_id=00000000 ngfwid=n/a
npu_state=00000000
no_ofld_reason:  local
```

Example 2: local-out traffic shaping

In this example, the traffic shaping policy applies to local-out traffic. The local-out traffic originates from port2 on the FortiGate and is destined to an external web server.



To configure the traffic shaping policy:

```
 config firewall shaping-policy
    edit 3
        set traffic-type local-out
        set service "ALL"
        set schedule "always"
        set class-id 2
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

To verify the configuration:

1. Check the shaping policy information for local-out traffic to verify that the correct class ID (2) is applied:

```
# diagnose firewall iprope list 100019
policy index=3 uuid_idx=1301 action=accept
flag (0):
schedule(always)
cos_fwd=0  cos_rev=0
group=00100019 av=00000000 au=00000000 split=00000000
host=1 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 0 -> zone(1): 0
source(1): 0.0.0.0-255.255.255.255, uuid_idx=1106,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=1106,
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto
class_id: 2
```

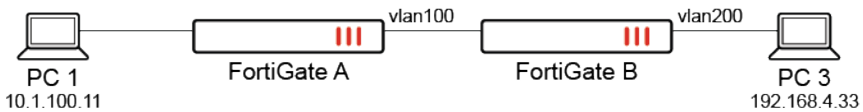
2. Check the session list to verify that the class ID (2) matches the shaping policy ID (3):

```
# diagnose sys session list
session info: proto=6 proto_state=05 duration=40 expire=110 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=2 shaping_policy_id=3 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=255/255
```

```
state=log local
statistic(bytes/packets/allow_err): org=3676/14/1 reply=3848/11/1 tuples=2
tx speed(Bps/kbps): 90/0 rx speed(Bps/kbps): 94/0
orgin->sink: org out->post, reply pre->in dev=34->17/17->34 gwy=0.0.0.0/172.16.200.2
hook=out dir=org act=noop 172.16.200.2:19178->140.174.22.68:443(0.0.0.0:0)
hook=in dir=reply act=noop 140.174.22.68:443->172.16.200.2:19178(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
dst_mac=08:5b:0e:7d:42:db
misc=0 policy_id=0 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=1
serial=00000f1b tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=00000000
no_ofld_reason:  local
```

Example 3: VLAN CoS marking on a firewall policy and matching on a shaping policy

In this example, FortiGate A forwards traffic to FortiGate B with VLAN CoS 3, which matches firewall policy 6. When FortiGate B receives traffic, it applies the traffic shaping policy and will prioritize based on the CoS value.



The VLAN CoS range is 000 to 111 (0 - 7), which includes the following values: 000, 001, 010, 011, 100, 101, 110, and 111. The `cos` and `cos-mask` settings can be used to match multiple `vlan_cos` values with a single shaping policy. The following matching logic is used: (`vlan_cos` AND `cos-mask`) == (`cos` AND `cos-mask`).

To match all possible `vlan_cos` values, set the `cos-mask` to `000`.

To configure VLAN CoS marking with traffic shaping:

1. Configure the firewall policy on FortiGate A with VLAN CoS forwarding:

```
config firewall policy
    edit 6
        set srcintf "port1"
        set dstintf "vlan100"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set vlan-cos-fwd 3
    next
end
```

   Traffic marked with CoS 3 will be forwarded to FortiGate B.

2. On FortiGate A, check the session list to verify that CoS 3 is marked:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=1 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=3/255
state=log may_dirty npu f00
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=19->47/47->19 gwy=20.20.20.2/10.1.100.11
hook=pre dir=org act=noop 10.1.100.11:28489->192.168.4.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.4.33:28489->10.1.100.11:0(0.0.0.0:0)
src_mac=00:0c:29:57:2a:01  dst_mac=70:4c:a5:7d:d4:95
misc=0 policy_id=6 pol_uuid_idx=1128 auth_info=0 chk_client_info=0 vd=2
serial=000717ca tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=79/78, ipid=78/79,
vlan=0x0000/0x0064
vlifid=78/79, vtag_in=0x0000/0x0064 in_npu=2/2, out_npu=2/2, fwd_en=0/0, qid=0/1
```

3. Configure the traffic shaping policy to match VLAN CoS 3:

```
config firewall shaping-policy
    edit 1
        set traffic-type forwarding
        set name "vlan-cos-matching"
        set service "ALL"
        set srcintf "vlan100"
        set dstintf "vlan200"
        set class-id 2
        set cos-mask 111
        set cos 011
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

Based on this shaping policy:

- `vlan_cos` = 3, which corresponds to `011`

  `cos-mask` = `111`

  AND both get `011`
- `cos-mask` = `111`

  `cos` = `011`

  AND both get `011`
- (`vlan_cos` AND `cos-mask`) == (`cos` AND `cos-mask`), so traffic will pass

The shaping policy will match `vlan_cos3`.

4. Configure the firewall policy on FortiGate B:

```
config firewall policy
    edit 3
        set srcintf "vlan100"
        set dstintf "vlan200"
```

```
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
end
```

5. On FortiGate B, check the session list to verify that the class ID (2) matches the shaping policy ID (1):

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=672 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=2 shaping_policy_id=1 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=56532/673/1 reply=56532/673/1 tuples=2
tx speed(Bps/kbps): 82/0 rx speed(Bps/kbps): 82/0
orgin->sink: org pre->post, reply pre->post dev=59->61/61->59 gwy=20.20.200.3/20.20.20.1
hook=pre dir=org act=noop 10.1.100.11:28735->192.168.4.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.4.33:28735->10.1.100.11:0(0.0.0.0:0)
src_mac=90:6c:ac:fb:bb:97  dst_mac=04:d5:90:36:73:3f
misc=0 policy_id=3 pol_uuid_idx=1245 auth_info=0 chk_client_info=0 vd=1
serial=0000160b tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x040000
no_ofld_reason:  non-npu-intf
```

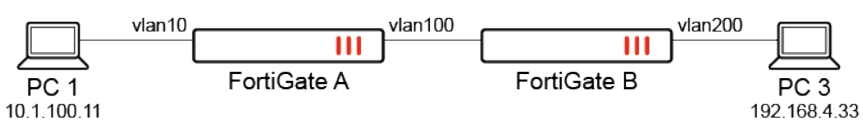
If a particular session matches both the `firewall policy` and `firewall shaping-policy`, then anything configured in the `firewall shaping-policy` overrides whatever was configured in the `firewall policy`.

Example 4: mutli-stage VLAN CoS marking on a traffic shaper

In this example, mutli-stage VLAN CoS marking is configured using traffic shapers on FortiGate A and FortiGate B. FortiGate A applies multi-stage CoS marking with the following traffic shaper settings:

- Traffic below the guaranteed bandwidth will apply CoS 6.
- Traffic greater than the guaranteed bandwidth will apply CoS 6 and 5.
- Traffic greater than the exceed bandwidth will apply CoS 6, 5, and 4.

A traffic shaper and shaping policy are configured on FortiGate B. When traffic comes from FortiGate A with CoS 6, the traffic shaping policy will be applied because the CoS matches.



Multi-stage VLAN CoS marking is not supported on NP models. Traffic is not offloaded when it is enabled.

To configure mutli-stage VLAN CoS marking on FortiGate A:

1. Configure the firewall policy:

```
config firewall policy
    edit 7
        set srcintf "port1"
        set dstintf "vlan100"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set traffic-shaper "multi-stage-cos-fgta"
        set traffic-shaper-reverse "multi-stage-cos-fgta"
    next
end
```

2. Configure the traffic shaper:

```
config firewall shaper traffic-shaper
    edit "multi-stage-cos-fgta"
        set guaranteed-bandwidth 1000
        set maximum-bandwidth 4000
        set per-policy enable
        set exceed-bandwidth 2000
        set cos-marking enable
        set cos-marking-method multi-stage
        set cos 110
        set exceed-cos 101
        set maximum-cos 100
    next
end
```

3. Check the session list to verify that CoS 6 is marked:

```
# diagnose sys session list
session info: proto=17 proto_state=00 duration=6 expire=180 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=multi-stage-cos-fgta prio=2 guarantee 125000Bps max 500000Bps traffic
504900Bps drops 163905268B
reply-shaper=multi-stage-cos-fgta prio=2 guarantee 125000Bps max 500000Bps traffic
504900Bps drops 0B
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=6/6
state=log may_dirty npu npd os rs f00
statistic(bytes/packets/allow_err): org=3804176/292/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 583462/4667 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=19->47/47->19 gwy=20.20.20.2/0.0.0.0
```

```
hook=pre dir=org act=noop 10.1.100.11:37586->192.168.4.33:5001(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.4.33:5001->10.1.100.11:37586(0.0.0.0:0)
src_mac=00:0c:29:57:2a:01  dst_mac=70:4c:a5:7d:d4:95
misc=0 policy_id=7 pol_uuid_idx=1129 auth_info=0 chk_client_info=0 vd=2
serial=0006613c tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0,
vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:  offload-denied
```

To configure mutli-stage VLAN CoS marking on FortiGate B:

1. Configure the firewall policy:

```
config firewall policy
    edit 4
        set srcintf "vlan100"
        set dstintf "vlan200"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
end
```

2. Configure the traffic shaper:

```
config firewall shaper traffic-shaper
    edit "multi-stage-cos-fgtb"
        set guaranteed-bandwidth 250
        set maximum-bandwidth 1000
        set per-policy enable
        set cos-marking enable
        set cos-marking-method multi-stage
        set cos 100
        set exceed-cos 101
        set maximum-cos 110
        set exceed-bandwidth 500
    next
end
```

Based on this traffic shaper, the following CoS marking rules will be applied:

- If all traffic is less than the guaranteed bandwidth, then the traffic will be marked with CoS 4.
- If all traffic is greater than the guaranteed bandwidth but less than the exceed bandwidth, then 50% of the traffic will be marked as CoS 4 and 50% as CoS 5.
- If traffic is greater than the guaranteed bandwidth but less than the maximum bandwidth, then 50% of the traffic will be marked as CoS 6; CoS 4 and 5 will have another 50%.
- If traffic is greater than the maximum bandwidth, then 50% of the traffic will be marked as CoS 6, 25% will be marked as CoS 4, and 25% will be marked as CoS 5. Packet drops will be visible in the debug output.

3. Configure the traffic shaping policy:

```
config firewall shaping-policy
    edit 1
        set service "ALL"
        set srcintf "vlan100"
        set dstintf "vlan200"
        set traffic-shaper "multi-stage-cos-fgtb"
        set traffic-shaper-reverse "multi-stage-cos-fgtb"
        set class-id 2
        set cos-mask 111
        set cos 110
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

4. Check the session list to verify that the shaping ID (1) applied and CoS 4 is marked:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=1 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=multi-stage-cos-fgtb prio=2 guarantee 31250Bps max 125000Bps traffic
236Bps drops 0B
reply-shaper=multi-stage-cos-fgtb prio=2 guarantee 31250Bps max 125000Bps traffic 236Bps
drops 0B
per_ip_shaper=
class_id=2 shaping_policy_id=1 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=4/4
state=log may_dirty os rs f00
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 120/0 rx speed(Bps/kbps): 120/0
orgin->sink: org pre->post, reply pre->post dev=59->61/61->59 gwy=20.20.200.3/20.20.20.1
hook=pre dir=org act=noop 10.1.100.11:29899->192.168.4.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.4.33:29899->10.1.100.11:0(0.0.0.0:0)
src_mac=90:6c:ac:fb:bb:97  dst_mac=04:d5:90:36:73:3f
misc=0 policy_id=3 pol_uuid_idx=1377 auth_info=0 chk_client_info=0 vd=4
serial=00024329 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x040000
no_ofld_reason:  non-npu-intf
total session 1
```

Protocol options

This section includes information about protocol options related new features:

Stripping the X-Forwarded-For value in the HTTP header - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- Stripping the X-Forwarded-For value in the HTTP header

The X-Forwarded-For value in the HTTP header can be stripped when the `strip-x-forwarded-for` option is enabled under `firewall profile-protocol-options`. This feature sets the value to empty using the IPS engine.

The following types of traffic support X-Forwarded-For stripping:

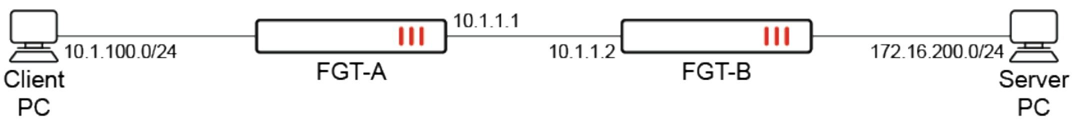
- HTTP/1.1, HTTP/2, and HTTP/3 traffic that matches an NGFW mode security policy with flow-based inspection.
- Plain HTTP/1.1 traffic that matches a firewall policy with proxy-based inspection.

The following types of traffic do not support X-Forwarded-For stripping:

- HTTPS traffic that matches a firewall policy with proxy-based inspection.
- HTTP and HTTPS traffic that matches an explicit web proxy policy.

Example

In this example, FGT-A is configured with `strip-x-forwarded-for` enabled for HTTP. On FGT-B, the IPS sensor is configured to monitor the Eicar.Virus.Test.File signature. The IPS logs on FGT-B are used to verify the traffic sent from FGT-A to FGT-B, namely the `forwardedfor` value in the `rawdata` field.



To configure X-Forwarded-For stripping:

1. Configure FGT-A:
   a. Configure the protocol options for HTTP:

```
config firewall profile-protocol-options
    edit "protocol-xff"
        config http
            set ports 80
            unset options
            set strip-x-forwarded-for enable
            unset post-lang
        end
    next
end
```

   b. Configure the firewall policy (ensure that an IPS sensor is applied):

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port5"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set profile-protocol-options "protocol-xff"
        set ssl-ssh-profile "ssl-deep"
        set ips-sensor "default"
        set nat enable
```

```
    next
end
```

2. Configure FGT-B:
   a. Configure the IPS sensor with extended logging:

```
config ips sensor
    edit "monitor-eicar"
        set extended-log enable
        config entries
            edit 1
                set rule 29844
                set status enable
                set action pass
            next
        end
    next
end
```

   b. Configure the firewall policy (ensure that an IPS sensor is applied):

```
config firewall policy
    edit 3
        set srcintf "port5"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "ssl-deep"
        set ips-sensor "monitor-eicar"
        set nat enable
    next
end
```

To verify the configuration:

1. Use a cURL request to send HTTPS traffic with HTTP header X-Forwarded-For from the Client PC to the Server PC:

```
curl -vk -H "X-Forwarded-For: 10.22.22.22" https://172.16.200.52/eicar.com
```

2. On FGT-B, verify the corresponding IPS logs.
   a. For HTTP/1.1, the `X-Forwarded-For` value is removed from the `rawdata` field, and the `forwardedfor` value is not included:

```
1: date=2023-09-21 time=14:05:34 eventtime=1695330334919589600 logid="0419016384"
type="utm" subtype="ips" eventtype="signature" level="alert" vd="root"
severity="info" srcip=10.1.1.1 srccountry="Reserved" dstip=172.16.200.42
dstcountry="Reserved" srcintf="port5" srcintfrole="undefined" dstintf="port1"
dstintfrole="undefined" sessionid=2471 action="detected" proto=6 service="HTTPS"
policyid=3 poluuid="782b9e86-58a3-51ee-8e0f-79c7682223dd" policytype="policy"
attack="Eicar.Virus.Test.File" srcport=36018 dstport=443 hostname="172.16.200.42"
url="/eicar.com" agent="curl/7.61.1" httpmethod="GET" direction="incoming"
attackid=29844 profile="monitor-eicar" ref="http://www.fortinet.com/ids/VID29844"
incidentserialno=75497475 msg="file_transfer: Eicar.Virus.Test.File" rawdataid="1/1"
```

```
rawdata="Response-Content-Type=application/x-msdos-program" crscore=5 craction=65536
crlevel="low"
```

b. For HTTP/2 and HTTP/3, the `X-Forwarded-For` value is removed from the `rawdata` field, and `forwardedfor` is included:

```
1: date=2023-09-21 time=14:05:56 eventtime=1695330356543624871 logid="0419016384"
type="utm" subtype="ips" eventtype="signature" level="alert" vd="root"
severity="info" srcip=10.1.1.1 srccountry="Reserved" dstip=172.16.200.52
dstcountry="Reserved" srcintf="port5" srcintfrole="undefined" dstintf="port1"
dstintfrole="undefined" sessionid=2474 action="detected" proto=6 service="HTTPS"
policyid=3 poluuid="782b9e86-58a3-51ee-8e0f-79c7682223dd" policytype="policy"
attack="Eicar.Virus.Test.File" srcport=37786 dstport=443 hostname="172.16.200.52"
url="/eicar.com" agent="curl/7.61.1" httpmethod="GET" direction="incoming"
attackid=29844 profile="monitor-eicar" ref="http://www.fortinet.com/ids/VID29844"
incidentserialno=75497476 msg="file_transfer: Eicar.Virus.Test.File" rawdataid="1/1"
forwardedfor="\r\n" rawdata="Response-Content-Type=application/x-msdos-program"
crscore=5 craction=65536 crlevel="low"
```

3. On FGT-A, disable `strip-x-forwarded-for` for HTTP:

```
config firewall profile-protocol-options
    edit "protocol-xff"
        config http
            set strip-x-forwarded-for disable
        end
    next
end
```

4. Send the same HTTPS traffic with HTTP header X-Forwarded-For from the Client PC to the Server PC.
5. On FGT-B, verify the corresponding IPS log, which includes `forwardedfor` and `X-Forwarded-For` values in the `rawdata` field:

```
1: date=2023-09-21 time=16:33:06 eventtime=1695339187144132034 logid="0419016384"
type="utm" subtype="ips" eventtype="signature" level="alert" vd="root" severity="info"
srcip=10.1.1.1 srccountry="Reserved" dstip=172.16.200.52 dstcountry="Reserved"
srcintf="port5" srcintfrole="undefined" dstintf="port1" dstintfrole="undefined"
sessionid=3776 action="detected" proto=6 service="HTTPS" policyid=3 poluuid="782b9e86-
58a3-51ee-8e0f-79c7682223dd" policytype="policy" attack="Eicar.Virus.Test.File"
srcport=37788 dstport=443 hostname="172.16.200.52" url="/eicar.com" agent="curl/7.61.1"
httpmethod="GET" direction="incoming" attackid=29844 profile="monitor-eicar"
ref="http://www.fortinet.com/ids/VID29844" incidentserialno=75497478 msg="file_transfer:
Eicar.Virus.Test.File" rawdataid="1/1" forwardedfor="10.22.22.22" rawdata="Response-
Content-Type=application/x-msdos-program|X-Forwarded-For=10.22.22.22" crscore=5
craction=65536 crlevel="low"
```

Zero Trust Network Access

This section includes information about ZTNA related new features:

- Tags and EMS connectors on page 354
- ZTNA policies on page 360

General

This section includes information about general ZTNA related new features:

- Introduce new ZTNA replacement message types 7.4.1 on page 332
- Condense ZTNA server mapping configurations 7.4.2 on page 346
- Introduce Fabric integration with FortiGSLB 7.4.2 on page 350

Introduce new ZTNA replacement message types - 7.4.1

This information is also available in the FortiOS 7.4 ZTNA Reference Guide:

- Error codes and replacement messages

Four new categories and 14 subtypes of ZTNA replacement messages have been added that correspond to new error codes error messages. Additional information is displayed for specific errors, and provides end users with more information about the error encountered.

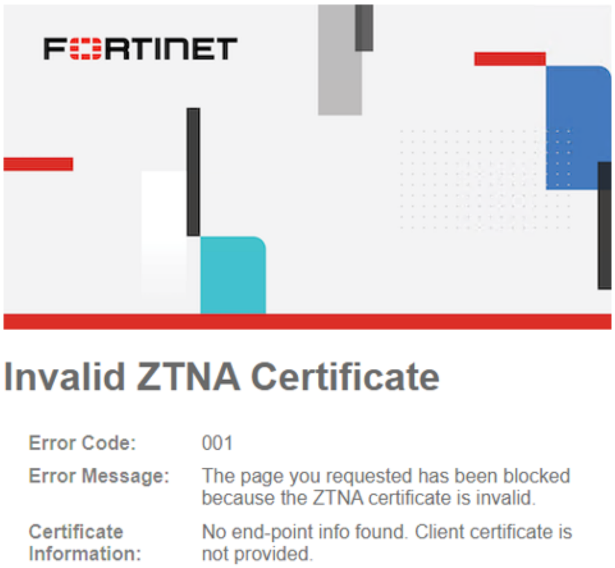
The new ZTNA replacement message categories and error subtypes are as follows:

- Invalid ZTNA Certificate
  - 001: the ZTNA certificate is invalid
  - 002: the ZTNA certificate is empty
  - 003: the device is manageable but with an empty ZTNA certificate
- ZTNA Application Not Found
  - 021: no API gateway was matched
  - 022: the real server in the API gateway cannot be found
  - 023: ZTNA FQDN DNS failed
- ZTNA Portal Error
  - 041: SSL VPN bookmark address failed
- ZTNA Policy Deny
  - 061: no policy was matched
  - 062: a policy with action deny was matched
  - 063: the client certificate is revoked
  - 064: denied by matched tags
  - 065: denied by no matched tags

- 066: no device information
- 067: the device is offline

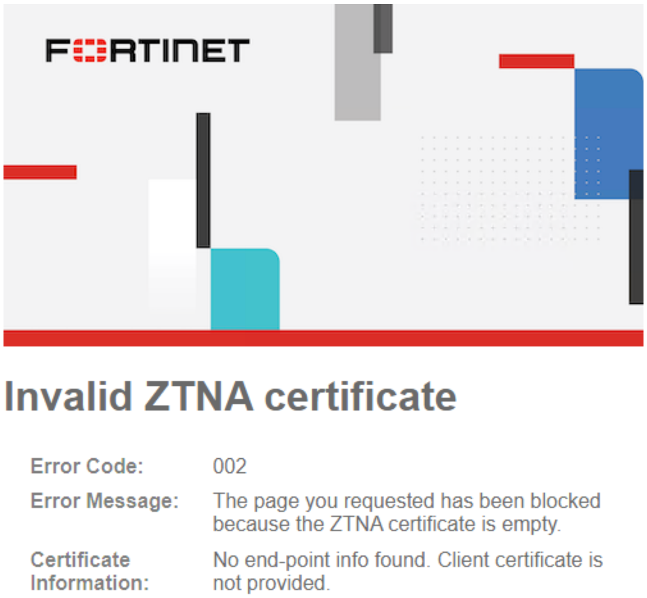
Example replacement messages and ZTNA traffic logs by error subtype

001: the ZTNA certificate is invalid



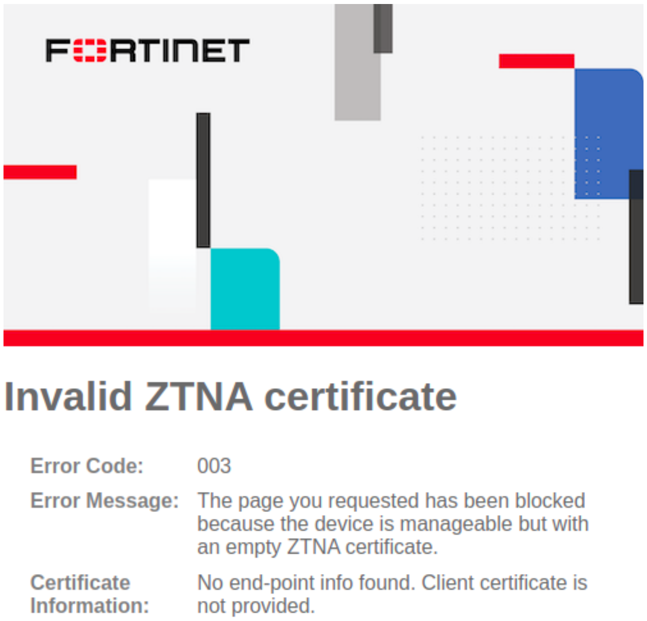

```
date=2023-06-29 time=18:35:51 id=7250286837292335106 itime="2023-06-29 18:35:51" euid=1029
epid=101 dsteuid=3 dstepid=101 logflag=419 logver=704012411 type="traffic" subtype="ztna"
level="notice" action="deny" policyid=7 sessionid=41387 srcip=21.21.21.120
dstip=172.18.62.32 srcport=49269 dstport=443 duration=0 proto=6 sentbyte=2363 rcvdbyte=0
logid=0005000024 unauthuser="frank" srcname="PC120" service="HTTPS" app="HTTPS"
appcat="unscanned" srcintfrole="undefined" dstintfrole="undefined" policytype="policy"
eventtime=1688088950860033973 wanin=0 wanout=0 lanin=2363 lanout=2052 crscore=30
craction=131072 crlevel="high" poluuid="9d55f2c6-0649-51ee-b2cc-94e51f44998d"
srccountry="United States" dstcountry="Reserved" srcintf="port2" dstintf="root"
unauthusersource="forticlient" policyname="ZTNA_policy_01" msg="Traffic denied because of
cert auth failed, cert-cn:6CB4E52E85BE45E8A9ADDE54E89A6B38, cert-issuer:FCTEMS8822002070,
cert-status:untrusted  fail-reason:certificate signature failure" threatwgts=30 threatcnts=1
threatlvls=3 threats=blocked-connection threattyps=blocked-connection tz="-0700" vip="ZTNA_
S1" accessproxy="ZTNA_S1" proxyapptype="http" clientdevicemanageable="unknown"
devid="FGVM32TM22000588" vd="root" dtime="2023-06-29 18:35:51" itime_t=1688088951
devname="EC_VM64_474"
```

002: the ZTNA certificate is empty




```
date=2023-06-12 time=14:50:30 id=7243920325665095680 itime="2023-06-12 14:50:32" euid=1029
epid=101 dsteuid=3 dstepid=101 logflag=419 logver=704012394 type="traffic" subtype="ztna"
level="notice" action="deny" policyid=7 sessionid=508001 srcip=21.21.21.120
dstip=172.18.62.32 srcport=58225 dstport=443 duration=0 proto=6 sentbyte=589 rcvdbyte=0
logid=0005000024 unauthuser="frank" srcname="PC120" service="HTTPS" app="HTTPS"
appcat="unscanned" srcintfrole="undefined" dstintfrole="undefined" policytype="policy"
eventtime=1686606630292857376 wanin=0 wanout=0 lanin=589 lanout=2052 crscore=30
craction=131072 crlevel="high" poluuid="9d55f2c6-0649-51ee-b2cc-94e51f44998d"
srccountry="United States" dstcountry="Reserved" srcintf="port2" dstintf="root"
unauthusersource="forticlient" policyname="ZTNA_policy_01" msg="Traffic denied because of
empty client certificate" threatwgts=30 threatcnts=1 threatlvls=3 threats=blocked-connection
threattyps=blocked-connection tz="-0700" vip="ZTNA_S1" accessproxy="ZTNA_S1"
proxyapptype="http" clientdevicemanageable="unknown" devid="FGVM32TM22000588" vd="root"
dtime="2023-06-12 14:50:30" itime_t=1686606632 devname="EC_VM64_474"
```

003: the device is manageable but with an empty ZTNA certificate




```
date=2023-06-29 time=18:50:56 id=7250290732827672576 itime="2023-06-29 18:50:58" euid=1029
epid=101 dsteuid=3 dstepid=101 logflag=419 logver=704012411 type="traffic" subtype="ztna"
level="notice" action="deny" policyid=7 sessionid=45501 srcip=21.21.21.120
dstip=172.18.62.32 srcport=49967 dstport=443 duration=0 proto=6 sentbyte=1151 rcvdbyte=0
logid=0005000024 unauthuser="frank" srcname="PC120" service="HTTPS" app="HTTPS"
appcat="unscanned" srcintfrole="undefined" dstintfrole="undefined" policytype="policy"
eventtime=1688089856235780791 wanin=0 wanout=0 lanin=1151 lanout=2273 crscore=30
craction=131072 crlevel="high" poluuid="9d55f2c6-0649-51ee-b2cc-94e51f44998d"
srccountry="United States" dstcountry="Reserved" srcintf="port2" dstintf="root"
unauthusersource="forticlient" policyname="ZTNA_policy_01" msg="Traffic denied because of
empty client certificate" threatwgts=30 threatcnts=1 threatlvls=3 threats=blocked-connection
threattyps=blocked-connection tz="-0700" vip="ZTNA_S1" accessproxy="ZTNA_S1"
proxyapptype="http" clientdevicemanageable="manageable" devid="FGVM32TM22000588" vd="root"
dtime="2023-06-29 18:50:56" itime_t=1688089858 devname="EC_VM64_474"
```

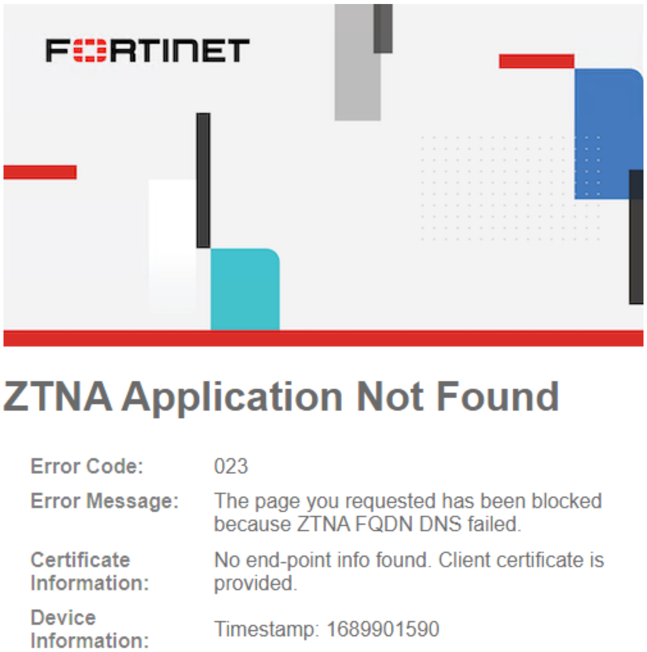
021: no API gateway was matched


```
date=2023-06-09 time=13:20:27 id=7242783860138704910 itime="2023-06-09 13:20:28" euid=1029
epid=1035 dsteuid=3 dstepid=101 logflag=419 logver=704012394 type="traffic" subtype="ztna"
level="notice" action="deny" policyid=7 sessionid=2290 srcip=21.21.21.120 dstip=172.18.62.25
srcport=50985 dstport=443 duration=66 proto=6 sentbyte=4470 rcvdbyte=4786 logid=0005000024
unauthuser="frank" srcname="PC120" service="HTTPS" app="HTTPS" appcat="unscanned"
fctuid="6CB4E52E85BE45E8A9ADDE54E89A6B38" srcintfrole="undefined" dstintfrole="undefined"
policytype="policy" eventtime=1686342027825210329 wanin=4786 wanout=4598 lanin=4470
lanout=395434 crscore=30 craction=131072 crlevel="high" poluuid="9d55f2c6-0649-51ee-b2cc-
94e51f44998d" srccountry="United States" dstcountry="Reserved" srcintf="port2"
dstintf="port3" unauthusersource="forticlient" policyname="ZTNA_policy_01" msg="Traffic
denied because of HTTP url (https://v2.qa.fortinet.com/favicon.ico) failed to match an API-
gateway with vhost(name/hostname:auto-ZTNA_S1-0/v2.qa.fortinet.com)" threatwgts=30
threatcnts=1 threatlvls=3 threats=blocked-connection threattyps=blocked-connection tz="-
0700" vip="ZTNA_S1" accessproxy="ZTNA_S1" gatewayid=5
clientdeviceid="6CB4E52E85BE45E8A9ADDE54E89A6B38" clientdevicetags="EMS4_ZTNA_ems133_
vulnerability_tag/EMS4_ZTNA_ems133_win_tag" proxyapptype="http"
clientdevicemanageable="manageable" emsconnection="online" devid="FGVM32TM22000588"
vd="root" dtime="2023-06-09 13:20:27" itime_t=1686342028 devname="EC_VM64_474"
```

022: the real server in the API gateway cannot be found

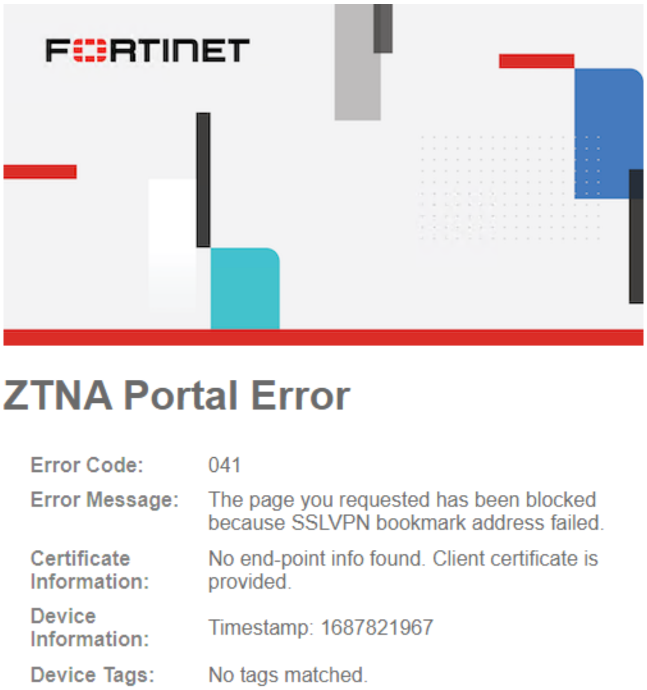

```
date=2023-06-29 time=18:25:02 id=7250284049858560001 itime="2023-06-29 18:25:02" euid=1029
epid=101 dsteuid=3 dstepid=101 logflag=419 logver=704012411 type="traffic" subtype="ztna"
level="notice" action="deny" policyid=7 sessionid=37591 srcip=21.21.21.120
dstip=172.18.62.32 srcport=65051 dstport=443 duration=0 proto=6 sentbyte=2395 rcvdbyte=0
logid=0005000024 unauthuser="frank" srcname="PC120" service="HTTPS" app="HTTPS"
appcat="unscanned" srcintfrole="undefined" dstintfrole="undefined" policytype="policy"
eventtime=1688088301509031772 wanin=0 wanout=0 lanin=2395 lanout=2355 crscore=30
craction=131072 crlevel="high" poluuid="9d55f2c6-0649-51ee-b2cc-94e51f44998d"
srccountry="United States" dstcountry="Reserved" srcintf="port2" dstintf="root"
unauthusersource="forticlient" policyname="ZTNA_policy_01" msg="Traffic denied because of
failed to find a server: reason: Cannot find the real server in the API gateway., hostname:
172.18.62.32" threatwgts=30 threatcnts=1 threatlvls=3 threats=blocked-connection
threattyps=blocked-connection tz="-0700" vip="ZTNA_S1" accessproxy="ZTNA_S1"
proxyapptype="http" clientdevicemanageable="manageable" devid="FGVM32TM22000588" vd="root"
dtime="2023-06-29 18:25:02" itime_t=1688088302 devname="EC_VM64_474"
```

023: ZTNA FQDN DNS failed



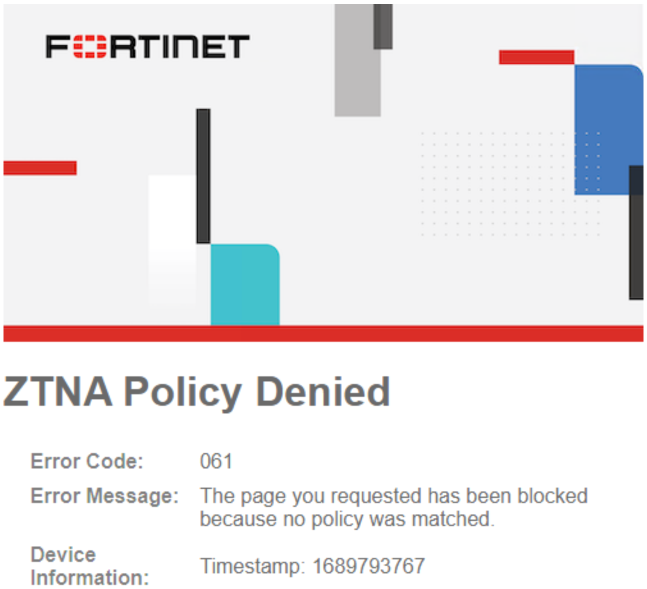

```
date=2023-06-19 time=16:34:17 id=7246544662352101380 itime="2023-06-19 16:34:18" euid=1029
epid=101 dsteuid=3 dstepid=101 logflag=419 logver=704012394 type="traffic" subtype="ztna"
level="notice" action="deny" policyid=7 sessionid=36913 srcip=21.21.21.120
dstip=172.18.62.32 srcport=56222 dstport=443 duration=0 proto=6 sentbyte=2667 rcvdbyte=0
logid=0005000024 unauthuser="frank" srcname="PC120" service="HTTPS" app="HTTPS"
appcat="unscanned" srcintfrole="undefined" dstintfrole="undefined" policytype="policy"
eventtime=1687217657552817038 wanin=0 wanout=0 lanin=2667 lanout=37910 crscore=30
craction=131072 crlevel="high" poluuid="9d55f2c6-0649-51ee-b2cc-94e51f44998d"
srccountry="United States" dstcountry="Reserved" srcintf="port2" dstintf="root"
unauthusersource="forticlient" policyname="ZTNA_policy_01" msg="Traffic denied because of
HTTP url (https://webportal.fortinet.com/favicon.ico) failed to match an API-gateway with
vhost(name/hostname:webportal/webportal.fortinet.com)" threatwgts=30 threatcnts=1
threatlvls=3 threats=blocked-connection threattyps=blocked-connection tz="-0700" vip="ZTNA_
S1" accessproxy="ZTNA_S1" proxyapptype="http" clientdevicemanageable="manageable"
devid="FGVM32TM22000588" vd="root" csf="EC_VM09_csf_root" dtime="2023-06-19 16:34:17" itime_
t=1687217658 devname="EC_VM64_474"
```

041: SSL VPN bookmark address failed



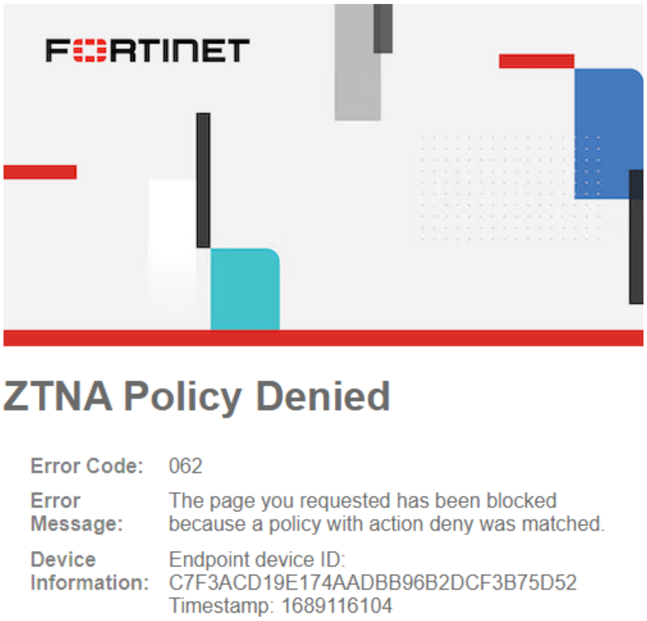

```
date=2023-06-26 time=16:26:12 id=7249140175513583617 itime="2023-06-26 16:26:13" euid=1030
epid=101 dsteuid=3 dstepid=101 logflag=419 logver=704012409 type="traffic" subtype="ztna"
level="notice" action="deny" policyid=7 sessionid=21602 srcip=21.21.21.119
dstip=172.18.62.32 srcport=59498 dstport=443 duration=5 proto=6 sentbyte=4829 rcvdbyte=0
logid=0005000024 unauthuser="fosqa" srcname="DESKTOP-TDD7MND" service="HTTPS" app="HTTPS"
appcat="unscanned" srcintfrole="undefined" dstintfrole="undefined" policytype="policy"
eventtime=1687821972368162788 wanin=0 wanout=0 lanin=4829 lanout=277700 crscore=30
craction=131072 crlevel="high" poluuid="9d55f2c6-0649-51ee-b2cc-94e51f44998d"
srccountry="United States" dstcountry="Reserved" srcintf="port2" dstintf="root"
unauthusersource="forticlient" policyname="ZTNA_policy_01" msg="Traffic denied because of
failed to find a server: reason: SSLVPN Bookmark Address Failed., hostname:
webportal.fortinet.com" threatwgts=30 threatcnts=1 threatlvls=3 threats=blocked-connection
threattyps=blocked-connection tz="-0700" vip="ZTNA_S1" accessproxy="ZTNA_S1" gatewayid=99
proxyapptype="http" clientdevicemanageable="manageable" devid="FGVM32TM22000588" vd="root"
dtime="2023-06-26 16:26:12" itime_t=1687821973 devname="EC_VM64_474"
```

061: no policy was matched



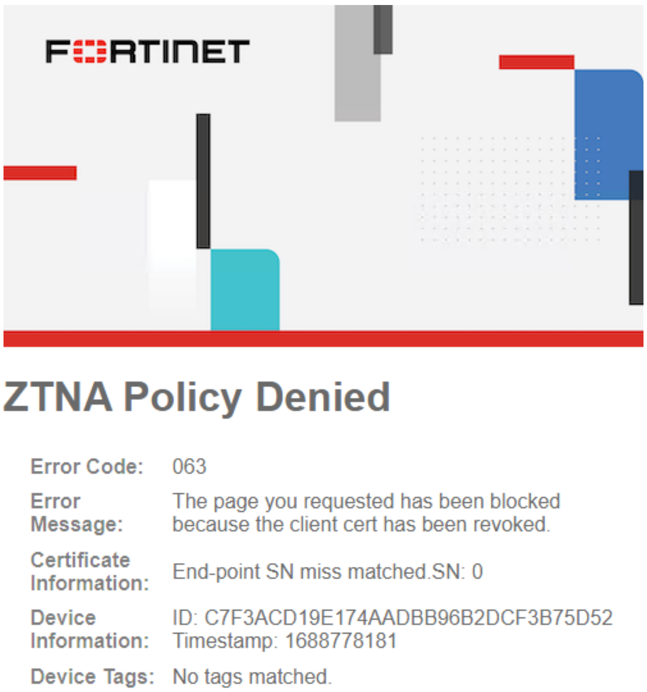

```
date=2023-07-19 time=12:09:35 id=7257609004912738305 itime="2023-07-19 12:09:36" euid=1038
epid=101 dsteuid=3 dstepid=101 logflag=419 logver=704012423 type="traffic" subtype="ztna"
level="notice" action="deny" policyid=0 sessionid=244448 srcip=21.21.21.119
dstip=172.18.62.32 srcport=49820 dstport=443 duration=9 proto=6 sentbyte=2761 rcvdbyte=0
logid=0005000024 user="test2" unauthuser="fosqa" srcname="DESKTOP-TDD7MND" service="HTTPS"
app="HTTPS" appcat="unscanned" srcintfrole="undefined" dstintfrole="undefined"
policytype="proxy-policy" eventtime=1689793776343168805 wanin=0 wanout=0 lanin=2761
lanout=72679 crscore=30 craction=131072 crlevel="high" srccountry="United States"
dstcountry="Reserved" srcintf="port2" dstintf="root" unauthusersource="forticlient"
authserver="radius_server" msg="Traffic denied because failed to match a proxy-policy"
threatwgts=30 threatcnts=1 threatlvls=3 threats=blocked-connection threattyps=blocked-
connection tz="-0700" vip="ZTNA_S1" accessproxy="ZTNA_S1" proxyapptype="http"
clientdevicemanageable="manageable" devid="FGVM32TM22000588" vd="root" dtime="2023-07-19
12:09:35" itime_t=1689793776 devname="EC_VM64_474"
```

062: a policy with action deny was matched



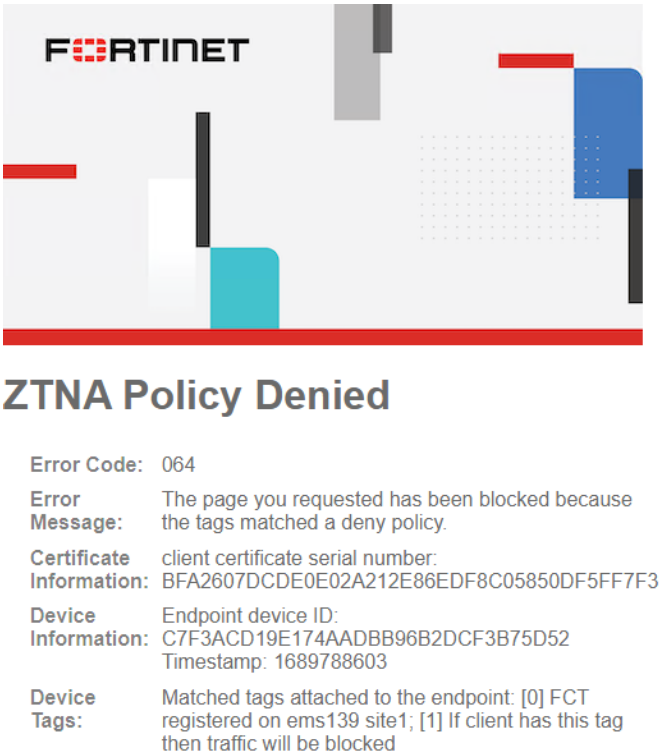

```
date=2023-07-20 time=23:38:32 id=7258157648330096641 itime="2023-07-20 23:38:37" euid=3
epid=101 dsteuid=3 dstepid=101 logflag=3 logver=704012423 type="traffic" subtype="ztna"
level="notice" action="deny" policyid=10 sessionid=333849 srcip=21.21.21.119
dstip=172.18.62.32 srcport=56183 dstport=443 duration=0 proto=6 sentbyte=2630 rcvdbyte=0
logid=0005000024 service="HTTPS" app="HTTPS" appcat="unscanned" srcintfrole="undefined"
dstintfrole="undefined" policytype="policy" eventtime=1689921512660140357 wanin=0 wanout=0
lanin=2630 lanout=37508 crscore=30 craction=131072 crlevel="high" poluuid="d98092d4-2038-
51ee-aa34-c5ab997e596f" srccountry="United States" dstcountry="Reserved" srcintf="port2"
dstintf="root" policyname="ZTNA_deny_policy_specific_host" msg="Traffic denied because
proxy-policy action is deny." threatwgts=30 threatcnts=1 threatlvls=3 threats=blocked-
connection threattyps=blocked-connection tz="-0700" vip="ZTNA_S1" accessproxy="ZTNA_S1"
proxyapptype="http" clientdevicemanageable="manageable" devid="FGVM32TM22000588" vd="root"
dtime="2023-07-20 23:38:32" itime_t=1689921517 devname="EC_VM64_474"
```

063: the client certificate is revoked



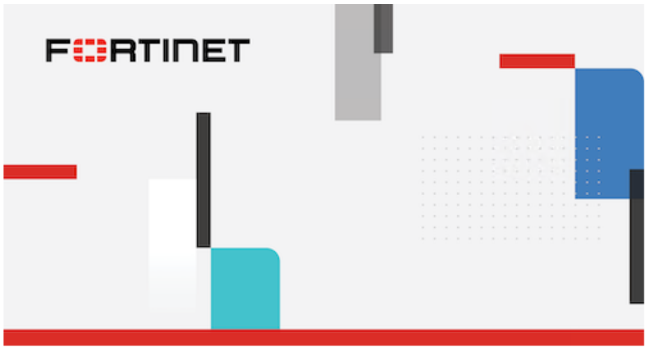

```
date=2023-07-20 time=23:13:39 id=7258151227353989121 itime="2023-07-20 23:13:42" euid=3
epid=1045 dsteuid=3 dstepid=101 logflag=3 logver=704012423 type="traffic" subtype="ztna"
level="notice" action="deny" policyid=0 sessionid=326920 srcip=21.21.21.33
dstip=172.18.62.32 srcport=41684 dstport=443 duration=1 proto=6 sentbyte=1934 rcvdbyte=0
logid=0005000024 service="HTTPS" app="HTTPS" appcat="unscanned"
fctuid="A9DB1F65BC1A218B00234A2481290696" srcintfrole="undefined" dstintfrole="undefined"
policytype="policy" eventtime=1689920018691818431 wanin=0 wanout=0 lanin=1934 lanout=2081
crscore=30 craction=131072 crlevel="high" srccountry="United States" dstcountry="Reserved"
srcintf="port2" dstintf="root" msg="Traffic denied because client cert is revoked."
threatwgts=30 threatcnts=1 threatlvls=3 threats=blocked-connection threattyps=blocked-
connection tz="-0700" vip="ZTNA_S1" accessproxy="ZTNA_S1"
clientdeviceid="A9DB1F65BC1A218B00234A2481290696" clientdevicetags="EMS4_ZTNA_ems133_
vulnerability_tag/EMS4_ZTNA_ems133_win_tag" proxyapptype="http"
clientdevicemanageable="manageable" emsconnection="online" devid="FGVM32TM22000588"
vd="root" dtime="2023-07-20 23:13:39" itime_t=1689920022 devname="EC_VM64_474"
```

064: denied by matched tags




```
date=2023-07-20 time=23:42:37 id=7258158679122247683 itime="2023-07-20 23:42:37" euid=1030
epid=1039 dsteuid=3 dstepid=101 logflag=3 logver=704012423 type="traffic" subtype="ztna"
level="notice" action="deny" policyid=10 sessionid=335062 srcip=21.21.21.119
dstip=172.18.62.32 srcport=56407 dstport=443 duration=0 proto=6 sentbyte=2630 rcvdbyte=0
logid=0005000024 unauthuser="fosqa" srcname="DESKTOP-TDD7MND" service="HTTPS" app="HTTPS"
appcat="unscanned" fctuid="C7F3ACD19E174AADBB96B2DCF3B75D52" srcintfrole="undefined"
dstintfrole="undefined" policytype="policy" eventtime=1689921757387765903 wanin=0 wanout=0
lanin=2630 lanout=37927 crscore=30 craction=131072 crlevel="high" poluuid="d98092d4-2038-
51ee-aa34-c5ab997e596f" srccountry="United States" dstcountry="Reserved" srcintf="port2"
dstintf="root" unauthusersource="forticlient" policyname="ZTNA_deny_policy_specific_host"
msg="Traffic denied because proxy-policy action is deny. Matched tag: EMS4_ZTNA_ems133_
management_tag" threatwgts=30 threatcnts=1 threatlvls=3 threats=blocked-connection
threattyps=blocked-connection tz="-0700" vip="ZTNA_S1" accessproxy="ZTNA_S1"
clientdeviceid="C7F3ACD19E174AADBB96B2DCF3B75D52" clientdevicetags="EMS4_ZTNA_ems133_
vulnerability_tag/EMS4_ZTNA_ems133_win_tag" proxyapptype="http"
clientdevicemanageable="manageable" emsconnection="online" devid="FGVM32TM22000588"
vd="root" dtime="2023-07-20 23:42:37" itime_t=1689921757 devname="EC_VM64_474"
```

065: denied by no matched tags

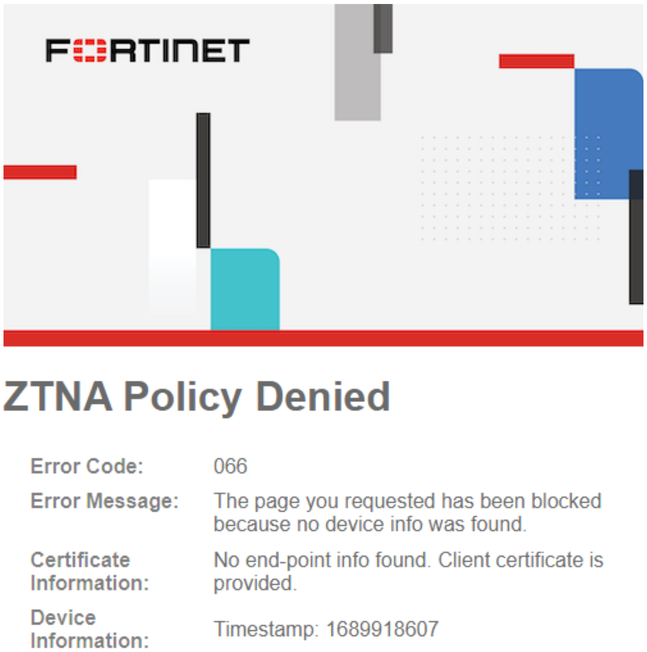


ZTNA Policy Denied

Error Code:	065
Error Message:	The page you requested has been blocked because the tags didn't match any policy.
Certificate Information:	client certificate serial number: BFA2607DCDE0E02A212E86EDF8C05850DF5FF7F3
Device Information:	Endpoint device ID: C7F3ACD19E174AADBB96B2DCF3B75D52 Timestamp: 1689787657
Device Tags:	All tags attached to the endpoint: [0] ; [1] FCT registered on ems139 site1; [2] If client has this tag then traffic will be blocked

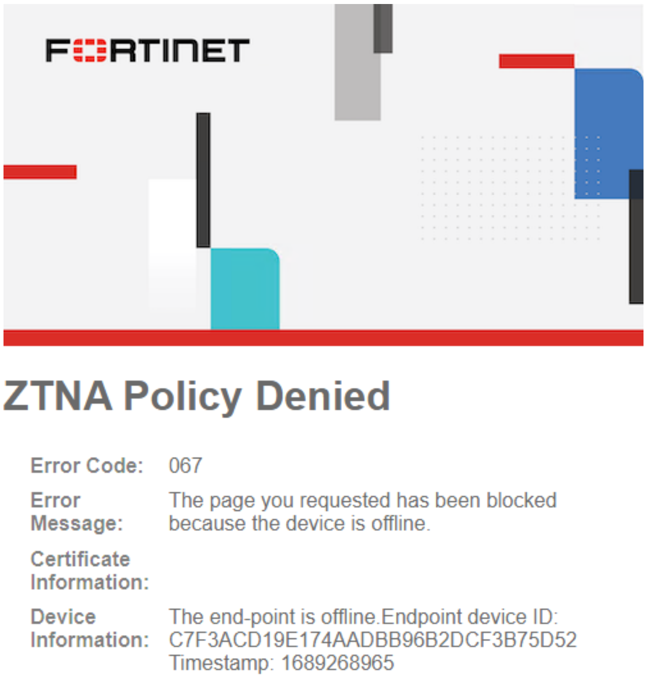
```
date=2023-07-20 time=23:26:10 id=7258154439989526534 itime="2023-07-20 23:26:10" euid=1030
epid=1039 dsteuid=3 dstepid=101 logflag=3 logver=704012423 type="traffic" subtype="ztna"
level="notice" action="deny" policyid=0 sessionid=330766 srcip=21.21.21.119
dstip=172.18.62.32 srcport=55758 dstport=443 duration=0 proto=6 sentbyte=2630 rcvdbyte=0
logid=0005000024 unauthuser="fosqa" srcname="DESKTOP-TDD7MND" service="HTTPS" app="HTTPS"
appcat="unscanned" fctuid="C7F3ACD19E174AADBB96B2DCF3B75D52" srcintfrole="undefined"
dstintfrole="undefined" policytype="policy" eventtime=1689920770603617305 wanin=0 wanout=0
lanin=2630 lanout=37982 crscore=30 craction=131072 crlevel="high" srccountry="United States"
dstcountry="Reserved" srcintf="port2" dstintf="root" unauthusersource="forticlient"
msg="Traffic denied because failed to match a proxy-policy" threatwgts=30 threatcnts=1
threatlvls=3 threats=blocked-connection threattyps=blocked-connection tz="-0700" vip="ZTNA_
S1" accessproxy="ZTNA_S1" clientdeviceid="C7F3ACD19E174AADBB96B2DCF3B75D52"
clientdevicetags="EMS5_ZTNA_all_registered_clients/EMS5_ZTNA_site1_ems139_management_tag"
proxyapptype="http" clientdevicemanageable="manageable" emsconnection="online"
devid="FGVM32TM22000588" vd="root" dtime="2023-07-20 23:26:10" itime_t=1689920770
devname="EC_VM64_474"
```

066: no device information




```
date=2023-07-20 time=22:50:46 id=7258145317478989831 itime="2023-07-20 22:50:46" euid=1029
epid=101 dsteuid=3 dstepid=101 logflag=3 logver=704012423 type="traffic" subtype="ztna"
level="notice" action="deny" policyid=0 sessionid=319630 srcip=21.21.21.120
dstip=172.18.62.32 srcport=59871 dstport=443 duration=38 proto=6 sentbyte=2849 rcvdbyte=0
logid=0005000024 unauthuser="frank" srcname="PC120" service="HTTPS" app="HTTPS"
appcat="unscanned" srcintfrole="undefined" dstintfrole="undefined" policytype="policy"
eventtime=1689918645481724057 wanin=0 wanout=0 lanin=2849 lanout=73142 crscore=30
craction=131072 crlevel="high" srccountry="United States" dstcountry="Reserved"
srcintf="port2" dstintf="root" unauthusersource="forticlient" msg="Traffic denied because
failed to match a proxy-policy" threatwgts=30 threatcnts=1 threatlvls=3 threats=blocked-
connection threattyps=blocked-connection tz="-0700" vip="ZTNA_S1" accessproxy="ZTNA_S1"
proxyapptype="http" clientdevicemanageable="manageable" devid="FGVM32TM22000588" vd="root"
dtime="2023-07-20 22:50:46" itime_t=1689918646 devname="EC_VM64_474"
```

067: the device is offline




```
date=2023-07-20 time=23:33:25 id=7258156308300300291 itime="2023-07-20 23:33:25" euid=1030
epid=1039 dsteuid=3 dstepid=101 logflag=3 logver=704012423 type="traffic" subtype="ztna"
level="notice" action="deny" policyid=0 sessionid=332783 srcip=21.21.21.119
dstip=172.18.62.32 srcport=56081 dstport=443 duration=0 proto=6 sentbyte=2630 rcvdbyte=0
logid=0005000024 service="HTTPS" app="HTTPS" appcat="unscanned"
fctuid="C7F3ACD19E174AADBB96B2DCF3B75D52" srcintfrole="undefined" dstintfrole="undefined"
policytype="policy" eventtime=1689921205206083026 wanin=0 wanout=0 lanin=2630 lanout=37700
crscore=30 craction=131072 crlevel="high" srccountry="United States" dstcountry="Reserved"
srcintf="port2" dstintf="root" msg="Traffic denied because failed to match a proxy-policy"
threatwgts=30 threatcnts=1 threatlvls=3 threats=blocked-connection threattyps=blocked-
connection tz="-0700" vip="ZTNA_S1" accessproxy="ZTNA_S1"
clientdeviceid="C7F3ACD19E174AADBB96B2DCF3B75D52" proxyapptype="http"
clientdevicemanageable="manageable" emsconnection="offline" devid="FGVM32TM22000588"
vd="root" dtime="2023-07-20 23:33:25" itime_t=1689921205 devname="EC_VM64_474"
```

Condense ZTNA server mapping configurations - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- Basic ZTNA configuration

To reduce the number of clicks to configure a ZTNA server object, the settings to create a new *Server/service mapping* are condensed. Real server mappings can be configured directly in the *Service/Server Mapping* pane. To display additional real servers or load balancing options in the GUI, first create a second real server in the CLI.

Example

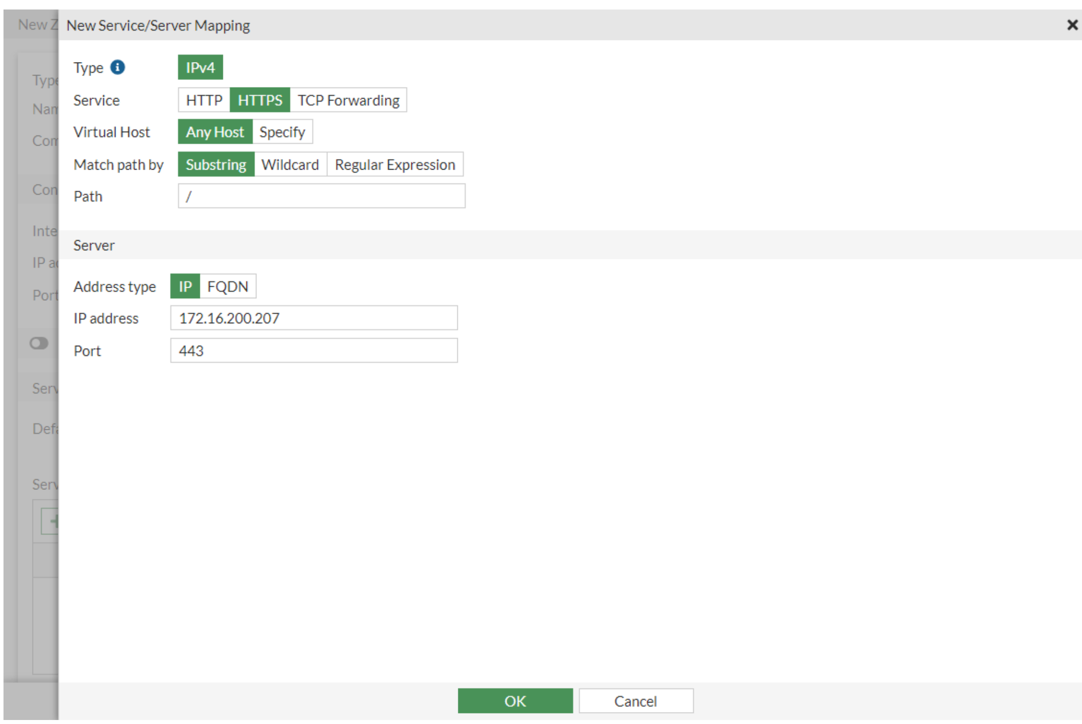
In this example, a ZTNA server with HTTPS and TCP forwarding mapping is created in the GUI. A second set of HTTPS and TCP forwarding mapping is added in the CLI. After the second server mappings are added in the CLI, additional options are available in the GUI for load balancing and adding more real servers.

To configure a ZTNA server with HTTPS and TCP forwarding mapping:

1. Go to *Policy & Objects > ZTNA*, select the *ZTNA Servers* tab, and click *Create new*.
2. Enter a name for the server, such as *new_ztna*.
3. Select an *Interface*. The *IP address* and *Port* fields are automatically filled in based on the interface selection.

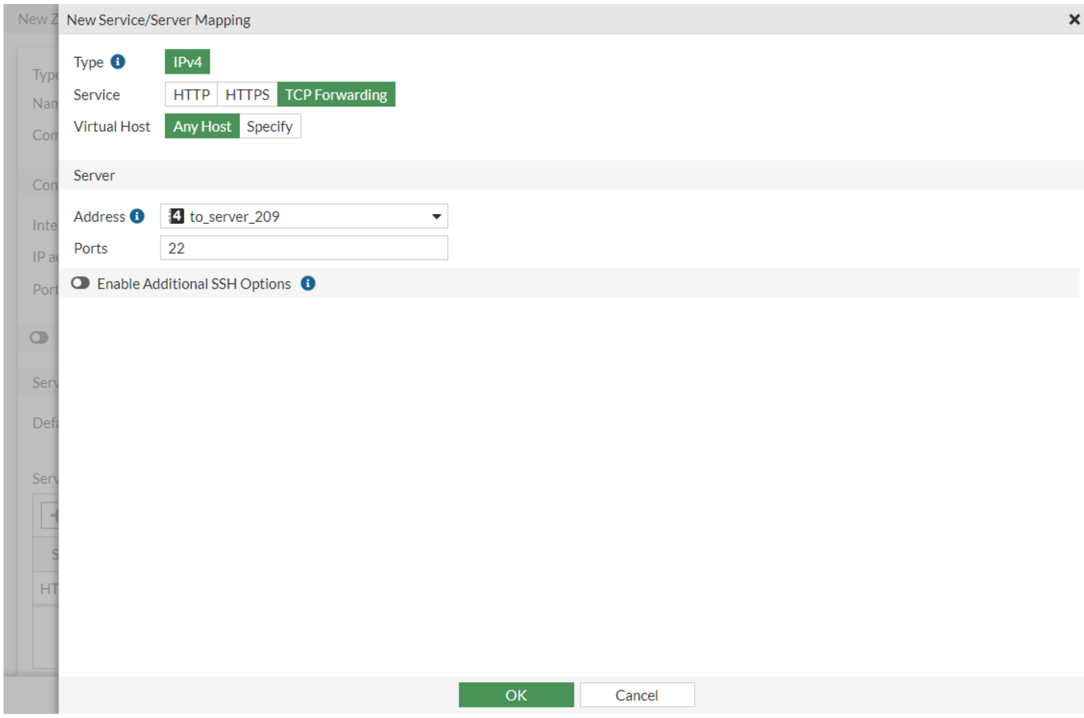
   Verify that the IP address and port does not conflict with management access to the interface. Otherwise, change the IP address to another address on that subnet.

4. Select the *Default certificate*.
5. Add the HTTPS mapping:
   a. In the *Service/server mapping* table, click *Create New*.
   b. Set *Service* to *HTTPS*.
   c. Configure the settings for the *Virtual Host*, *Match path by*, and *Path* fields.
   d. In the *Server* section, set the *Address type* to *IP*.
   e. Set the *IP address* to *172.16.200.207*.
   f. Set the *Port* to *443*.



   g. Click *OK*.
6. Add the TCP forwarding mapping:

a. In the *Service/server mapping* table, click *Create New*.
b. Set *Service* to *TCP Forwarding*.
c. Configure the setting for the *Virtual Host*.
d. In the *Server* section, set the *Address* to *to_server_209*.
e. Set the *Ports* to *22*.



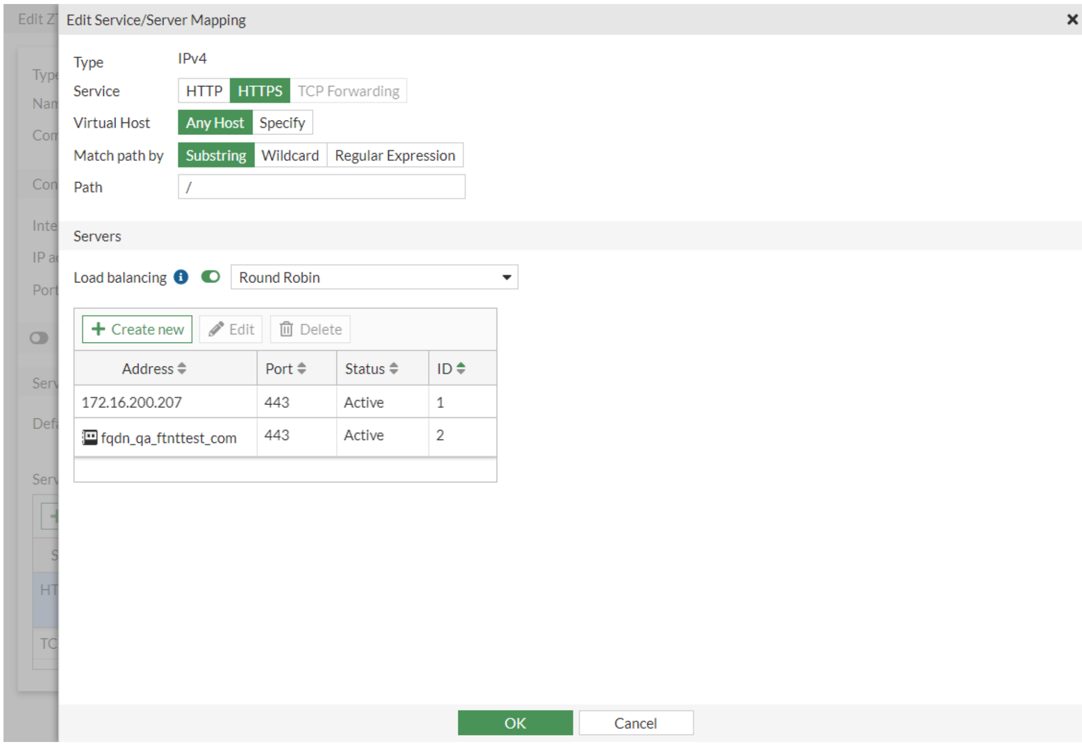
f. Click *OK*.

7. Click *OK*.
8. Configure the second servers with HTTPS and TCP forwarding mapping in the CLI:

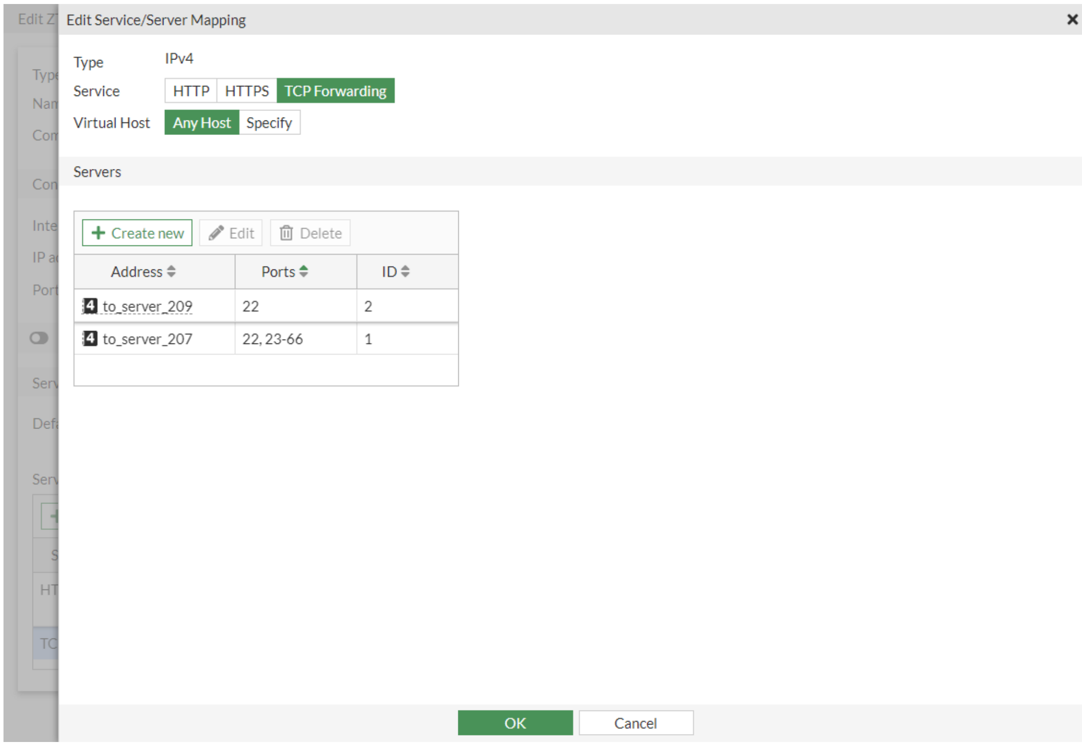
```
config firewall access-proxy
    edit "new_ztna"
        config api-gateway
            edit 1
                config realservers
                    edit 2
                        set addr-type fqdn
                        set address "fqdn_qa_ftnttest_com"
                    next
                end
                set ldb-method round-robin
            next
            edit 2
                config realservers
                    edit 1
                        set address "to_server_207"
                        set domain "server.209"
                        set mappedport 22 23-66
                    next
                end
            next
```

```
        end
    next
end
```

9. In the GUI, edit the *new_ztna* server and verify the current server mapping:
   a. In the *Service/server mapping* table, edit the *172.16.200.207* entry. The *Load balancing* option is visible, and additional real servers can be added by clicking *Create new*.



   b. Click *Cancel*.
   c. Edit the TCP forwarding entry. Additional real servers can be added by clicking *Create New*.



d. Click *Cancel*.

Automatic pre-fill when creating a new service/server mapping

When adding a new mapped server, the server settings are automatically filled in with the previous HTTPS server settings. Verify that the IP address of the mapped server is configured as intended. The automatic filling helps simplify the configuration when trying to create different mappings to the same server with different paths. But, when mapping to different servers, the *IP address* and/or *Port* should be changed.

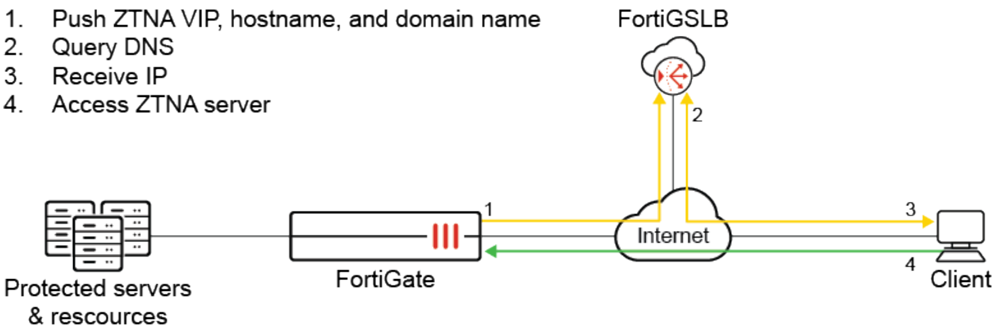
Introduce Fabric integration with FortiGSLB - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

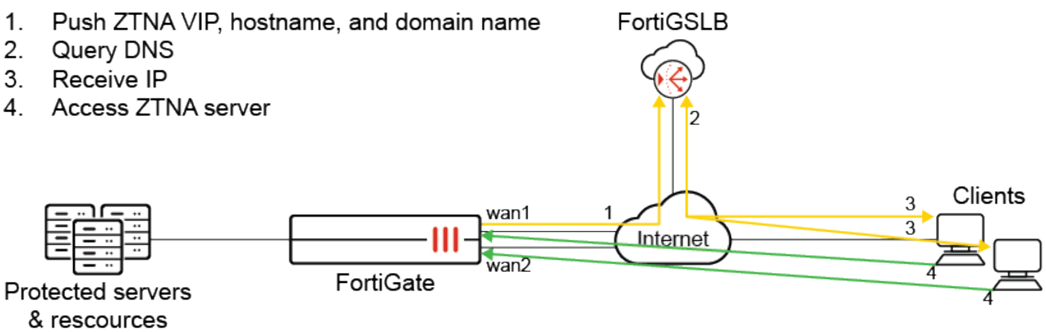
- Fabric integration with FortiGSLB

Fabric integration between the FortiGate and FortiGSLB allows a FortiGate to publish custom host and domain names directly to FortiGSLB. This enables external IPs on VIPs used in ZTNA server objects to be published with the host and domain names directly to FortiGSLB, where its DNS service can provide nameserver lookups for the FQDNs.

In a basic use case, the hostname, domain name, and external IP of a ZTNA server can be published, and any subsequent updates to the address are immediately pushed to FortiGSLB.



In more advanced setup, an FQDN may map to different external IPs, which can be load balanced by FortiGSLB.



In addition, FortiGSLB can perform health checks on the external IPs, and then return the link with the better metrics. See How to add FortiGate SD-WAN Inbound Load Balancing to FortiGSLB in the FortiGSLB Handbook for more information.

This feature requires a valid FortiGSLB account contract (FGCS). If no valid FGCS contract is found, the CLI will return a warning message during configurations:

```
No license detected for FortiGSLB.
GSLB configuration and statistics will not be reported unless the
account is licensed.
```

To enable VIP and ZTNA server integration with the FortiGSLB Cloud service:

```
config system global
    set fortigslb-integration {enable | disable}
end
```

To configure the FortiGSLB setting in the VIP:

```
config firewall vip
    edit <name>
        set one-click-gslb-server {enable | disable}
        set gslb-hostname <string>
        set gslb-domain-name <string>
        config gslb-public-ips
            edit <id>
                set ip <IP_address>
            next
        end
```

```
    next
end
```

`one-click-gslb-server {enable | disable}`	Enable/disable integration with FortiGSLB.
`gslb-hostname <string>`	Enter the hostname portion of the FQDN that will be used within the configured FortiGSLB domain.
`gslb-domain-name <string>`	Enter the domain name of the FQDN that will be used within the configured FortiGSLB domain.
`ip <IP_address>`	Enter the custom publicly accessible IP address that overrides the external IP address (`extip`). This setting is optional.

Example

In this example, a FortiGate has three WAN interfaces, each configured with different VIPs that are used in ZTNA server objects that point to the same real server. These VIPs are configured with the same GSLB hostname and domain name. As a result, the hostname and domain name are mapped to three different addresses and sent to FortiGSLB. FortiGSLB's default setting will perform load balancing and respond to DNS queries by returning the addresses in a round-robin fashion.

To configure FortiGSLB integration:

1. Enable integration with FortiGSLB in the global settings:

```
config system global
    set fortigslb-integration enable
end
```

2. Enable integration with FortiGSLB on each firewall VIP:

```
config firewall vip
    edit "ztna_vip1"
        set type access-proxy
        set server-type https
        set extip 172.18.62.66
        set extintf "port2"
        set one-click-gslb-server enable
        set gslb-hostname "qa.test"
        set gslb-domain-name "wangd.com"
        set extport 4443
        set ssl-certificate "default.test.com"
    next
    edit "ztna_vip2"
        set type access-proxy
        set server-type https
        set extip 172.18.62.67
        set extintf "port3"
        set one-click-gslb-server enable
        set gslb-hostname "qa.test"
        set gslb-domain-name "wangd.com"
        set extport 4443
        set ssl-certificate "default.test.com"
```

```
        next
        edit "ztna_vip3"
            set type access-proxy
            set server-type https
            set extip 172.18.62.68
            set extintf "port4"
            set one-click-gslb-server enable
            set gslb-hostname "qa.test"
            set gslb-domain-name "wangd.com"
            config gslb-public-ips
                edit 1
                    set ip 172.18.62.69
                next
            end
            set extport 4443
            set ssl-certificate "default.test.com"
        next
    end
```

3. Enable debugs:

```
# diagnose debug application cloudapid -1
# diagnose debug enable
```

A successful connection will produce output similar to the following:

```
<4234> 10 cloudapi_curl_debug()-19: CURL HEADER OUT: POST /api/v1.0/one-click-glb-
fgt/modifyconfig HTTP/2
Host: 1clickfgt.fortigslb-cloud.com
Accept: application/json
Content-Type: application/json
Content-Length: 553

<4234> 10 cloudapi_curl_debug()-19: CURL DATA OUT: {"members":[{"vdom_
name":"vdom1","name_key":"ztna_vip1","type":"ztna","ip_list":
["172.18.62.66"],"host":"qa.test","domain":"wangd.com"},{"vdom_name":"vdom1","name_
key":"ztna_vip2","type":"ztna","ip_list":
["172.18.62.67"],"host":"qa.test","domain":"wangd.com"},{"vdom_name":"vdom1","name_
key":"ztna_vip3","type":"ztna","ip_list":
["172.18.62.69"],"host":"qa.test","domain":"wangd.com"}],"ha_cluster":
[{"sn":"FG181FTK22902632","host_name":"FGT1801F-ZTNA"},{"sn":"FG181FTK22902625","host_
name":"FGT1801F-ZTNA"}],"timestamp":"2023-11-23 00:28:43"}
```

Verification

Upon successfully passing the hostname, domain name, and IP address mappings to FortiGSLB, clients that are using FortiGSLB's DNS for DNS resolution can now get responses to their queries. Results on consecutive queries return the IP addresses in a round-robin fashion.

First query:

```
fosqa@ztna-client4:~/ztna_pytest$ dig @15.197.150.26 qa.test.wangd.com
; <<>> DiG 9.16.1-Ubuntu <<>> @15.197.150.26 qa.test.wangd.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33860
```

```
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;qa.test.wangd.com.             IN      A

;; ANSWER SECTION:
qa.test.wangd.com.      5       IN      A       172.18.62.66

;; AUTHORITY SECTION:
wangd.com.              86400   IN      NS      defaultprimary.wangd.com.

;; ADDITIONAL SECTION:
defaultprimary.wangd.com. 86400 IN      A       15.197.150.26

;; Query time: 15 msec
;; SERVER: 15.197.150.26#53(15.197.150.26)
;; WHEN: Thu Nov 16 10:56:23 PST 2023
;; MSG SIZE  rcvd: 107
```

Second query:

```
fosqa@ztna-client4:~/ztna_pytest$ dig @15.197.150.26 qa.test.wangd.com
; <<>> DiG 9.16.1-Ubuntu <<>> @15.197.150.26 qa.test.wangd.com
…
;; QUESTION SECTION:
;qa.test.wangd.com.             IN      A

;; ANSWER SECTION:
qa.test.wangd.com.      5       IN      A       172.18.62.69
…
```

Third query:

```
fosqa@ztna-client4:~/ztna_pytest$ dig @15.197.150.26 qa.test.wangd.com
; <<>> DiG 9.16.1-Ubuntu <<>> @15.197.150.26 qa.test.wangd.com
…
;; QUESTION SECTION:
;qa.test.wangd.com.             IN      A

;; ANSWER SECTION:
qa.test.wangd.com.      5       IN      A       172.18.62.67
…
```

Tags and EMS connectors

This section includes information about tag and EMS connector related new features:

- Support logical AND for tag matching between primary and secondary EMS tags in a firewall policy on page 355
- Support sending the FortiGate interface subnet list to EMS on page 357

- Add the Any and All options back for security posture tags in the GUI 7.4.2 on page 358
- Rename ZTNA Tag to Security Posture Tag in the GUI 7.4.2 on page 358

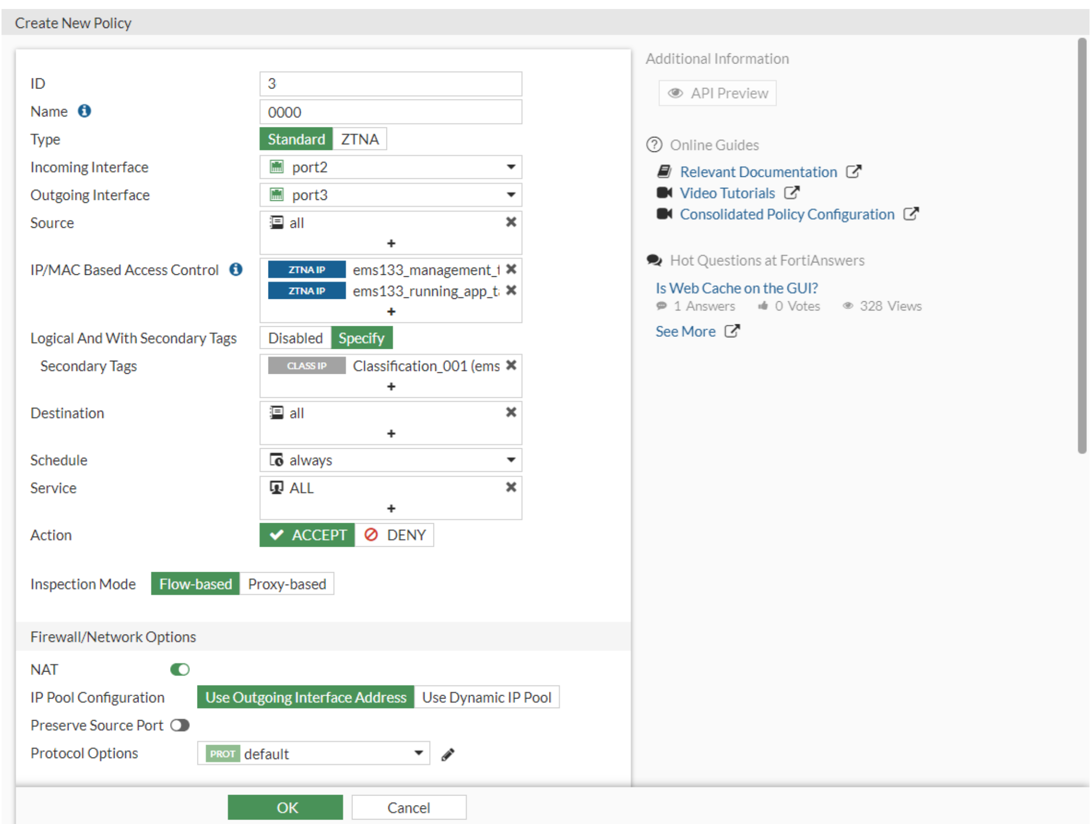
Support logical AND for tag matching between primary and secondary EMS tags in a firewall policy

When configuring a firewall policy for IP- or MAC-based access control that uses different EMS tag types (such as ZTNA tags and classification tags), a logical AND can be used for matching. By separating each tag type into primary and secondary groups, the disparate tag types will be matched with a logical AND operator.

In this example, IP-based access control is configured by allowing only clients that have the ems133_management_tag OR ems133_running_app_tag ZTNA tag, AND the CLASS_Classification_001 classification tag.

To configure logical AND tag matching in the GUI:

1. Go to *Policy & Objects > Firewall Policy*.
2. Create a new policy, or edit an existing one.
3. For *IP/MAC Based Access Control*, click the + and select the desired EMS tags (*ems133_management_tag* and *ems133_running_app_tag*).
4. Set *Logical And With Secondary Tags* to *Specify*, and click the + to add the secondary EMS tag (*CLASS_Classification_001*).



5. Configure the other settings as needed.
6. Click *OK*.

To configure logical AND tag matching in the CLI:

```
config firewall policy
    edit 3
        set name "0000"
        set srcintf "port2"
        set dstintf "port3"
        set action accept
        set ztna-status enable
        set srcaddr "all"
        set dstaddr "all"
        set ztna-ems-tag "EMS2_ZTNA_ems133_management_tag" "EMS2_ZTNA_ems133_running_app_
tag"
        set ztna-ems-tag-secondary "EMS2_CLASS_Classification_001"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

To verify the tag matching in the firewall policy:

```
# diagnose firewall iprope list 100004

policy index=2 uuid_idx=16180 action=accept
flag (8050108): redir nat master use_src pol_stats
flag2 (4000): resolve_sso
flag3 (a0): link-local best-route
schedule(always)
cos_fwd=255  cos_rev=255
group=00100004 av=00004e20 au=00000000 split=00000000
host=4 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 4 -> zone(1): 5
source(1): 0.0.0.0-255.255.255.255, uuid_idx=16088,
dest(2): 172.16.200.133-172.16.200.133, uuid_idx=16097, 172.17.254.148-172.17.254.148, uuid_
idx=16275,
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto

policy index=3 uuid_idx=16277 action=accept
flag (8050108): redir nat master use_src pol_stats
flag2 (4000): resolve_sso
flag3 (a0): link-local best-route
schedule(always)
cos_fwd=255  cos_rev=255
group=00100004 av=00004e20 au=00000000 split=00000000
host=4 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 4 -> zone(1): 5
source(1): 0.0.0.0-255.255.255.255, uuid_idx=16088,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=16088,
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto
ztna-ems-tag address (2):  uuid_idx=16118
        EMS2_ZTNA_ems133_running_app_tag ID(68) uuid_idx=16122
```

```
        EMS2_ZTNA_ems133_management_tag ID(122) ADDR(10.1.100.115) ADDR(10.1.100.117)
ztna-ems-tag-secondary address (1):  uuid_idx=16273
        EMS2_CLASS_Classification_001 ID(108) ADDR(10.1.100.115)
```

Support sending the FortiGate interface subnet list to EMS

In order to allow FortiClient EMS to share FortiClient information based on IP subnet mask, the FortiGate must send its interface IP and netmask to EMS. This enhancement allows the FortiGate to include its IP and netmask information in the `gateway-mac-request`.

To view the gateway MAC REST API output:

```
# diagnose endpoint fctems json gateway-mac-request
JSON:
"""
{
  "gateway_mac_list":[
    {
      "ip_list":[
        "10.6.30.4"
      ],
      "ip_subnet_list":[
        {
          "gateway_ip":"10.6.30.4",
          "netmask":"255.255.255.0"
        }
      ],
      "mac":"00:0c:29:8f:c5:19",
      "vdom":"root",
      "interface":"port1",
      "sn":"FGVM32TM22000***"
    },
    {
      "ip_list":[
        "10.1.100.4"
      ],
      "ip_subnet_list":[
        {
          "gateway_ip":"10.1.100.4",
          "netmask":"255.255.255.0"
        }
      ],
      "mac":"00:0c:29:8f:c5:23",
      "vdom":"root",
      "interface":"port2",
      "sn":"FGVM32TM22000**"
    }
  ]
}
"""
```

Add the Any and All options back for security posture tags in the GUI - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- Full versus simple ZTNA policies

The *Any* and *All* options in the GUI for the *Security Posture Tag* field are added back to the simple and full ZTNA policy configuration pages. The default setting is *Any*.

In FortiOS 7.4.2 and later, the field name is *Security Posture Tag*. In earlier versions of FortiOS, the field name is *ZTNA Tag*.

For more information about this feature, see Add the Any and All options back for ZTNA tags in the GUI.

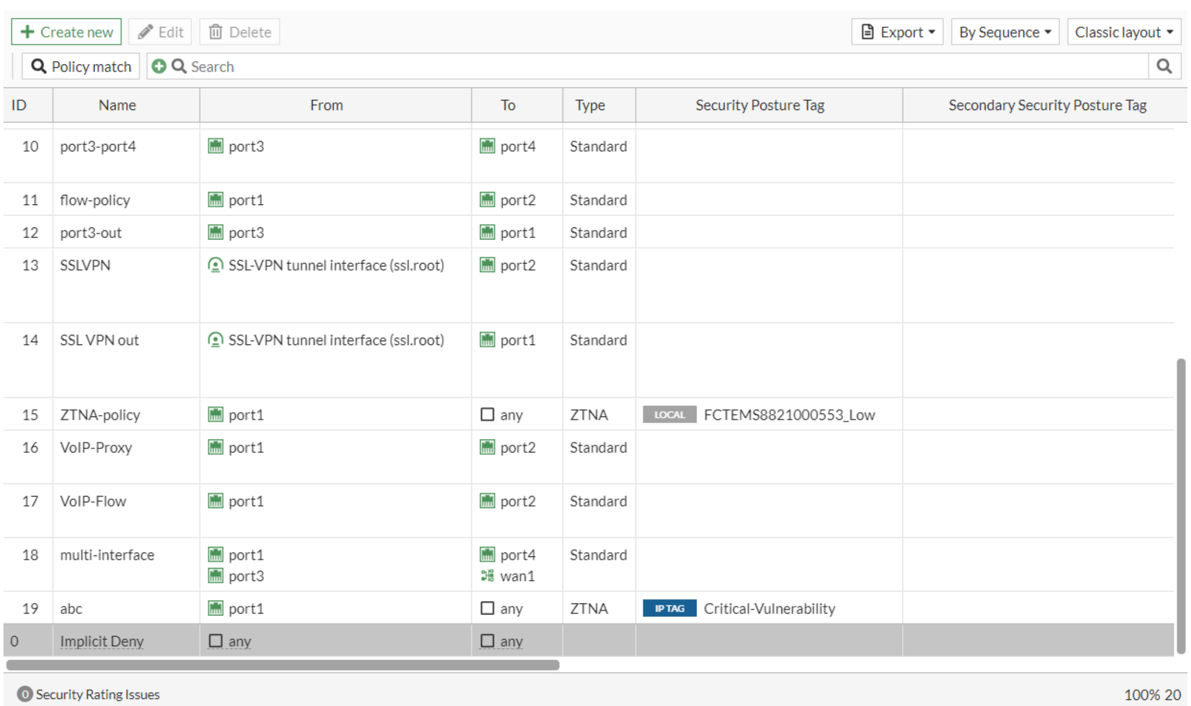
Rename ZTNA Tag to Security Posture Tag in the GUI - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

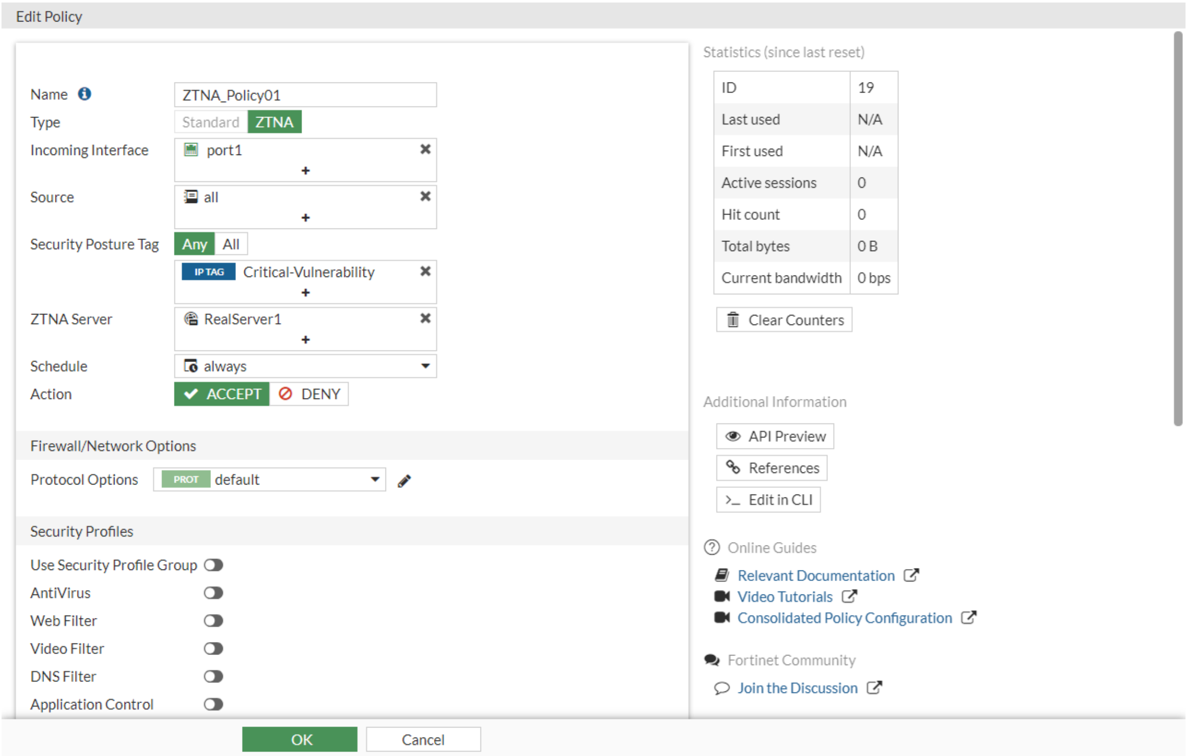
- Full versus simple ZTNA policies

On the *Policy & Objects > Firewall Policy*, *Proxy Policy*, and *ZTNA* pages, *ZTNA Tag* references are renamed *Security Posture Tag*.

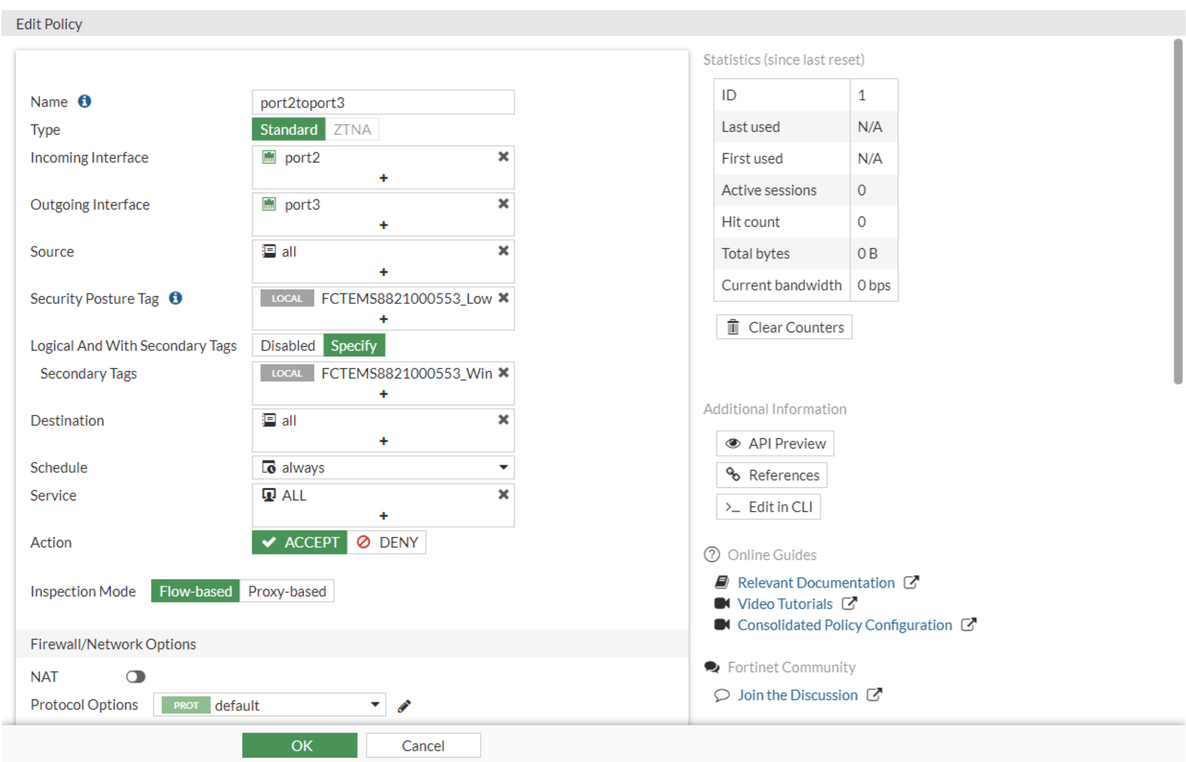
Policy & Objects > Firewall Policy page:



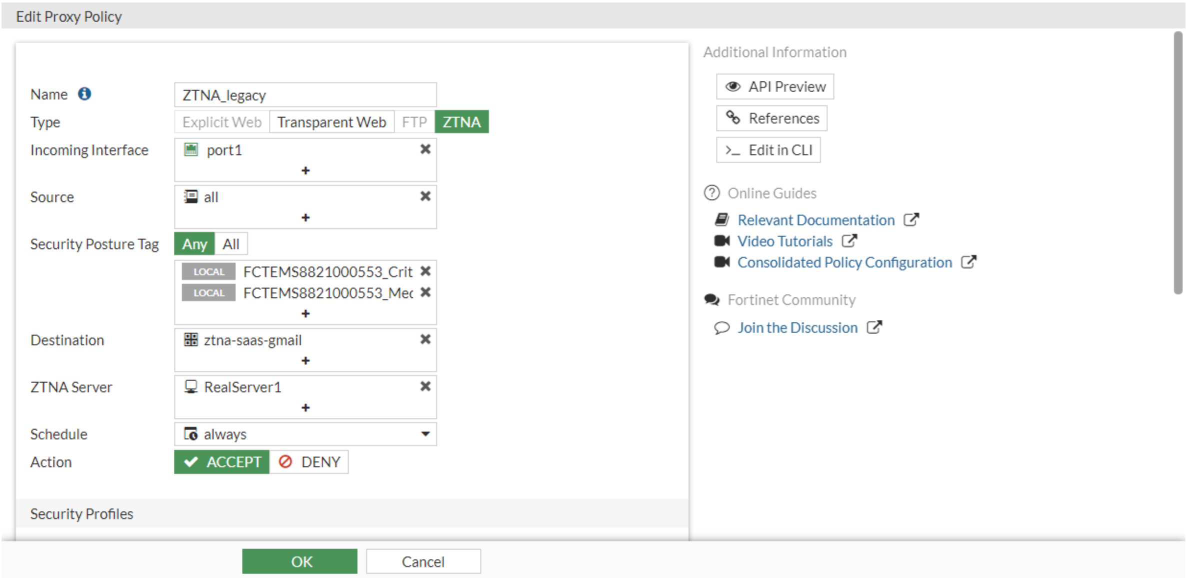
Policy & Objects > *Firewall Policy* ZTNA policy dialog:



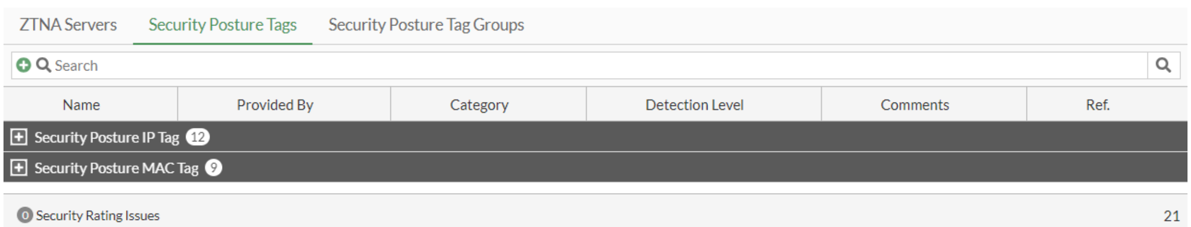
Policy & Objects > *Firewall Policy* standard policy dialog:



Policy & Objects > *Proxy Policy* policy dialog:



Policy & Objects > *ZTNA* page:



The individual tags and groups now appear in separate tabs on the *Policy & Objects* > *ZTNA* page.

ZTNA policies

This section includes information about ZTNA policy related new features:

- Introduce simplified ZTNA rules within firewall policies on page 360

Introduce simplified ZTNA rules within firewall policies

Prior to this enhancement, a ZTNA configuration required configuring:

- An EMS connection and EMS tags
- A ZTNA server configuration
- ZTNA rules
- An authentication scheme and rules (optional)

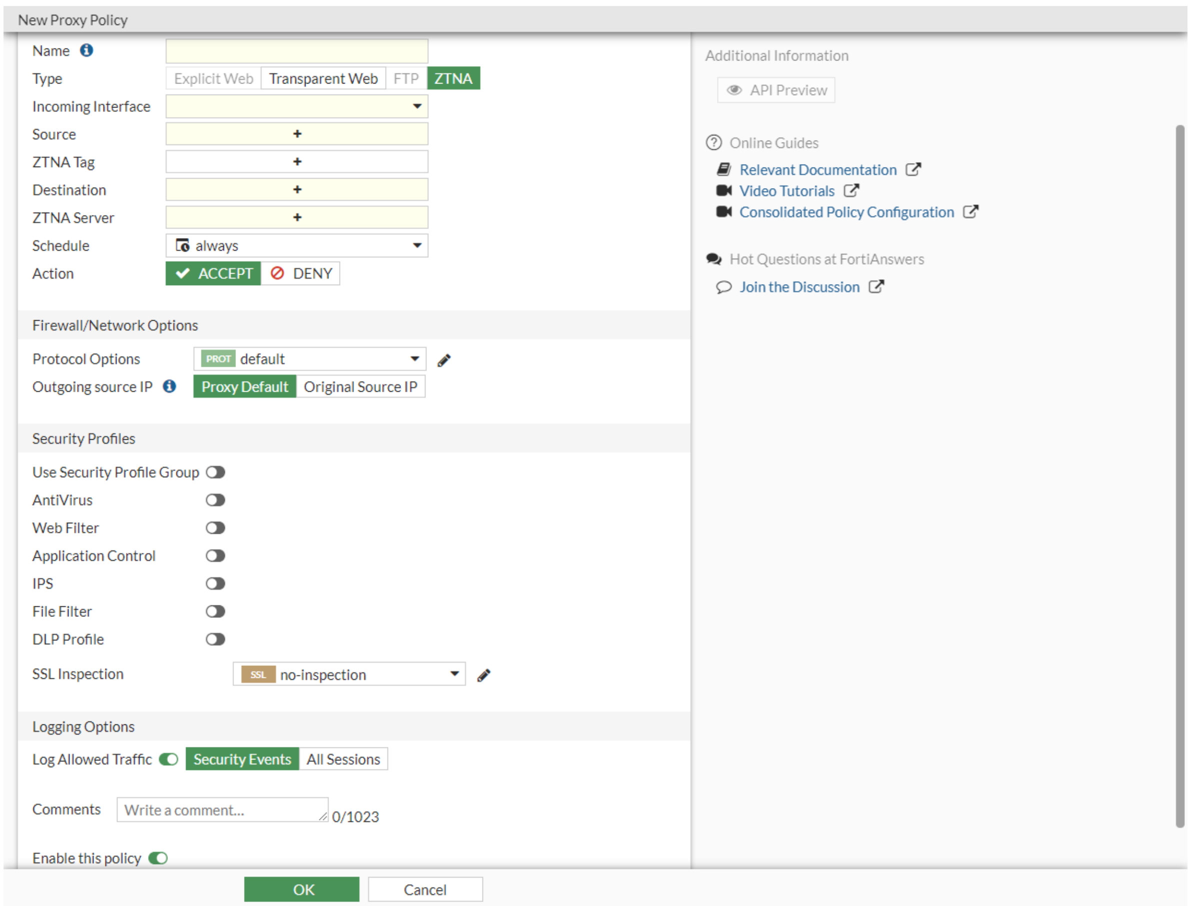
In these settings, ZTNA rules were special proxy policies that controlled access to the ZTNA servers, and they could be configured from the *Policy & Objects* > *ZTNA* > *ZTNA Rules* tab.

In this enhancement, there are now two ways to configure ZTNA rules in the GUI by using a full or simple ZTNA policy.

With the new options to create a full or simple ZTNA policy in the GUI, the *Policy & Objects > ZTNA > ZTNA Rules* tab has been removed.

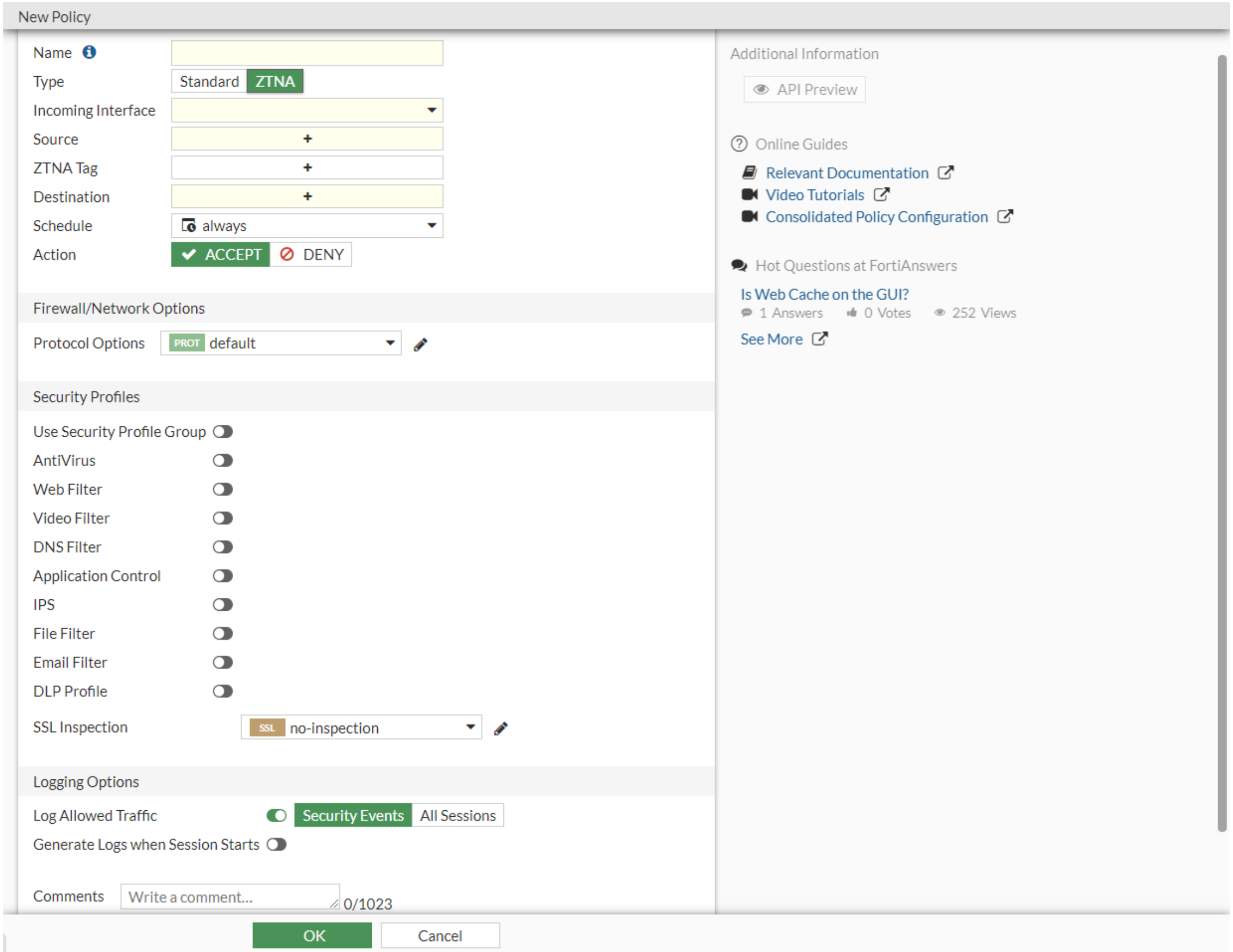
Full ZTNA policy

In a full ZTNA policy, the CLI configuration remains the same as previous versions. In the GUI, the *Policy & Objects > ZTNA > ZTNA Rules* tab has been removed. Administrators can configure ZTNA policies from the *Policy & Objects > Proxy Policy* page, and by setting the *Type* to *ZTNA*.



Simple ZTNA policy

In a simple ZTNA policy, a regular firewall policy is used for policy management. When creating a new firewall policy, administrators can configure a ZTNA policy by setting the *Type* to *ZTNA*.



A simple ZTNA policy cannot control access based on the destination interface or the real server's destination address. See the Examples section for detailed configurations.

Authentication for ZTNA policies

Authentication remains largely the same between both ZTNA policy configuration modes. You can specify user groups under *Source* to define the groups in which the access control applies to. However, the underlying authentication schemes and rules must still be in place to direct the traffic to the ZTNA application gateway.

Authentication for regular firewall policies

Authentication for regular firewall policies is traditionally handled by authd, which does not require an authentication scheme and rules to be configured in order to function. This enhancement allows authentication for regular firewall policies to be handled by WAD so that the authentication scheme and rules are used to determine the type of authentication and the traffic that requires authentication. This option is disabled by default, but can be enabled as follows:

```
config firewall auth-portal
    set proxy-auth {enable | disable}
end
```

Redirecting a simple ZTNA policy to a full ZTNA policy

An option is added so that after matching a simple ZTNA policy, the traffic can be redirected for a full ZTNA policy match. This setting can only be configured from the CLI, and it is disabled by default.

```
config firewall policy
    edit <id>
        set ztna-policy-redirect {enable | disable}
    next
end
```

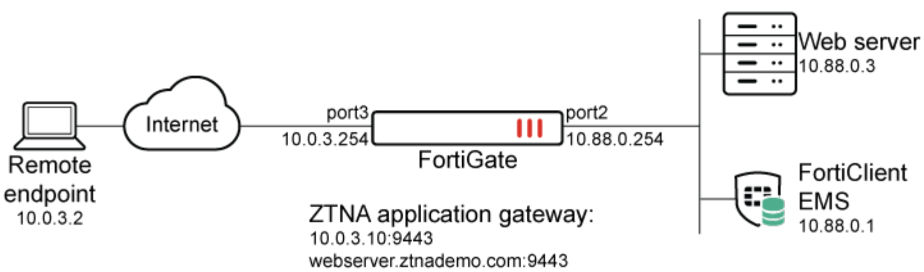
For example, a client has both tag A and tag B. In the simple ZTNA policy, the client matches a policy that requires tag A for a posture check. If they are using the `ztna-policy-redirect` option, then it will also require a full ZTNA policy match.

If a full ZTNA policy allows either tag A or tag B or all traffic in general, then the traffic is allowed. Otherwise, if a full ZTNA policy explicitly denies one of the tags, the traffic will be denied.

If no full ZTNA policy is matched, then the traffic is implicitly denied.

Examples

The following examples demonstrate how to configure a ZTNA policy using the full and simple ZTNA policy modes.



It is assumed that the following settings are already configured:

- EMS connection and EMS tags (Malicious-File-Detected and FortiAD.Info)
- ZTNA server configuration (ZTNA-webserver)
- Authentication scheme and rule

Configuring a full ZTNA policy

To configure a full ZTNA policy in the GUI:

1. Go to *Policy & Objects > Proxy Policy* and click *Create New*.
2. Configure the following settings:

Name	*ZTNA-webserver*
Type	*ZTNA*
Incoming Interface	*port3*
Source	*all*
Destination	*Webserver1* (10.88.0.3/32)
ZTNA Server	*ZTNA-webserver*
Schedule	*always*
Action	*ACCEPT*

3. Click *OK*.

To configure a full ZTNA policy in the CLI:

```
config firewall proxy-policy
    edit 1
        set name "ZTNA-webserver"
        set proxy access-proxy
        set access-proxy "ZTNA-webserver"
        set srcintf "port3"
        set srcaddr "all"
        set dstaddr "Webserver1"
        set action accept
        set schedule "always"
    next
end
```

When traffic is allowed, the ZTNA logs show traffic passing through policy 1 on a policy called `ZTNA-webserver`, which is a proxy policy.

To verify the traffic logs:

```
# execute log filter category traffic
# execute log filter field subtype ztna
# execute log display
9 logs found.
9 logs returned.
1: date=2023-03-06 time=20:16:11 eventtime=1678162572109525759 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=28597
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved"
dstip=10.88.0.3 dstport=9443 dstintf="port2" dstintfrole="dmz" sessionid=20140
srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1" service="tcp/9443" proxyapptype="http"
proto=6 action="accept" policyid=1 policytype="proxy-policy" poluuid="1c0a04b8-bc85-51ed-
48ba-7d43279fb899" policyname="ZTNA-webserver" duration=3604 gatewayid=1 vip="ZTNA-
webserver" accessproxy="ZTNA-webserver" clientdevicemanageable="manageable" wanin=303150
rcvdbyte=303150 wanout=3755 lanin=2813 sentbyte=2813 lanout=304697 appcat="unscanned"
```

Configuring a simple ZTNA policy

To configure a simple ZTNA policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Configure the following settings:

Name	*ZTNA-webserver-fp*
Type	*ZTNA*
Incoming Interface	*port3*
Source	*all*
Destination	*ZTNA-webserver*
Schedule	*always*
Action	*ACCEPT*

3. Click *OK*.

To configure a simple ZTNA policy in the CLI:

```
config firewall policy
    edit 9
        set name "ZTNA-webserver-fp"
        set srcintf "port3"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "ZTNA-webserver"
        set schedule "always"
        set service "ALL"
    next
end
```

When traffic is allowed, the ZTNA logs show traffic passing through policy 9 on a policy called `ZTNA-webserver-fp`, which is a firewall policy.

To verify the traffic logs:

```
# execute log filter category traffic
# execute log filter field subtype ztna
# execute log display
14 logs found.
10 logs returned.

1: date=2023-03-06 time=23:01:55 eventtime=1678172515724776640 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=31687
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved"
dstip=10.88.0.3 dstport=9443 dstintf="port2" dstintfrole="dmz" sessionid=28076
srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1" service="tcp/9443" proxyapptype="http"
proto=6 action="accept" policyid=9 policytype="proxy-policy" poluuid="1f1d5036-bcaa-51ed-
1d28-687edafe9439" policyname="ZTNA-webserver-fp" duration=75 gatewayid=1 vip="ZTNA-
```

```
webserver" accessproxy="ZTNA-webserver" clientdevicemanageable="manageable" wanin=3445
rcvdbyte=3445 wanout=1189 lanin=2358 sentbyte=2358 lanout=4759 appcat="unscanned"
```

Configuring a ZTNA simple policy with ZTNA tags and authentication

In this example, a simple ZTNA policy uses the FortiAD.Info tag for a posture check and authentication against a pre-configured Active Directory server where the user tsmith resides. The authentication scheme and rule have already been configured as follows:

```
config authentication scheme
    edit "ZTNA-Auth-scheme"
        set method basic
        set user-database "LDAP-fortiad"
    next
end
```

```
config authentication rule
    edit "ZTNA-Auth-rule"
        set srcintf "port3"
        set srcaddr "all"
        set active-auth-method "ZTNA-Auth-scheme"
    next
end
```

To append ZTNA tag and authentication settings to the simple ZTNA policy:

1. Go to *Policy & Objects* > *Firewall Policy* and edit the *ZTNA-webserver-fp* policy.
2. For the *Source* field, click the + and add the user group named *LDAP-Remote-Allowed-Group*.
3. For the *ZTNA Tag* field, click the + and add the *FortiAD.Info tag*.
4. Click *OK*.

To verify the configuration:

1. Connect to the web server from a client.
2. After selecting the client certificate, the browser will prompt for a username and password. Enter the username (tsmith) and their password.

   Upon a successful authentication, the user will be able to access the web server.
3. On the FortiGate, verify that the logs for the allowed traffic show the user tsmith and the tag EMS1_ZTNA_FortiAD.Info:

   ```
   # execute log filter field subtype ztna
   # execute log display
   18 logs found.
   10 logs returned.
   1: date=2023-03-06 time=23:25:23 eventtime=1678173923745891128 tz="-0800"
   logid="0005000024" type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2
   srcport=32017 srcintf="port3" srcintfrole="wan" dstcountry="Reserved"
   srccountry="Reserved" dstip=10.88.0.3 dstport=9443 dstintf="port2" dstintfrole="dmz"
   sessionid=29615 srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1" service="tcp/9443"
   proxyapptype="http" proto=6 action="accept" policyid=9 policytype="proxy-policy"
   poluuid="1f1d5036-bcaa-51ed-1d28-687edafe9439" policyname="ZTNA-webserver-fp"
   duration=106 user="tsmith" group="LDAP-Remote-Allowed-Group" authserver="LDAP-fortiad"
   gatewayid=1 vip="ZTNA-webserver" accessproxy="ZTNA-webserver"
   ```

```
clientdeviceid="9A016B5A6E914B42AD4168C066EB04CA" clientdevicemanageable="manageable"
clientdevicetags="MAC_EMS1_ZTNA_all_registered_clients/EMS1_ZTNA_all_registered_
clients/MAC_EMS1_ZTNA_FortiAD.Info/EMS1_ZTNA_FortiAD.Info" emsconnection="online"
wanin=301793 rcvdbyte=301793 wanout=3331 lanin=2877 sentbyte=2877 lanout=333000
fctuid="9A016B5A6E914B42AD4168C066EB04CA" appcat="unscanned"
```

Security profiles

This section includes information about security profile related new features:

Antivirus

This section includes information about antivirus related new features:

Download quarantined files in archive format - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:
- Downloading quarantined files in archive format

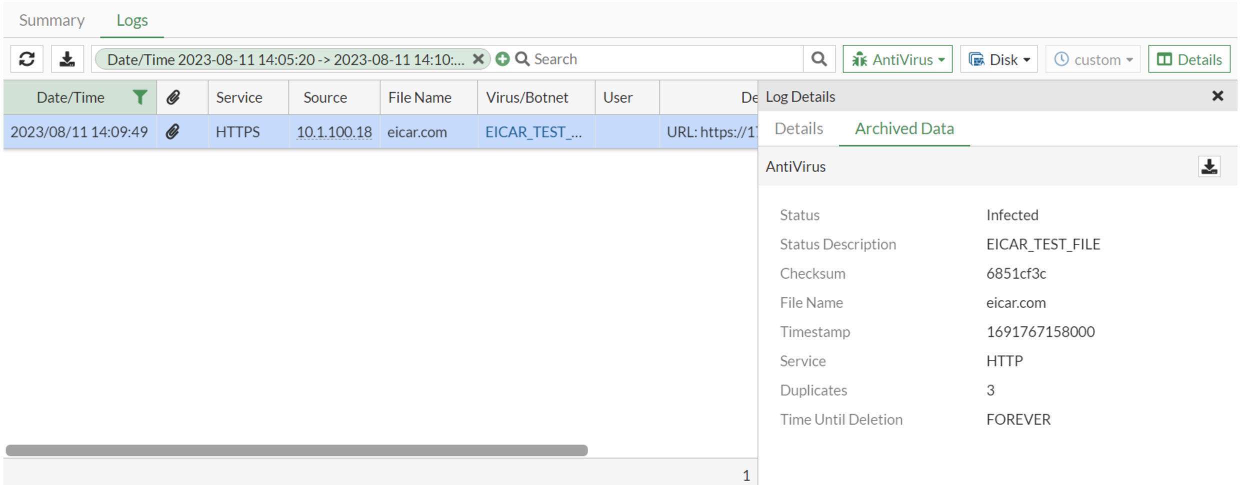
The FortiGate can download quarantined files in an archive format (.TGZ) instead of the original raw file. This allows for a more detailed analysis of the quarantined files and reduces the risk of malware infection.

The FortiGate must have a disk logging capacity or be connected to FortiAnalyzer for logging.

To download a quarantined archive file:

1. Ensure that quarantining files is enabled in the AV profile:
   a. Go to *Security Profiles > AntiVirus* and edit the AV profile.
   b. In the *APT Protection Options* section, verify that *Quarantine* is enabled. At least one protocol must be enabled in the AV profile for inspection, and *AntiVirus scan* must be enabled for the *Quarantine* option to work.
2. Go to *Log & Report > Security Events* and select the *AntiVirus* card.
3. Select a log entry and click *Details*. The *Log Details* pane opens.

4. Select the *Archived Data* tab and click the download icon (in the *AntiVirus* title bar).



Web filter

This section includes information about web filter related new features:

- Add FortiGuard web filter categories for AI and cryptocurrency 7.4.1 on page 369
- Support Punycode encoding for the url and hostname fields in flow inspection logs 7.4.2 on page 372

Add FortiGuard web filter categories for AI and cryptocurrency - 7.4.1

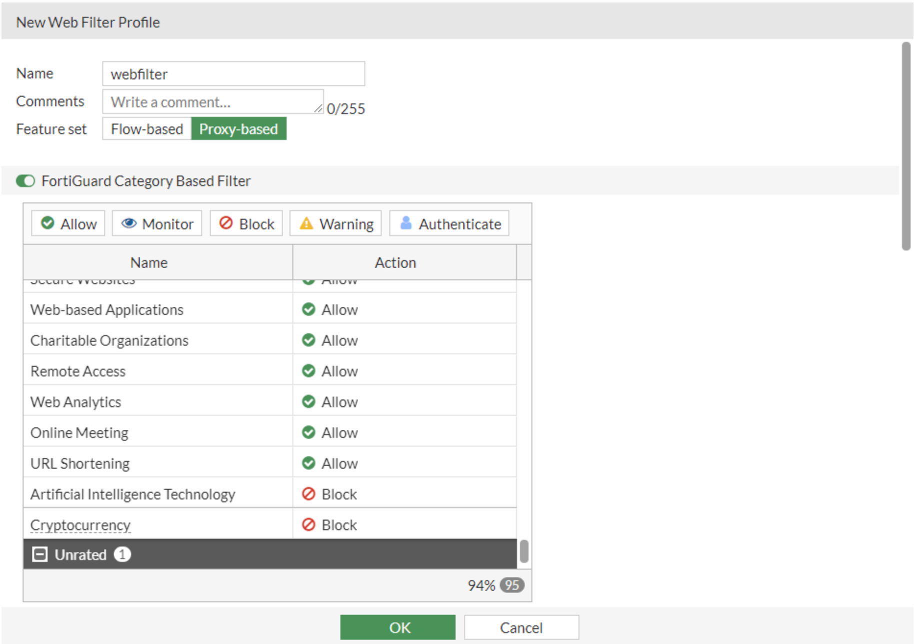
This information is also available in the FortiOS 7.4 Administration Guide:
- Configuring web filter profiles to block AI and cryptocurrency

Two new FortiGuard web filter categories have been added:

- Artificial intelligence technology (category 100): sites that offer solutions, insights, and resources related to artificial intelligence (AI).
- Cryptocurrency (category 101): sites that specialize in digital or virtual currencies that are secured by cryptography and operate on decentralized networks.

To configure a web filter profile to block the AI and cryptocurrency categories in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*.
2. Enter a name for the web filter profile.
3. In the category table, locate the *General Interest - Business* section. Select the *Artificial Intelligence Technology* and *Cryptocurrency* categories, and set the *Action* to *Block*.



4. Configure the remaining settings as needed.
5. Click *OK*.

To configure a web filter profile to block the AI and cryptocurrency categories in the CLI:

```
config webfilter profile
    edit "webfilter"
        set feature-set proxy
        config ftgd-wf
            unset options
            config filters
                edit 100
                    set category 100
                    set action block
                next
                edit 101
                    set category 101
                    set action block
                next
                edit 52
                    set category 52
                next
            end
        end
        set log-all-url enable
    next
end
```

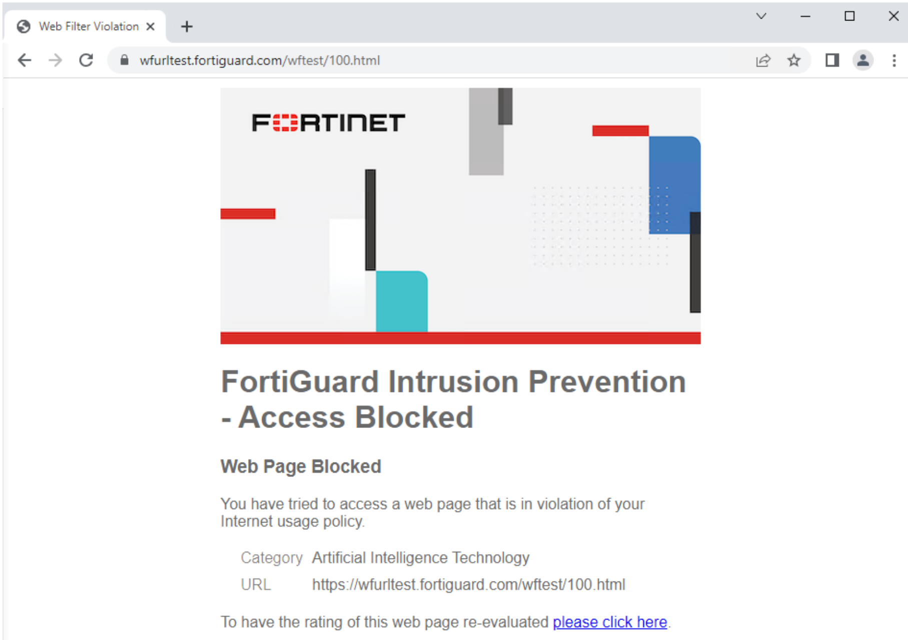
To verify that the categories are blocked:

1. Apply the web filter profile in a firewall policy.
2. On a device that is connected through the FortiGate and uses the policy, visit the test URLs for each category:

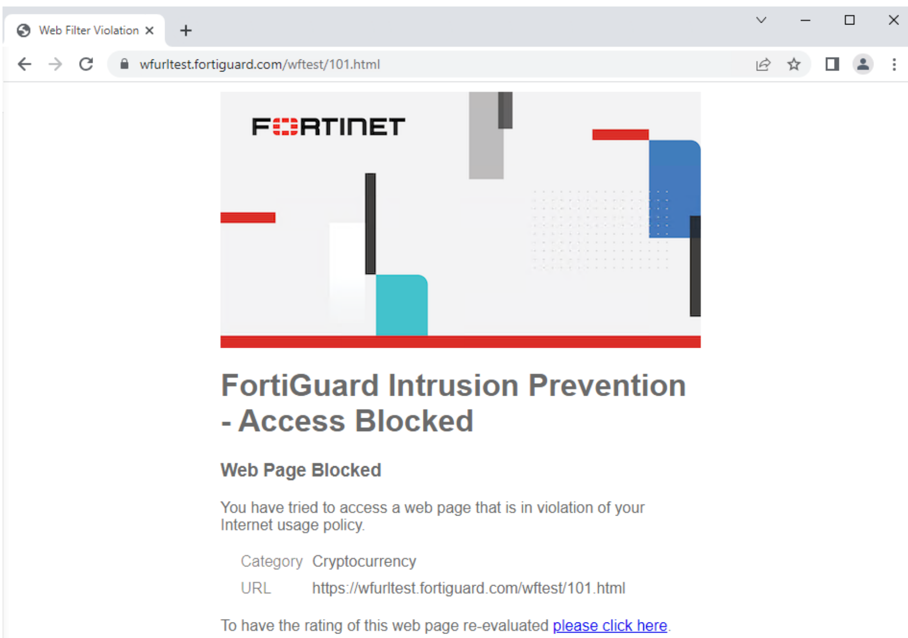
a. `https://wfurltest.fortiguard.com/wftest/100.html`
b. `https://wfurltest.fortiguard.com/wftest/101.html`

The browser displays a replacement message that the URL is blocked based on the FortiGuard category.

- Artificial intelligence technology:



- Cryptocurrency:



To verify the web filter logs:

1. In the GUI, go to *Log & Report > Security Events* and click *Web Filter*.
2. In the CLI, enter the following:

```
# execute log filter category utm-webfilter
# execute log display
1: date=2023-07-12 time=10:39:18 eventtime=1689183557968026063 tz="-0700"
logid="0316013056" type="utm" subtype="webfilter" eventtype="ftgd_blk" level="warning"
vd="vdom1" policyid=1 poluuid="996b0a68-2055-51ee-b841-2b3f373c9b37" policytype="policy"
sessionid=3258 srcip=10.1.100.31 srcport=35116 srccountry="Reserved" srcintf="port2"
srcintfrole="undefined" srcuuid="124f368a-2055-51ee-c7d6-857ab36dd6cb"
dstip=154.52.5.202 dstport=443 dstcountry="United States" dstintf="port1"
dstintfrole="undefined" dstuuid="124f368a-2055-51ee-c7d6-857ab36dd6cb" proto=6
httpmethod="GET" service="HTTPS" hostname="wfurltest.fortiguard.com" agent="curl/7.68.0"
profile="webfilter" action="blocked" reqtype="direct"
url="https://wfurltest.fortiguard.com/wftest/101.html" sentbyte=849 rcvdbyte=3633
direction="outgoing" msg="URL belongs to a denied category in policy"
ratemethod="domain" cat=101 catdesc="Cryptocurrency"

2: date=2023-07-12 time=10:39:13 eventtime=1689183553021358734 tz="-0700"
logid="0316013056" type="utm" subtype="webfilter" eventtype="ftgd_blk" level="warning"
vd="vdom1" policyid=1 poluuid="996b0a68-2055-51ee-b841-2b3f373c9b37" policytype="policy"
sessionid=3255 srcip=10.1.100.31 srcport=35102 srccountry="Reserved" srcintf="port2"
srcintfrole="undefined" srcuuid="124f368a-2055-51ee-c7d6-857ab36dd6cb"
dstip=154.52.5.202 dstport=443 dstcountry="United States" dstintf="port1"
dstintfrole="undefined" dstuuid="124f368a-2055-51ee-c7d6-857ab36dd6cb" proto=6
httpmethod="GET" service="HTTPS" hostname="wfurltest.fortiguard.com" agent="curl/7.68.0"
profile="webfilter" action="blocked" reqtype="direct"
url="https://wfurltest.fortiguard.com/wftest/100.html" sentbyte=849 rcvdbyte=3633
direction="outgoing" msg="URL belongs to a denied category in policy"
ratemethod="domain" cat=100 catdesc="Artificial Intelligence Technology"
```

Support Punycode encoding for the url and hostname fields in flow inspection logs - 7.4.2

Punycode encoding is supported in the `url` and `hostname` fields in flow mode web filter UTM logs. This caters to domain names containing non-ASCII characters, such as internationalized domain names (IDNs). Is also aligns the functionality of flow and proxy modes, offering a more unified and improved user experience.

```
config webfilter profile
    edit <name>
        set web-flow-log-encoding {utf-8 | punycode}
    next
end
```

Example 1: UTF-8 encoding

To configure the web filter profile:

```
config webfilter profile
    edit "webfilter_flowbase"
        set web-flow-log-encoding utf-8
```

```
    next
end
```

Sample log:

```
1: date=2023-10-30 time=11:34:07 eventtime=1698690847433106658 tz="-0700" logid="0316013056"
type="utm" subtype="webfilter" eventtype="ftgd_blk" level="warning" vd="vdom1" policyid=1
poluuid="fc514aae-745e-51ee-d867-15932507e437" policytype="policy" sessionid=15525
srcip=10.1.100.33 srcport=46982 srccountry="Reserved" srcintf="port2"
srcintfrole="undefined" srcuuid="22387062-7397-51ee-fa1e-c6f3f4ae1b8a" dstip=172.16.200.36
dstport=443 dstcountry="Reserved" dstintf="port1" dstintfrole="undefined" dstuuid="22387062-
7397-51ee-fa1e-c6f3f4ae1b8a" proto=6 httpmethod="GET" service="HTTPS" hostname=".jp"
agent="curl/7.80.0-DEV" profile="webfilter_flowbase" action="blocked" reqtype="direct"
url="https://.jp/about/" sentbyte=91 rcvdbyte=0 direction="outgoing" msg="URL belongs to a
denied category in policy" ratemethod="domain" cat=52 catdesc="Information Technology"
```

Example 2: Punycode encoding

To configure the web filter profile:

```
config webfilter profile
    edit "webfilter_flowbase"
        set web-flow-log-encoding punycode
    next
end
```

Sample log:

```
1: date=2023-10-30 time=11:36:25 eventtime=1698690984163852468 tz="-0700" logid="0316013056"
type="utm" subtype="webfilter" eventtype="ftgd_blk" level="warning" vd="vdom1" policyid=1
poluuid="fc514aae-745e-51ee-d867-15932507e437" policytype="policy" sessionid=15552
srcip=10.1.100.33 srcport=42428 srccountry="Reserved" srcintf="port2"
srcintfrole="undefined" srcuuid="22387062-7397-51ee-fa1e-c6f3f4ae1b8a" dstip=172.16.200.36
dstport=443 dstcountry="Reserved" dstintf="port1" dstintfrole="undefined" dstuuid="22387062-
7397-51ee-fa1e-c6f3f4ae1b8a" proto=6 httpmethod="GET" service="HTTPS" hostname="xn--
wgv71a119e.jp" agent="curl/7.80.0-DEV" profile="webfilter_flowbase" action="blocked"
reqtype="direct" url="https://xn--wgv71a119e.jp/about/" sentbyte=91 rcvdbyte=0
direction="outgoing" msg="URL belongs to a denied category in policy" ratemethod="domain"
cat=52 catdesc="Information Technology"
```

IPS

This section includes information about IPS related new features:

- Support full extended IPS database for FortiGate VMs with eight cores or more on page 374
- Support Diameter protocol inspection on the FortiGate 7.4.2 on page 374

Support full extended IPS database for FortiGate VMs with eight cores or more

FortiGate VMs with eight or more vCPUs can be configured to have a minimum of eight cores to be eligible to run the full extended database (DB). Any FortiGate VM with less than eight cores will receive a slim version of the extended DB. The slim-extended DB is a smaller version of the full extended DB that contains top active IPS signatures. It is designed for customers who prefer performance.

Support Diameter protocol inspection on the FortiGate - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- Diameter protocol inspection

Diameter protocol inspection is supported on the FortiGate, which offers the following capabilities.

- Diameter-based packet forwarding and routing: the FortiGate can forward and route Diameter packets that match a firewall policy with an enabled and assigned `diameter-filter profile`. These diameter packets traverse over SCTP or TCP on the reserved port 3868.
- Packet sanity checking: this feature checks if the packet passing through the FortiGate conforms to the Diameter protocol standards as defined in RFC 3588.
  - This includes checking the release version field, error command flags, message length, reserved command flag bits, command code, and tracking the request and answer of the Diameter-based packets.
- Logging: for network auditing purposes, the traffic for both dropped and forwarded Diameter-based packets of the supported commands can be logged. By default, these are disabled.

Diameter protocol is particularly important on interfaces that are used to exchange information with roaming partners, through the Internetwork Packet Exchange (IPX) network.

This feature requires a valid IPS license.

```
config diameter-filter profile
    edit <name>
        set monitor-all-messages {enable | disable}
        set log-packet {enable | disable}
        set track-requests-answers {enable | disable}
        set missing-request-action {allow | block | reset | monitor}
        set protocol-version-invalid {allow | block | reset | monitor}
        set message-length-invalid {allow | block | reset | monitor}
        set request-error-flag-set {allow | block | reset | monitor}
        set cmd-flags-reserve-set {allow | block | reset | monitor}
        set command-code-invalid {allow | block | reset | monitor}
        set command-code-range <min-max>
    next
end
```

`monitor-all-messages {enable | disable}`	Enable/disable logging for all User-Name and Result-Code AVP messages.
`log-packet {enable | disable}`	Enable/disable packet log for triggered Diameter settings.
`track-requests-answers {enable | disable}`	Enable/disable validation that each answer has a corresponding request.
`missing-request-action {allow | block | reset | monitor}`	Set the action to be taken for answers without a corresponding request. • `allow`: allow or pass matching traffic. • `block`: block or drop matching traffic. • `reset`: reset sessions for matching traffic. • `monitor`: allow and log matching traffic.
`protocol-version-invalid {allow | block | reset | monitor}`	Set the action to be taken for an invalid protocol version. • `allow`: allow or pass matching traffic. • `block`: block or drop matching traffic. • `reset`: reset sessions for matching traffic. • `monitor`: allow and log matching traffic.
`message-length-invalid {allow | block | reset | monitor}`	Set the action to be taken for an invalid message length. • `allow`: allow or pass matching traffic. • `block`: block or drop matching traffic. • `reset`: reset sessions for matching traffic. • `monitor`: allow and log matching traffic.
`request-error-flag-set {allow | block | reset | monitor}`	Set the action to be taken for request messages with an error flag set. • `allow`: allow or pass matching traffic. • `block`: block or drop matching traffic. • `reset`: reset sessions for matching traffic. • `monitor`: allow and log matching traffic.
`cmd-flags-reserve-set {allow | block | reset | monitor}`	Set the action to be taken for messages with a command flag reserve bits set. • `allow`: allow or pass matching traffic. • `block`: block or drop matching traffic. • `reset`: reset sessions for matching traffic. • `monitor`: allow and log matching traffic.
`set command-code-invalid {allow | block | reset | monitor}`	Set the action to be taken for messages with an invalid command code. • `allow`: allow or pass matching traffic. • `block`: block or drop matching traffic. • `reset`: reset sessions for matching traffic. • `monitor`: allow and log matching traffic.
`set command-code-range <min-max>`	Set the valid range for command codes (min = 0, max = 16777215, default = 256-16777213).

To configure Diameter protocol inspection:

1. Configure the Diameter filter profile:

```
config diameter-filter profile
    edit "diameter_profile"
```

```
            set monitor-all-messages disable
            set log-packet enable
            set track-requests-answers enable
            set missing-request-action block
            set protocol-version-invalid block
            set message-length-invalid block
            set request-error-flag-set block
            set cmd-flags-reserve-set block
            set command-code-invalid block
            set command-code-range 256-1677213
        next
    end
```

2. Apply the Diameter filter to a firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "deep-inspection"
        set diameter-filter-profile "diameter_profile"
        set logtraffic all
        set auto-asic-offload disable
    next
end
```

NTurbo does not fully support SCTP, so if the configuration includes Diameter-over-SCTP, the `auto-asic-offload` setting should be disabled in the firewall policy. Otherwise, IPS does not get the full session packets.

Sample logs

No matching request:

```
1: date=2023-11-09 time=11:04:32 eventtime=1699556673071701052 logid="0419016386" type="utm"
subtype="ips" eventtype="signature" level="alert" vd="vdom1" severity="info"
srcip=10.1.100.32 srccountry="Reserved" dstip=172.16.200.33 dstcountry="Reserved"
srcintf="port1" srcintfrole="undefined" dstintf="port3" dstintfrole="undefined"
sessionid=163572 action="dropped" proto=132 service="sctp/3868" policyid=1
poluuid="c17362a6-7a84-51ee-0025-80ce4c60ec49" policytype="policy"
attack="Diameter.Response.Message.No.Matching.Request.Found" direction="outgoing"
attackid=52234 ref="http://www.fortinet.com/ids/VID52234" incidentserialno=60817776
msg="diameter_decoder: Diameter.Response.Message.No.Matching.Request.Found, command_
code=317"
```

Invalid protocol version:

```
1: date=2023-11-08 time=20:20:54 eventtime=1699503655386037801 logid="0419016386" type="utm"
subtype="ips" eventtype="signature" level="alert" vd="vdom1" severity="info"
srcip=10.1.100.32 srccountry="Reserved" dstip=172.16.200.33 dstcountry="Reserved"
srcintf="port1" srcintfrole="undefined" dstintf="port3" dstintfrole="undefined"
sessionid=117419 action="dropped" proto=132 service="sctp/3868" policyid=1
poluuid="c17362a6-7a84-51ee-0025-80ce4c60ec49" policytype="policy"
attack="Diameter.Invalid.Version" direction="outgoing" attackid=52229
ref="http://www.fortinet.com/ids/VID52229" incidentserialno=60817657 msg="diameter_decoder:
Diameter.Invalid.Version, protocol_version=2"
```

Incorrect message length:

```
1: date=2023-11-08 time=19:18:10 eventtime=1699499890820325221 logid="0419016386" type="utm"
subtype="ips" eventtype="signature" level="alert" vd="vdom1" severity="info"
srcip=10.1.100.32 srccountry="Reserved" dstip=172.16.200.33 dstcountry="Reserved"
srcintf="port1" srcintfrole="undefined" dstintf="port3" dstintfrole="undefined"
sessionid=113487 action="dropped" proto=132 service="sctp/3868" policyid=1
poluuid="c17362a6-7a84-51ee-0025-80ce4c60ec49" policytype="policy"
attack="Diameter.Incorrect.Message.Length" direction="outgoing" attackid=52230
ref="http://www.fortinet.com/ids/VID52230" incidentserialno=60817601 msg="diameter_decoder:
Diameter.Incorrect.Message.Length, message_length=174, packet_length=164"
```

Request error flag:

```
1: date=2023-11-08 time=19:27:29 eventtime=1699500449951027175 logid="0419016386" type="utm"
subtype="ips" eventtype="signature" level="alert" vd="vdom1" severity="info"
srcip=10.1.100.32 srccountry="Reserved" dstip=172.16.200.33 dstcountry="Reserved"
srcintf="port1" srcintfrole="undefined" dstintf="port3" dstintfrole="undefined"
sessionid=114134 action="dropped" proto=132 service="sctp/3868" policyid=1
poluuid="c17362a6-7a84-51ee-0025-80ce4c60ec49" policytype="policy"
attack="Diameter.Request.Message.Error.Flag.Set" direction="outgoing" attackid=52231
ref="http://www.fortinet.com/ids/VID52231" incidentserialno=60817619 msg="diameter_decoder:
Diameter.Request.Message.Error.Flag.Set, command_flags=A0"
```

Incorrect reserved bits:

```
1: date=2023-11-08 time=19:31:10 eventtime=1699500670891359990 logid="0419016386" type="utm"
subtype="ips" eventtype="signature" level="alert" vd="vdom1" severity="info"
srcip=10.1.100.32 srccountry="Reserved" dstip=172.16.200.33 dstcountry="Reserved"
srcintf="port1" srcintfrole="undefined" dstintf="po/cdoc/ImplementationDoc5906/FGT_
FileFilter_7-4_2512_202311090951_correct config.confrt3" dstintfrole="undefined"
sessionid=114400 action="dropped" proto=132 service="sctp/3868" policyid=1
poluuid="c17362a6-7a84-51ee-0025-80ce4c60ec49" policytype="policy"
attack="Diameter.Incorrect.Reserved.Bits" direction="outgoing" attackid=52232
ref="http://www.fortinet.com/ids/VID52232" incidentserialno=60817626 msg="diameter_decoder:
Diameter.Incorrect.Reserved.Bits, command_flags=82"
```

Out-of-range command code:

```
2: date=2023-11-08 time=16:59:41 eventtime=1699491581561225681 logid="0419016386" type="utm"
subtype="ips" eventtype="signature" level="alert" vd="vdom1" severity="info"
srcip=10.1.100.32 srccountry="Reserved" dstip=172.16.200.33 dstcountry="Reserved"
srcintf="port1" srcintfrole="undefined" dstintf="port3" dstintfrole="undefined"
```

```
sessionid=106658 action="dropped" proto=132 service="sctp/3868" policyid=1
poluuid="c17362a6-7a84-51ee-0025-80ce4c60ec49" policytype="policy"
attack="Diameter.Message.Command.Overlong" direction="outgoing" attackid=52233
ref="http://www.fortinet.com/ids/VID52233" incidentserialno=60817600 msg="diameter_decoder:
Diameter.Message.Command.Overlong, command_code=255, range_min=256, range_max=1677213"
```

Virtual patching

This section includes information about IPS related new features:

- Support OT and IoT virtual patching on NAC policies on page 378
- Virtual patching profile 7.4.1 on page 381
- Improve visibility of OT vulnerabilities and virtual patching signatures 7.4.2 on page 388

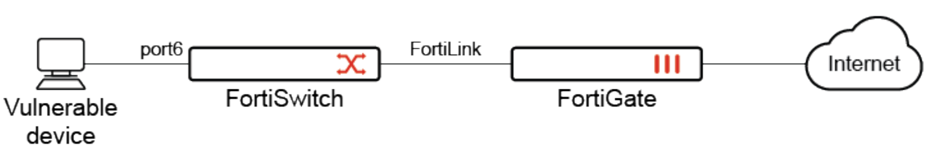
Support OT and IoT virtual patching on NAC policies

This information is also available in the FortiOS 7.4 Administration Guide:
- OT and IoT virtual patching on NAC policies

OT and IoT virtual patching can be applied to a NAC policy by setting the category to *Vulnerability* and configuring the *Match* criteria based on severity. Devices that match the criteria can be assigned and isolated to a NAC VLAN.

Example

In this example, a device with a certain vulnerability severity is detected by the NAC policy on the FortiGate. Subsequently, the FortiSwitch port in which it is connected to is moved to vlan300 where traffic can be controlled for vulnerable devices. For more information about NAC policies, see Defining a FortiSwitch NAC policy in the FortiLink Administration Guide. This example assumes the vlan300 has already been configured.



The following settings are required for IoT device detection:

- A valid IoT Detection Service license to download the IoT signature package.
- Enable device detection on the LAN interface used by IoT devices.
  - In the GUI, go to *Network* > *Interfaces*, edit a LAN interface, enable *Device detection*, and click *OK*.
  - In the CLI, enter:

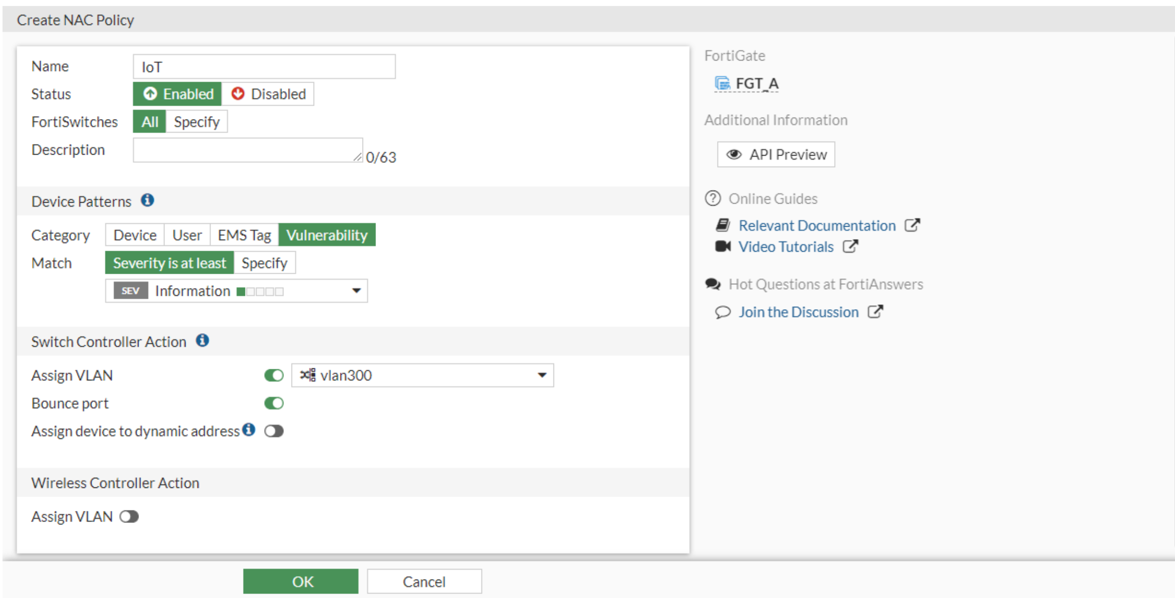
```
config system interface
    edit <name>
        set device-identification enable
    next
```

```
end
```

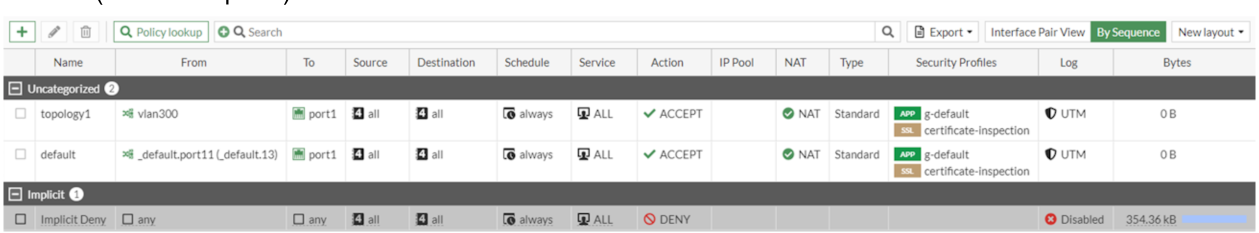
- Configure a firewall policy with an application control sensor.

To configure virtual patching on NAC policies

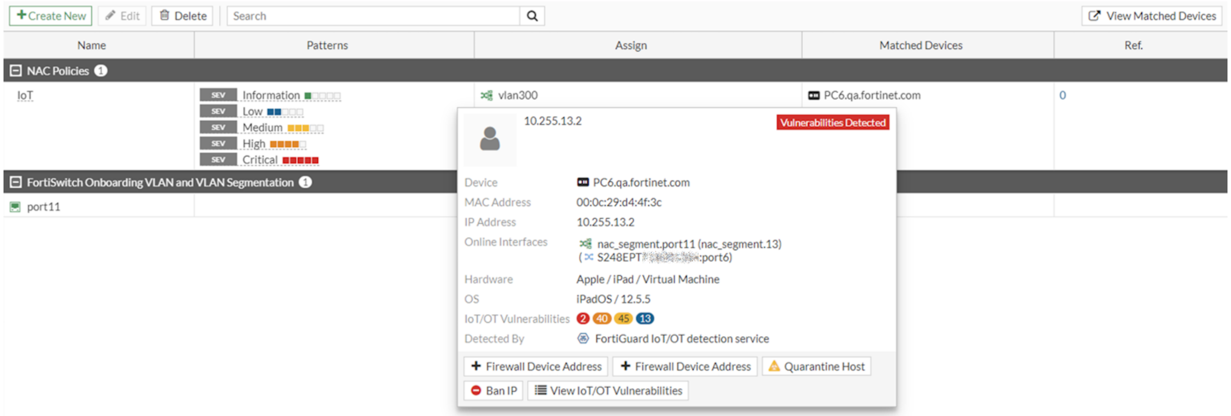
1. Configure the NAC policy:
   a. Go to *WiFi & Switch Controller > NAC Policies* and click *Create New*, or edit an existing policy.
   b. In the *Device Patterns* section, set *Category* to *Vulnerability*.
   c. Set *Match* to *Severity is at least* and select a severity level (*Information* is used in this example).
   d. In the *Switch Controller Action* section, enable *Assign VLAN* and select *vlan300*.



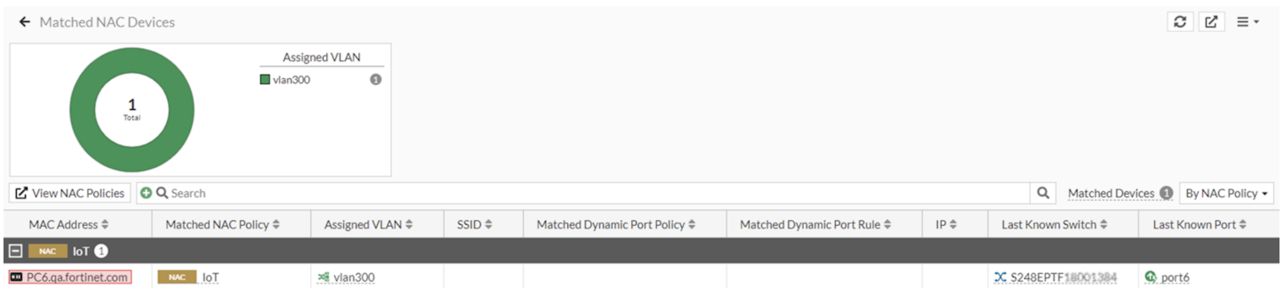
   e. Configure the other settings as needed.
   f. Click *OK*.
2. Enable NAC mode on the desired FortiSwitch ports (port6 in this example):
   a. Go to *WiFi & Switch Controller > FortiSwitch Ports*.
   b. Select *port6*, then right-click and set the *Mode* to *NAC*.
3. Enable application control on the firewall policy that is used to control outbound internet access for vulnerable devices (vlan300 to port1)



4. Generate traffic on the vulnerable client device.
5. Once the NAC policy is matched, go to *WiFi & Switch Controller > NAC Policies* to view the device matched to the policy.



The vulnerable device is also shown on *Dashboards > Assets & Identities* in the *Matched NAC Devices* widget.



To configure virtual patching on NAC policies in the CLI:

1. Configure the VLAN in the MAC policy:

```
config switch-controller mac-policy
    edit "IoT"
        set fortilink "fortilink"
        set vlan "vlan300"
    next
end
```

2. Configure the NAC policy:

```
config user nac-policy
    edit "IoT"
        set category vulnerability
        set severity 0 1 2 3 4
        set switch-fortilink "fortilink"
        set switch-mac-policy "IoT"
    next
end
```

3. Enable NAC mode on the desired FortiSwitch ports:

```
config switch-controller managed-switch
    edit "S248E***********"
        config ports
            edit "port6"
                set access-mode nac
            next
        end
```

```
    next
end
```

4. Configure a firewall policy to limit access for devices in this VLAN (vlan300).

Virtual patching profile - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Virtual patching

Virtual patching is a method for mitigating vulnerability exploits against OT devices by applying patches virtually on the FortiGate. This is done in several steps:

1. A FortiGate uses the OT Detection signatures and service to collect device information from OT devices that are connected to an interface.
2. The device information is used to perform a vulnerability lookup by querying FortiGuard for device-specific vulnerabilities and mitigation rules.
3. The FortiGate caches the applicable signatures and mitigation rules that apply to each device, mapped to the MAC address of the device.
4. When a virtual patching profile is applied to a firewall policy, traffic that enters the firewall policy is subject to signature matching on a per-device basis.
   a. The IPS engine uses the MAC address of the device to match any mitigation rules that should apply.
   b. If the MAC address is in the exempted list, then patching is exempted or skipped.
   c. If the signature rule is in the exempted list, then patching is also exempted or skipped for that signature.
   d. Otherwise, all applicable rules for the device will be applied.

A virtual patching profile can be applied to firewall policies in any direction, protecting traffic from or to the vulnerable OT devices. Virtual patching profiles can also be combined with virtual patching on NAC policies, so that vulnerable OT devices are first assigned to a protected VLAN, and then firewall policies associated with the VLAN will apply the virtual patching profile. See OT virtual patching on NAC policies for more information.

The following are requirements for the virtual patching feature:

- Purchase the appropriate OT-related license (virtual patching only applies to OT devices). See Operational Technology Security Service 7.4.1 on page 656 for more information.
- Enable device detection on the LAN interface.
  - In the GUI, go to *Network* > *Interfaces*, edit a LAN interface, enable *Device detection*, and click *OK*.
  - In the CLI, enter:

```
config system interface
    edit <name>
        set device-identification enable
    next
end
```

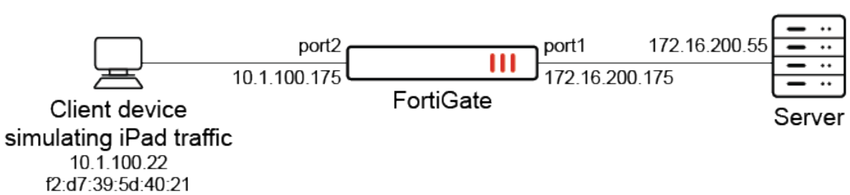
- Configure a firewall policy with an application control profile in order for device detection to occur. OT device detection collects device information by triggering application control signatures.

The following options can be configured in a virtual patching profile:

GUI option	CLI option	Description
Basic profile settings		
Name	`name <string>`	Enter a unique name for the profile.
Severity	`severity {low medium high critical}`	Set the relative severity of the signature, from low to critical.
Action	`action {pass | block}`	Set the action to take for a matched device: • *Pass*/`pass`: allow sessions that match the profile. • *Block*/`block`: block sessions that match the profile (default).
Logging	`log {enable | disable}`	Enable/disable detection logging. This setting is enabled by default.
Comments	`comment <var-string>`	Enter a comment (optional).
Virtual patching exemptions settings		
Status	`status {enable | disable}`	Enable/disable exemption.
MAC addresses	`device <mac_address1>, <mac_address2>, ...`	Enter the device MAC addresses to exempt.
Signature ID	`rule <id1>, <id2>, ...`	Enter the pre-defined or custom signatures to exempt.

Example 1: basic configuration

This example demonstrates the flow for OT virtual patching from start to finish. First, a device (10.1.100.22) goes through device detection, which matches an OT detection signature downloaded on the FortiGate. Next, known vulnerabilities and OT patch signatures for this device are mapped to its MAC address. When traffic is generated by this device, IPS scans the traffic to identify any traffic patterns that match known OT patch signatures for this device. If a match is found, traffic is blocked by the FortiGate.



For demonstrative purposes, the simulated vulnerable OT device is a PC simulating web traffic from an iPad. An OT detection signature is specially crafted to match this Apple iPad traffic to the OT device category. To simulate vulnerable traffic, a test OT patch signature is used to match a generic cross-site scripting (XSS) attack over HTTP.

To verify the status of the OT related definitions:

1. Verify the current contracts licensed to the FortiGate:

```
# diagnose test update info
…
```

```
OTDT,Mon Sep 24 17:00:00 2029
OTVP,Mon Sep 24 17:00:00 2029
…
```

2. Verify the versions and status of the OT definitions:

```
# diagnose autoupdate versions
…
OT Detect Definitions
---------
Version: 23.00545 signed
Contract Expiry Date: Sun Sep 23 2029
Last Updated using manual update on Thu Jul 20 09:40:03 2023
Last Update Attempt: n/a
Result: Updates Installed
--
OT Patch Definitions
---------
Version: 23.00505 signed
Contract Expiry Date: Sun Sep 23 2029
Last Updated using manual update on Thu Jul 20 09:39:50 2023
Last Update Attempt: n/a
Result: Updates Installed
…
```

3. View the OT detection rules downloaded on the FortiGate. In this example, the OT detection rule ID 1000870 is a specially crafted signature to match Apple iPad traffic to the OT category:

```
# get rule otdt status
app-name: "Apple.iPad"
id: 10000870
category: "OT"
cat-id: 34
popularity: 5.low
risk: 1.medium
weight: 10
shaping: 0
protocol: 1.TCP, 9.HTTP
vendor: 7.Apple
technology: 0.Network-Protocol
behavior:
dev_cat: Other
```

4. View the OT patch rules downloaded on the FortiGate. In this example, the OT patch rule is a specially crafted signature to match a generic XSS attack to a vulnerability:

```
# get rule otvp status
rule-name: "WAP.Generic.XSS"
rule-id: 10000684
rev: 20.321
date: 1653379200
action: pass
status: enable
log: disable
log-packet: disable
severity: 2.medium
service: TCP, HTTP
location: server
```

```
os: Other
application: Other
rate-count: 0
rate-duration: 0
rate-track: none
rate-mode: continuous
vuln_type: XSS
cve: 20198625
```

To configure virtual patching in the GUI:

1. Enable device detection on port2 :
   a. Go to *Network* > *Interfaces* and edit port2.
   b. In the *Network* section, enable *Device detection*.
   c. Click *OK*.
2. Configure the virtual patching profile:
   a. Go to *Security Profiles* > *Virtual Patching* and click *Create New*.
   b. Configure the following settings:

Name	*test*
Severity	Select *Low*, *Medium*, *High*, and *Critical*
Action	*Block*
Logging	*Enable*

   c. Click *OK*.
3. Apply the virtual patching profile to a firewall policy for traffic from port2 to port1:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. In the *Security Profiles* section, enable *Virtual Patching* and select the virtual patch profile (*test*).
   c. Enable *Application Control* and select an application control profile (*default*).
   d. Set *SSL Inspection* to a profile that uses deep inspection profile in order to scan SSL encrypted traffic.
   e. Configure the other settings as needed.
   f. Click *OK*.

To configure virtual patching in the CLI:

1. Enable device detection on port2:

```
config system interface
    edit "port2"
        set device-identification enable
    next
end
```

2. Configure the virtual patching profile:

```
config virtual-patch profile
    edit "test"
        set comment ''
        set severity low medium high critical
        set action block
```

```
        set log enable
    next
end
```

3. Apply the virtual patching profile to a firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "custom-deep-inspection"
        set application-list "default"
        set virtual-patch-profile "default"
        set nat enable
    next
end
```

To test the virtual patching:

1. On the PC, generate traffic that simulates web traffic from an iPad. This traffic is generated in order for the FortiGate to perform device detection on port2. The OT detection signature 10000870 will be triggered, which considers this traffic from an OT device in this simulated scenario:

```
# curl 172.16.200.55 -H "User-Agent: Mozilla/5.0 (iPad; CPU OS 12_5_5 like Mac OS X)
AppleWebKit/605.1.15 (KHTML, like Gecko) Version/10.1.2 Mobile/15E148 Safari/604.1"
```

A log is generated, indicating the traffic that triggered the match:

```
3: date=2023-07-24 time=15:31:26 eventtime=1690237885960202460 tz="-0700"
logid="1059028704" type="utm" subtype="app-ctrl" eventtype="signature"
level="information" vd="root" appid=10000870 srcip=10.1.100.22 srccountry="Reserved"
dstip=172.16.200.55 dstcountry="Reserved" srcport=51548 dstport=80 srcintf="port2"
srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" proto=6 service="HTTP"
direction="outgoing" policyid=1 poluuid="a3424268-1ffc-51ed-3ba9-f3a60e2271cf"
policytype="policy" sessionid=7284 applist="default" action="pass" appcat="OT"
app="Apple.iPad" hostname="172.16.200.55" incidentserialno=18882457 url="/"
agent="Mozilla/5.0 (iPad; CPU OS 12_5_5 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like
Gecko) Version/10.1.2 Mobile/15E148 Safari/604.1" httpmethod="GET" msg="OT: Apple.iPad"
clouddevice="Vendor=Apple, Product=ipados, Version=12.5.5, Firmware=IOS" apprisk="low"
```

The FortiGate queries the FortiGuard OT query service with information about the OT device vendor and product. The service responds with the vulnerabilities and patch_sign_id applicable to this device. IPS caches this information in its device vulnerability database.

2. Verify the vulnerability by device MAC and IP address:

```
# diagnose user-device-store device memory vulnerability-query f2:d7:39:5d:40:21
10.1.100.22
Got 28 vulnerabilities, response size:1792
[Vulnerability-0]
        'vulnerability_id' = '110977'
```

```
        'severity' = '2'
        'signature' = '10000684'
```

3. Verify the virtual patch signatures stored and enabled on the FortiGate:

```
# diagnose ips share list otvp_cfgcache
f2:d7:39:5d:40:21 1 10000684
```

4. Using the vulnerable device 10.1.100.22, generate vulnerable traffic to the destination server 172.16.200.55. The traffic from this IP and MAC address triggers OT patch signature 1000684 to match and is subsequently blocked by the firewall policy:

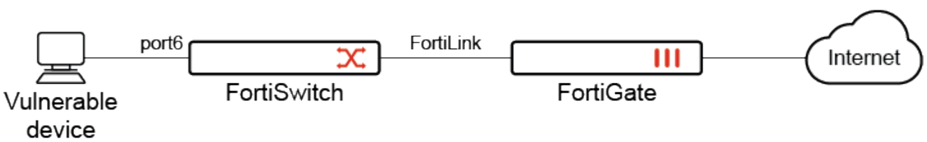
```
# curl -X POST http://172.16.200.55/'index.html?<javascript>'
```

5. Verify the UTM virtual patch log that was recorded with information about the vulnerability that was virtually patched:

```
# execute log filter category 24
# execute log display
2 logs found.
2 logs returned.

1: date=2023-07-20 time=16:03:00 eventtime=1689894179977743851 tz="-0700"
logid="2400064600" type="utm" subtype="virtual-patch" eventtype="virtual-patch"
level="warning" vd="root" count=medium srcip=10.1.100.22 profiletype="Reserved"
dstip=172.16.200.55 direction="Reserved" srcintfrole="port2" dstintf="undefined"
dstintfrole="port1" sessionid=undefined eventtype="12514" action="dropped" proto=6
service="HTTP" policyid=1 poluuid="a3424268-1ffc-51ed-3ba9-f3a60e2271cf"
policytype="policy" attack="WAP.Generic.XSS" srcport=47830 dstport=80
hostname="172.16.200.55" url="/index.html?<javascript>" agent="curl/7.61.1"
httpmethod="POST" direction="outgoing" attackid=10000684
```

Example 2: NAC policy

In this example, a NAC policy is pre-configured to detect devices with information or higher vulnerabilities, as demonstrated in OT virtual patching on NAC policies. The NAC policy assigns the devices to vlan300.



A virtual patching profile is created to block any vulnerabilities with low, medium, high, or critical severity. The profile is applied to a firewall policy for outbound traffic.

To configure virtual patching in the GUI:

1. Enable device detection on vlan300:
   a. Go to *Network* > *Interfaces* and edit vlan300.
   b. In the *Network* section, enable *Device detection*.
   c. Click *OK*.
2. Configure the virtual patching profile:
   a. Go to *Security Profiles* > *Virtual Patching* and click *Create New*, or edit an existing profile.
   b. Configure the following settings:

Name	*OT_check*
Severity	Select *Low*, *Medium*, *High*, and *Critical*
Action	*Block*
Logging	*Enable*

   c. Click *OK*.
3. Apply the virtual patching profile to a firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*, or edit an existing policy.
   b. In the *Security Profiles* section, enable *Virtual Patching* and select the virtual patch profile (*OT_check*).
   c. Enable *Application Control* and select an application control profile (*default*).
   d. Configure the other settings as needed.
   e. Click *OK*.

To configure virtual patching in the CLI:

1. Enable device detection on vlan300:

```
config system interface
    edit "vlan300"
        set device-identification enable
    next
end
```

2. Configure the virtual patching profile:

```
config virtual-patch profile
    edit "OT_check"
        set severity low medium high critical
    next
end
```

3. Apply the virtual patching profile to a firewall policy:

```
config firewall policy
    edit 1
        set name "virtualpatch-policy"
        set srcintf "vlan300"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set application-list "default"
        set virtual-patch-profile "OT_check"
        set logtraffic all
    next
end
```

4. Verify the logs:

```
# execute log filter category utm-virtual-patch
# execute log display
```

```
...
1: date=2023-06-20 time=16:21:00 eventtime=1686180059982988434 tz="-0700"
logid="2400064600" type="utm" subtype="virtual-patch" eventtype="virtual-patch"
level="warning" vd="root" severity="medium" srcip=10.1.100.11 srccountry="Reserved"
dstip=172.16.200.55 dstcountry="Reserved" srcintf="vlan300" srcintfrole="undefined"
dstintf="port1" dstintfrole="undefined" sessionid=1445 action="dropped" proto=6
service="HTTP" policyid=1 poluuid="ce6b724c-0558-51ee-e9d3-f0b8ef1c115f"
policytype="policy" attack="WAP.Generic.XSS" srcport=37062 dstport=80
hostname="172.16.200.55" url="/index.html?<javascript>" agent="curl/7.61.1"
httpmethod="POST" direction="outgoing" attackid=10000684
ref="http://www.fortinet.com/ids/VID10000684" incidentserialno=214959182 msg="vPatch:
WAP.Generic.XSS" crscore=10 craction=16384 crlevel="medium"
```

Improve visibility of OT vulnerabilities and virtual patching signatures - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

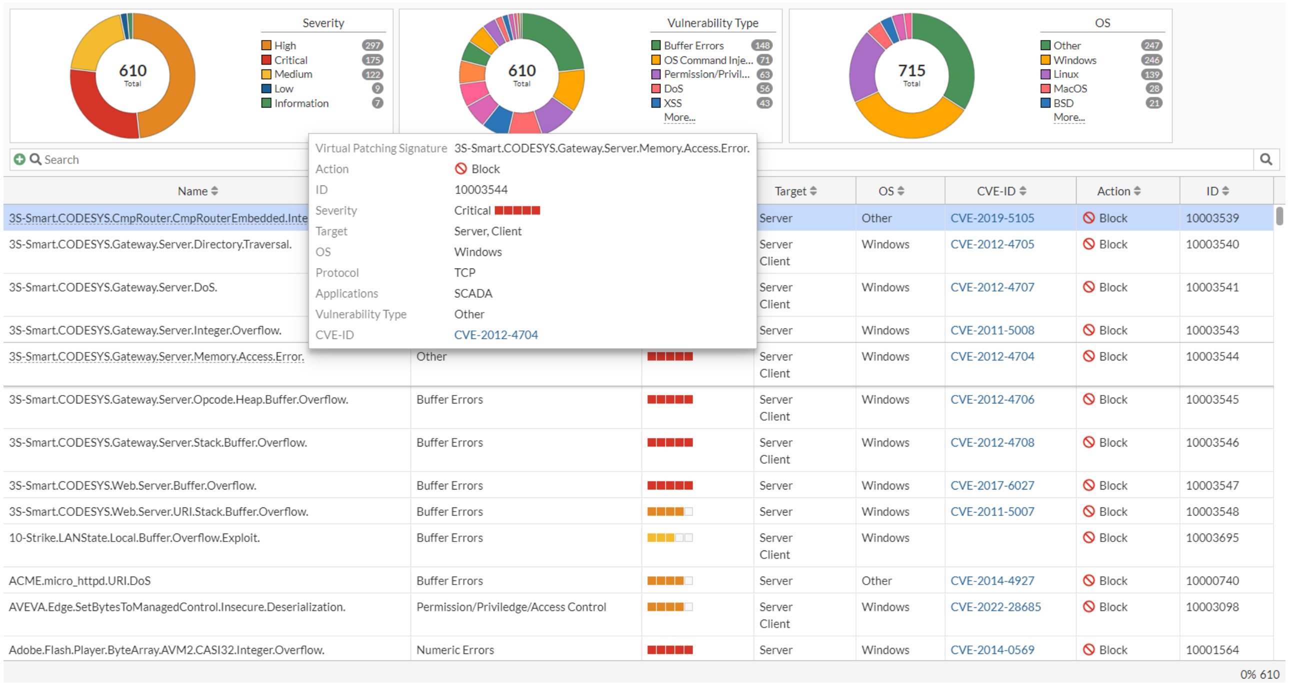
- Virtual patching signatures
- License and entitlement information
- Virtual patching exemptions

The following improvements have been made in the GUI for the visibility of OT vulnerabilities and virtual patching signatures:

- Add a *Security Profiles > Virtual Patching Signatures* page that displays all OT virtual patching signatures.
- In the *Assets* widget (*Dashboard > Assets & Identities*), display a tooltip for detected IoT and OT vulnerabilities when hovering over the *Vulnerabilities* column.
- Add the *View IoT/OT Vulnerabilities* option per device to drill down and list the IoT and OT vulnerabilities.
- Display the OT Security Service entitlement status and OT package versions in the right-side gutter of a virtual patching profile page.
- Display suggestions when creating a new virtual patching exemption.

Virtual Patching Signatures page:

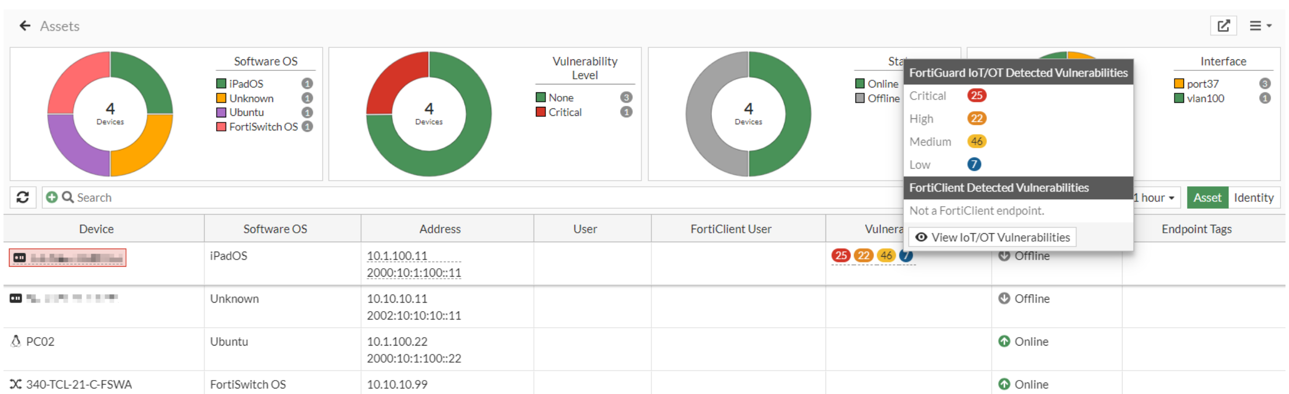
In order to view the *Virtual Patching* and *Virtual Patching Signatures* pages, ensure that *Virtual Patching* is enabled on the *System > Feature Visibility* page.



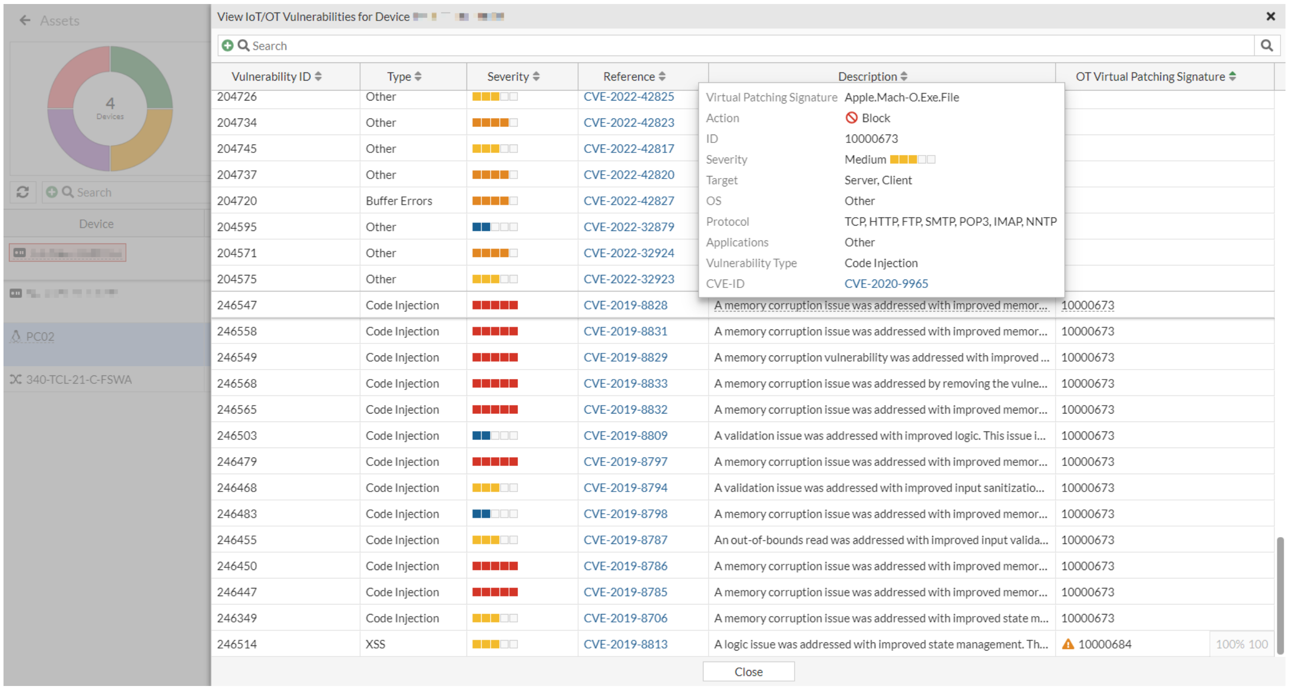
The *Virtual Patching Signatures* page displays all OT virtual patching signatures. When using multi VDOM mode, the OT virtual patching signatures are displayed per VDOM.

Assets widget:

Hovering over the *Vulnerabilities* column displays a tooltip with a summary of FortiGuard detected IoT and OT vulnerabilities for the selected device.

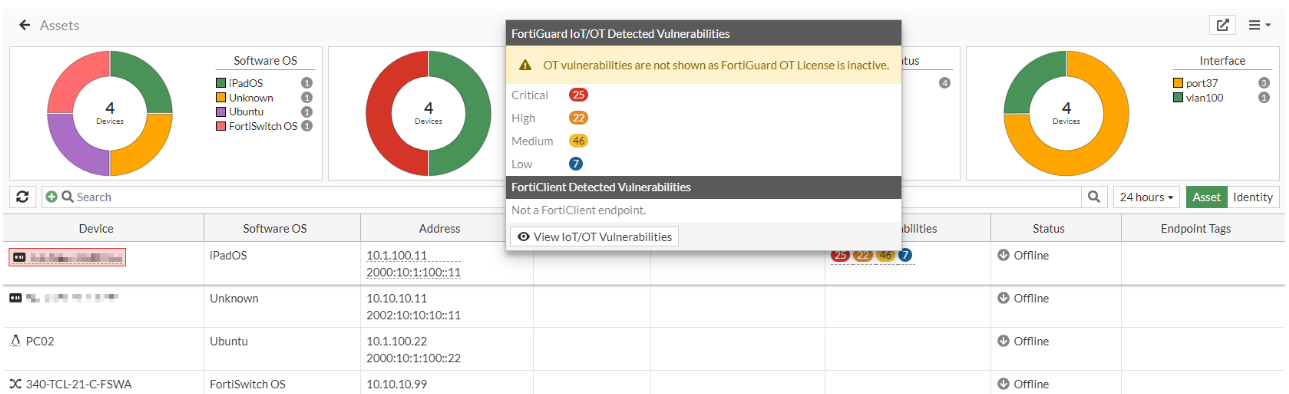


Clicking *View IoT/OT Vulnerabilities* in the tooltip displays a list of vulnerabilities retrieved from the FortiGuard API server for the device. The *OT Virtual Patching Signature* column includes the virtual patch signature ID that is mapped to the *Vulnerability ID*.



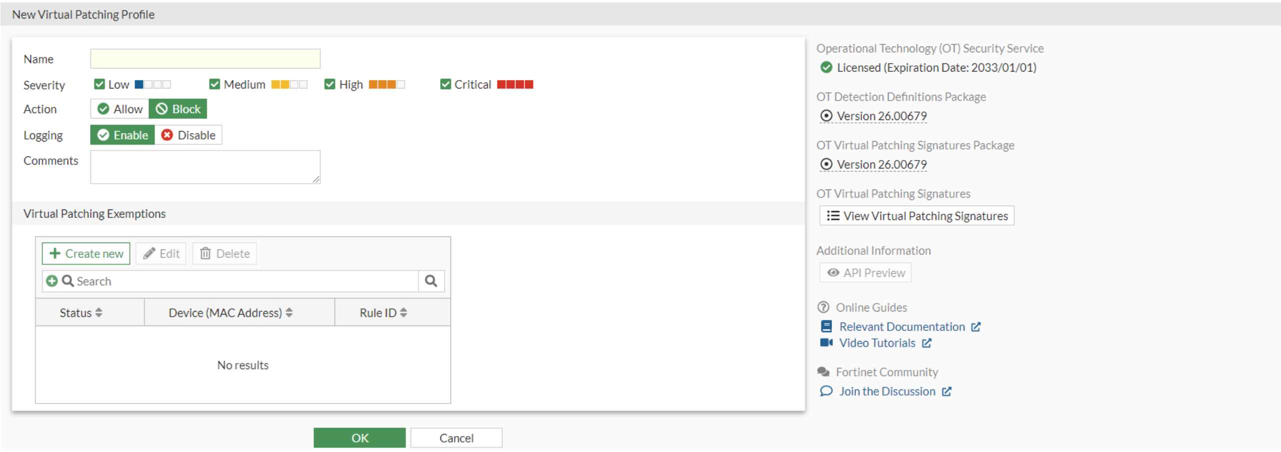
License and entitlement information:

If a FortiGate does not have a valid OT license, a warning message is included in top of the IoT and OT vulnerabilities tooltip (*Assets* widget), indicating that OT vulnerabilities will not be detected.

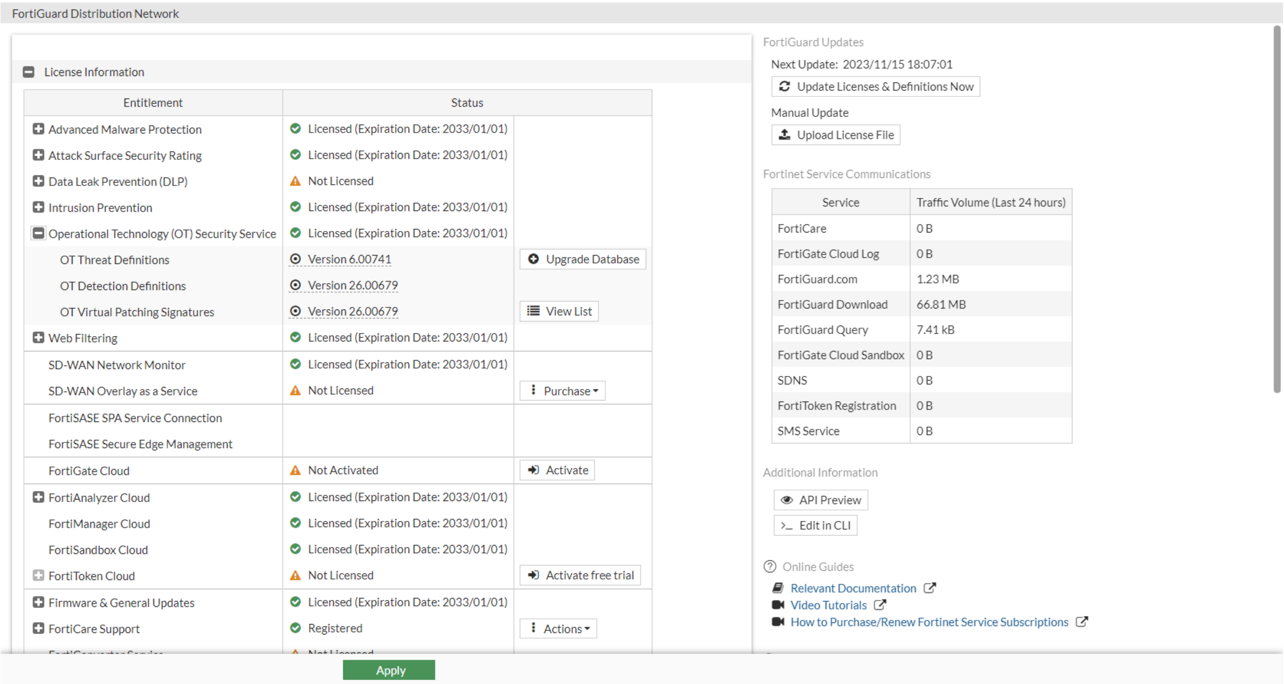


The right-side gutter of virtual patching profile pages includes information about the following:

- *Operational Technology (OT) Security Service* entitlement status
- *OT Detection Definitions Package* version
- *OT Virtual Patching Signatures Package* version

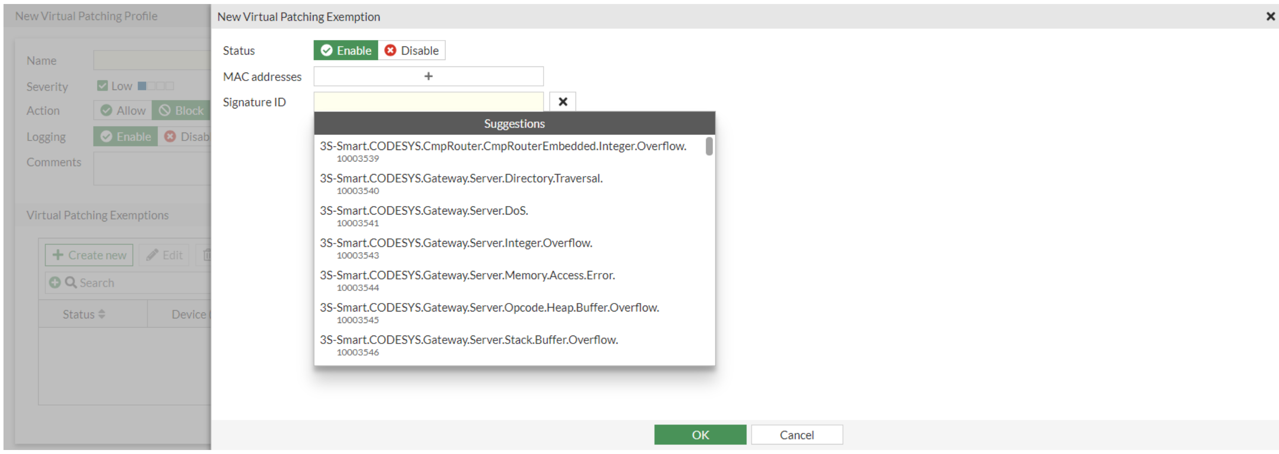


The *System > FortiGuard* page also includes the list of signatures under the *Operational Technology (OT) Security Service* entitlement.



Virtual patching exemptions:

When creating a new virtual patching exemption in a virtual patching profile, the *Signature ID* field includes a dropdown below it with suggestions (signature name and ID). Users can select a signature from the *Suggestions* dropdown or type in the *Signature ID* field to find a specific signature.



Others

This section includes information about other security profile related new features:

- Improve replacement message displayed in blocked videos on page 392
- Introduce SIP IPS profile as a complement to SIP ALG on page 394
- Add inline CASB security profile 7.4.1 on page 397
- Support domain name in XFF with ICAP 7.4.1 on page 413
- Enhance the video filter profile with a new level of customization and control 7.4.2 on page 417

Improve replacement message displayed in blocked videos

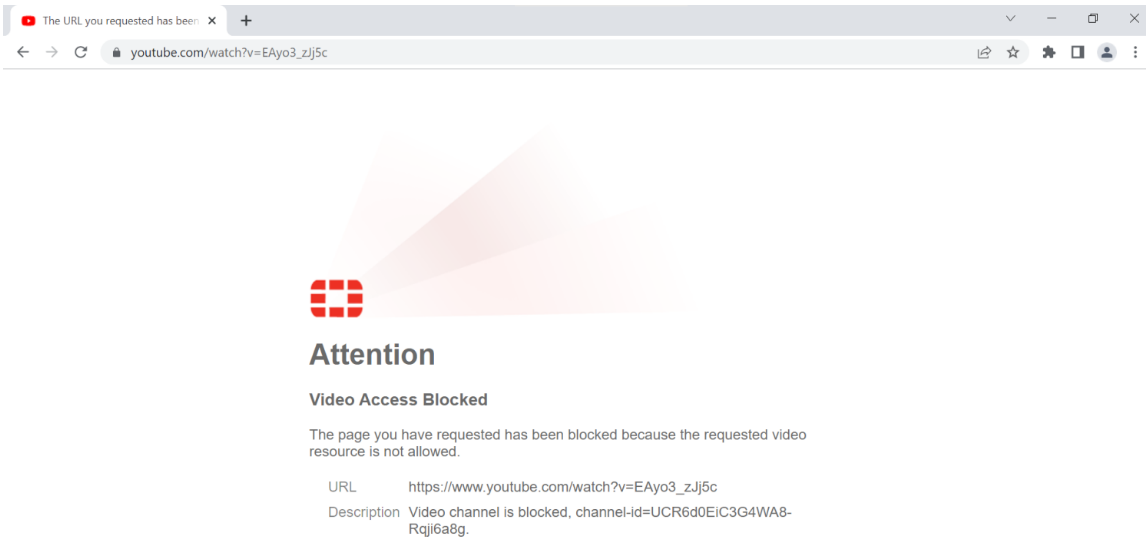
This information is also available in the FortiOS 7.4 Administration Guide:
- Replacement messages displayed in blocked videos

This enhancement improves how a replacement message is displayed for YouTube videos blocked by video filtering. When a user visits a video directly by a URL, a full page replacement message is displayed. When a user loads a video from the YouTube website (homepage or recommended videos), the page loads and the replacement message is displayed in the video frame.

For more information about configuring video filters, see Filtering based on FortiGuard categories and Filtering based on YouTube channel in the FortiOS Administration Guide.

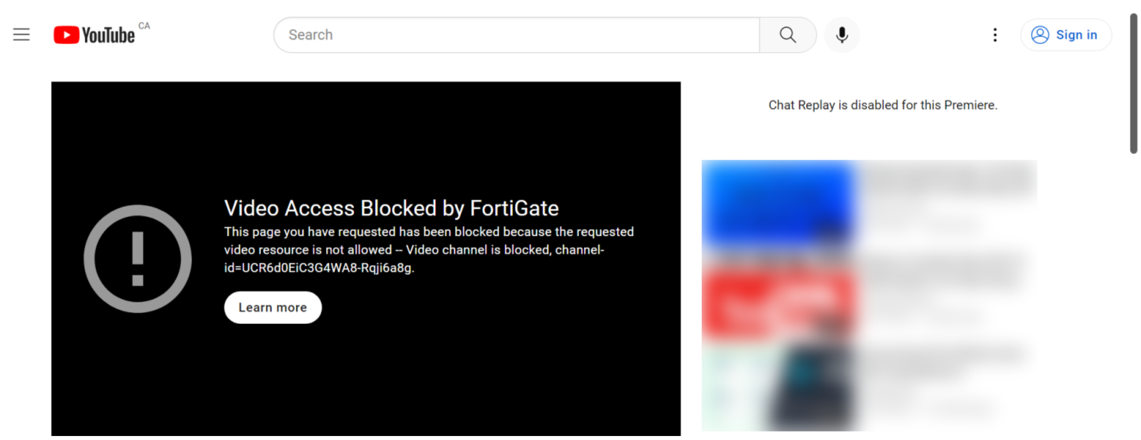
Example 1: blocking the video based on the URL

In this example, the user entered the URL of a blocked channel ID in their browser. The replacement message is displayed in the browser (full page).



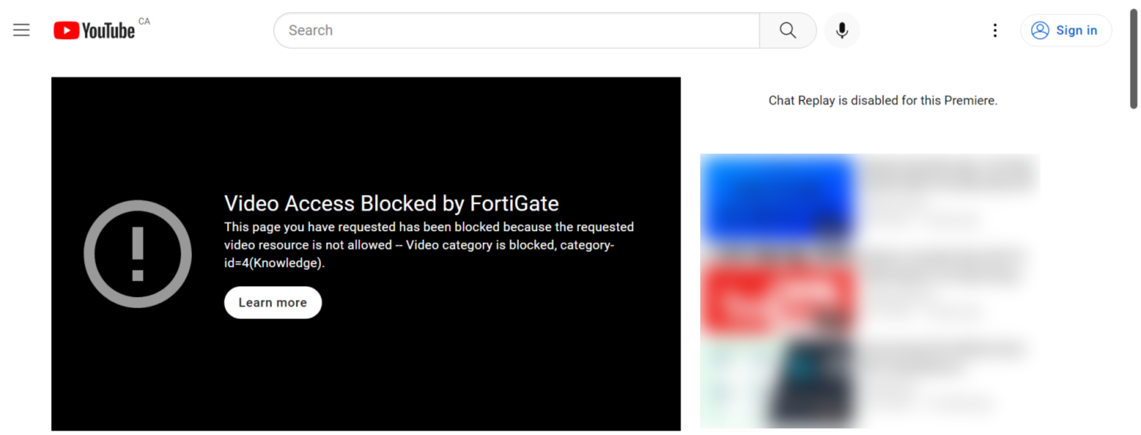
Example 2: blocking the video based on channel ID on YouTube

In this example, the user visited a blocked channel ID on the YouTube website. The replacement message is displayed in the video frame.



Example 3: blocking the video based on FortiGuard category on YouTube

In this example, the user visited a video on the YouTube website that belongs to a blocked FortiGuard category. The replacement message is displayed in the video frame.



Introduce SIP IPS profile as a complement to SIP ALG

This information is also available in the FortiOS 7.4 Administration Guide:

- SIP message inspection and filtering

In FortiOS 7.0, flow-based SIP inspection was introduced, which is handled by the IPS Engine. When a VoIP profile is applied to a firewall policy, the inspection mode determines whether SIP ALG or flow-based SIP is used. Therefore, SIP ALG and flow-based SIP were mutually exclusive. You could not use both at the same time.

Proxy-based SIP ALG is able to handle features such as pin hole creation and NAT that flow-based SIP inspection cannot. Flow-based SIP can handle features such as MSRP decoding and scanning that proxy-based SIP ALG cannot.

To solve this problem, FortiOS 7.4.0 introduces a new IPS-based VoIP profile (`ips-voip-filter`) that allows flow-based SIP to complement SIP ALG while working together.

```
config firewall policy
    edit <id>
        set ips-voip-filter <name>
    next
end
```

The VoIP profile selection within a firewall policy is restored to pre-7.0 behavior. The `voip-profile` can be selected regardless of the `inspection-mode` in the firewall policy.

Previously, in the VoIP profile, users were able to select either a `proxy` or `flow` based feature set. These have been renamed to `voipd` and `ips`. Two options are added in the SIP configuration.

```
config voip profile
    edit <name>
        set feature-set {ips | voipd}
        config sip
            set call-id-regex <string>
            set call-id-regex <string>
        end
    next
end
```

feature-set {ips | voipd}	Set the inspection feature set. • ips: (formerly flow) use the IPS Engine feature set for the ips-voip-filter firewall policy option. • voipd: (formerly proxy) use the SIP ALG feature set for voip-profile firewall policy option.
call-id-regex <string>	Available when the ips feature set is selected. Enter a validation PCRE regular expression for the Call-Id header value.
call-id-regex <string>	Available when the ips feature set is selected. Enter a validation PCRE regular expression for the Content-Type header value.

A SIP ALG VoIP profile can be selected in a firewall policy to handle VoIP traffic with SIP ALG features. For example:

```
config firewall policy
    edit 1
        set voip-profile "voip_sip_alg"
    next
end
```

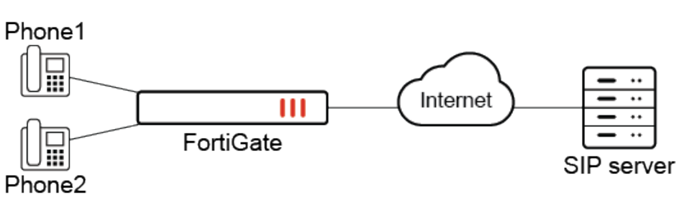
An IPS-based VoIP profile can be selected with a SIP ALG VoIP profile within the same firewall policy. For example:

```
config firewall policy
    edit 1
        set voip-profile "voip_sip_alg"
        set ips-voip-filter "voip_sip_ips"
    next
end
```

When both SIP ALG and SIP IPS are used and configured with same block rules, SIP IPS will take priority and do the blocking.

Example

In this example, SIP ALG is required for pinhole creation, handling NAT, and controlling SIP messages that requires flow-based SIP. The administrator needs to configure two SIP profiles, one with each feature set (voipd and ips), and apply these SIP profiles in the same firewall policy.



To configure SIP ALG with SIP IPS:

1. Configure the VoIP profiles:

```
config voip profile
    edit "voip_sip_alg"
        set feature-set voipd
```

```
            set comment "sip_alg_simple"
            config sip
                set log-violations enable
                set log-call-summary enable
            end
        next
        edit "voip_sip_ips"
            set feature-set ips
            set comment "ips_voip_blocking"
            config sip
                set block-invite enable
                set log-violations enable
            end
        next
    end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port1"
        set dstintf "port9"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ips-sensor "g-default"
        set voip-profile "voip_sip_alg"
        set ips-voip-filter "voip_sip_ips"
        set logtraffic all
        set nat enable
    next
end
```

To verify the SIP proxy SIP calls:

1. Verify the register request:

```
# diagnose sys sip-proxy calls
sip calls
  vdom 1 (vdom1) vrf 0 call 7f2b99828300
    call-id: 619216389
    txn 7f2b998ad600 (REGISTER)
      cseq 2 dir 0 state 5 status 200 expiry 527 HA 0
      i_session: 7f2b998aac00  r_session: 7f2b998aac00
      register: present
      from: sip:2001@172.16.200.44
      to: sip:2001@172.16.200.44
      src: 10.1.100.11:5060
      dst: 172.16.200.44:5060
```

2. Verify the invite request:

```
# diagnose sys sip-proxy calls
sip calls
  vdom 1 (vdom1) vrf 0 call 7f2b99828300
    call-id: 619216389
    txn 7f2b998ad600 (REGISTER)
      cseq 2 dir 0 state 5 status 200 expiry 316 HA 0
      i_session: 7f2b998aac00  r_session: 7f2b998aac00
      register: present
      from: sip:2001@172.16.200.44
      to: sip:2001@172.16.200.44
      src: 10.1.100.11:5060
      dst: 172.16.200.44:5060
```

Sample logs

Register request:

```
date=2023-01-13 time=09:46:03 eventtime=1673631963477298677 tz="-0800" logid="0814044032"
type="utm" subtype="voip" eventtype="voip" level="information" vd="vdom1" session_id=17092
epoch=0 event_id=1 srcip=10.1.100.11 src_port=5060 dstip=172.16.200.44 dst_port=5060
proto=17 src_int="port1" dst_int="port9" policy_id=1 profile="voip_sip_alg" voip_proto="sip"
kind="register" action="permit" status="succeeded" duration=0 dir="session_origin" call_
id="619216389" from="sip:2001@172.16.200.44" to="sip:2001@172.16.200.44"
```

Invite request:

```
date=2023-01-13 time=09:54:43 eventtime=1673632484065549240 tz="-0800" logid="0814044033"
type="utm" subtype="voip" eventtype="voip" level="notice" vd="vdom1" session_id=17092
epoch=0 event_id=0 srcip=10.1.100.11 src_port=5060 dstip=172.16.200.44 dst_port=5060
proto=17 src_int="port1" dst_int="port9" policy_id=1 profile="voip_sip_ips" voip_proto="sip"
kind="call" action="block" status="N/A" reason="block-request" duration=0 dir="session_
reverse" message_type="request" request_name="INVITE" call_id="1967779864" count=0
from="<sip:2001@172.16.200.44>" to="<sip:2002@172.16.200.44>" attackid=50083
attack="SIP.Invite.Method"
```

Add inline CASB security profile - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:
- Inline CASB

The inline CASB security profile enables the FortiGate to perform granular control over SaaS applications directly on firewall policies. The supported controls include:

Control	Description
Privilege control	Specify the action to apply to user activities per application such as upload, download, share, delete, log in, and so on.
Safe search	On SaaS applications that support searching, enable and select the level of safe search.
Tenant control	Allow only users belonging to specific domains to access the SaaS application.

Control	Description
UTM bypass	For each user activity, bypass further UTM scanning any of the following security profiles: • Antivirus • DLP • File filter • Video filter • Web filter

Administrators can customize their own SaaS applications, matching conditions, and custom controls and actions.

A firewall policy must use proxy-based inspection with a deep inspection SSL profile to apply the inline CASB profile and scan the traffic payload.

Inline CASB can be applied to a firewall policy or a proxy policy.

The Inline-CASB Application Definitions entitlement is licensed under the basic firmware and updates contract. To view the entitlement information, go to *System > FortiGuard* and expand the *Firmware & General Updates* section.

To enable inline CASB security profiles in the GUI:

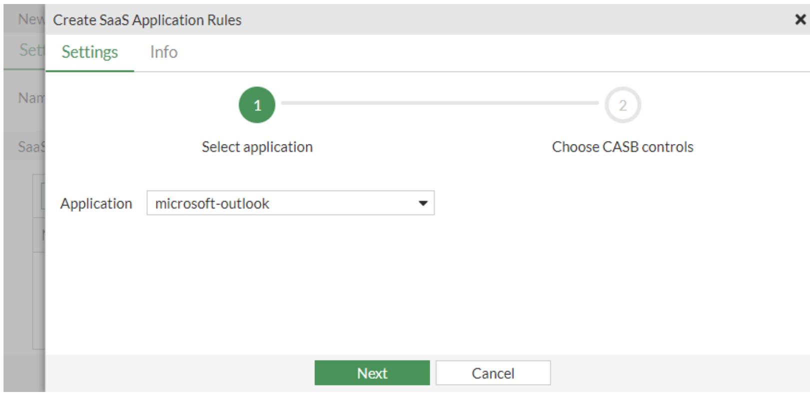
1. Go to *System > Feature Visibility*.
2. Enable *Inline-CASB* in the *Security Features* section.
3. Click *Apply*.

Example 1: privilege control

In this example, logging in to Microsoft Outlook is blocked by the privilege control settings in the inline CASB profile.

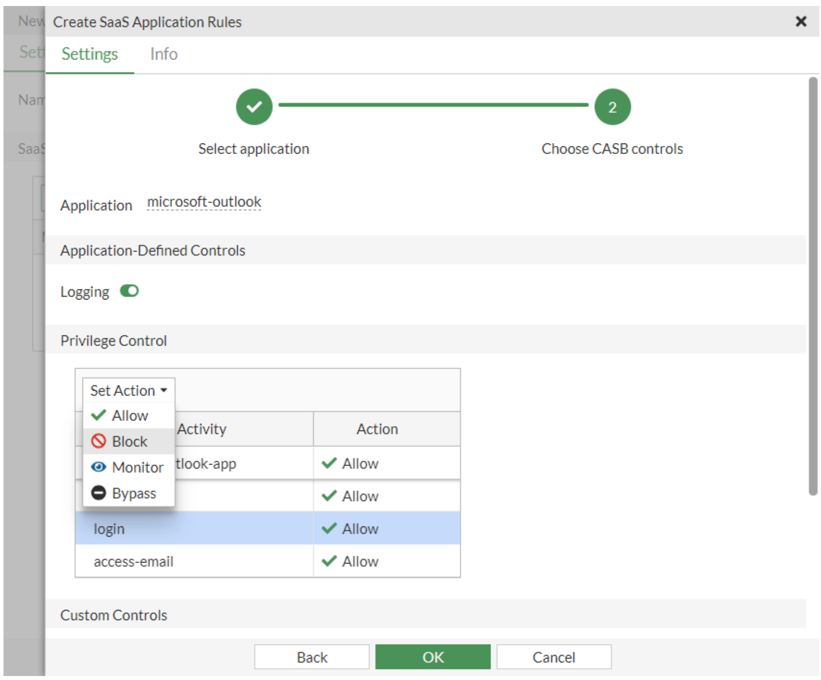
To configure an inline CASB profile with privilege control in the GUI:

1. Configure the inline CASB profile:
   a. Go to *Security Profiles > Inline-CASB* and click *Create new*.
   b. Enter a *Name*, such as *outlook_test*.
   c. In the *SaaS Applications* table, click *Create new*. The *Create SaaS Application Rules* pane opens.
   d. Set the *Application* to *microsoft-outlook*, then click *Next*.



e. Enable *Logging*.

f. In the *Privilege Control* table, select *login* and from the *Set Action* dropdown, select *Block*.



g. Click *OK*.

2. Configure the firewall policy:
   a. Go to *Policy & Objects > Firewall Policy*. Edit an existing policy, or create a new one.
   b. Set the *Inspection Mode* to *Proxy-based*.
   c. In the *Security Profiles* section, enable *Inline-CASB* and select the *outlook_test* profile.
   d. Set the *SSL Inspection* profile to one that uses deep inspection.
   e. Configure the other settings as needed.
   f. Click *OK*.

To configure an inline CASB profile with privilege control in the CLI:

1. Configure the inline CASB profile:

```
config casb profile
    edit "outlook_test"
```

```
        config saas-application
            edit "microsoft-outlook"
                config access-rule
                    edit "microsoft-outlook-login"
                        set action block
                    next
                end
            next
        end
    next
end
```

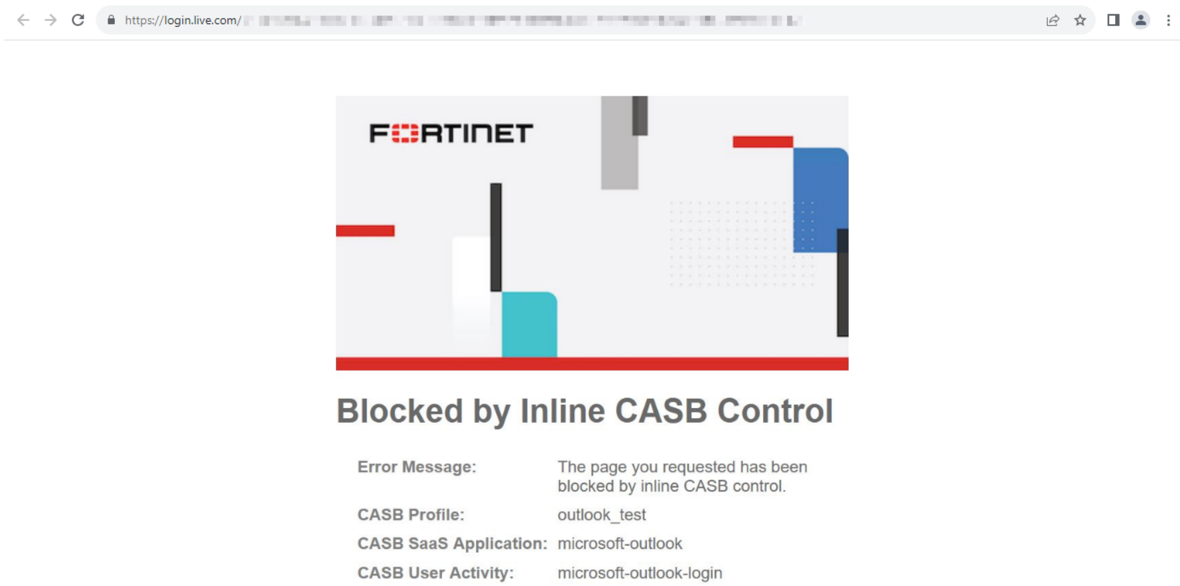
2. Configure the firewall policy:

```
config firewall policy
    edit 6
        set name "casb_test"
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "ssl"
        set casb-profile "outlook_test"
        set nat enable
    next
end
```

To test the configuration:

1. Open a browser and attempt to access the Outlook login page.
2. The traffic is blocked by the firewall policy. The browser displays a replacement message: *Blocked by Inline CASB*

Control.



Sample log:

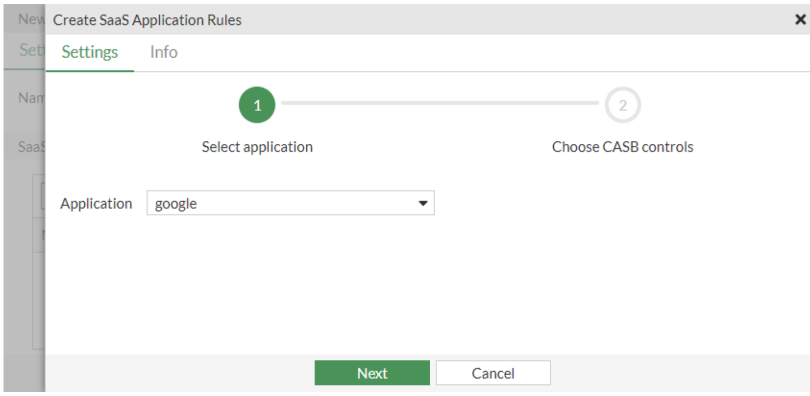
```
1: date=2023-08-18 time=16:59:32 eventtime=1692403171962221884 tz="-0700" logid="2500010000"
type="utm" subtype="casb" eventtype="casb" level="warning" vd="vdom1" msg="CASB access was
blocked because it contained banned activity." policyid=6 sessionid=63635 srcip=10.1.100.195
dstip=20.190.190.130 srcport=61013 dstport=443 srcintf="port1" srcintfrole="undefined"
dstintf="port3" dstintfrole="undefined" proto=6 action="block" profile="outlook_test"
saasapp="microsoft-outlook" useractivity="microsoft-outlook-login"
activitycategory="activity-control"
```

Example 2: safe search

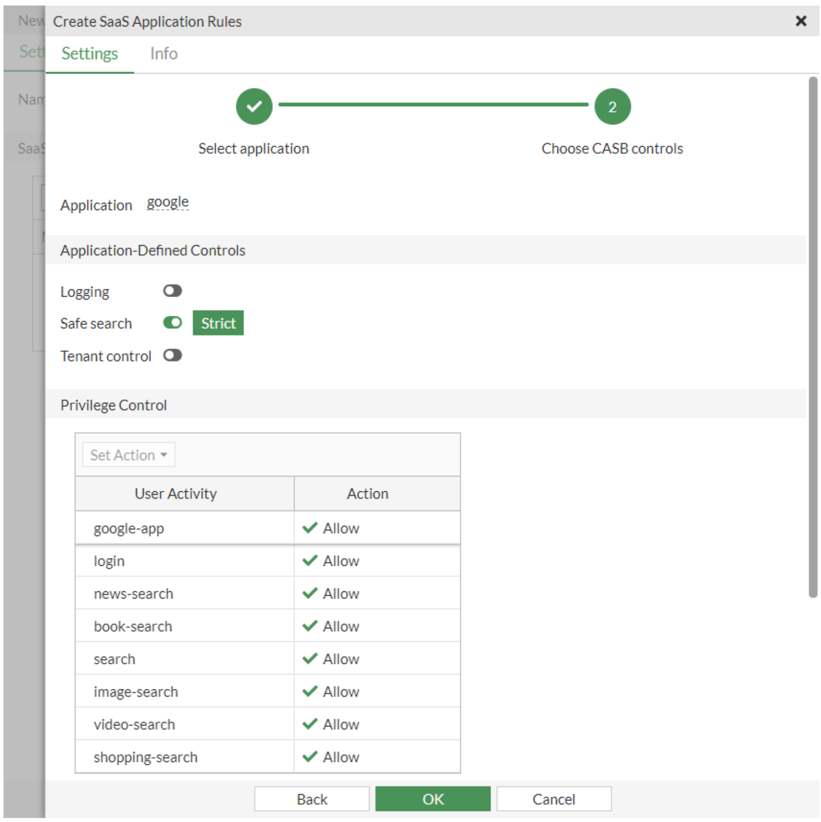
In this example, safe search is configured for Google in the inline CASB profile.

To configure an inline CASB profile with safe search in the GUI:

1. Configure the inline CASB profile:
   a. Go to *Security Profiles > Inline-CASB* and click *Create new*.
   b. Enter a *Name*, such as *google_test*.
   c. In the *SaaS Applications* table, click *Create new*. The *Create SaaS Application Rules* pane opens.
   d. Set the *Application* to *google*, then click *Next*.



e. Enable *Safe search*.



f. Click *OK*.

2. Configure the firewall policy:
   a. Go to *Policy & Objects > Firewall Policy*. Edit an existing policy, or create a new one.
   b. Set the *Inspection Mode* to *Proxy-based*.
   c. In the *Security Profiles* section, enable *Inline-CASB* and select the *google_test* profile.
   d. Set the *SSL Inspection* profile to one that uses deep inspection.
   e. Configure the other settings as needed.
   f. Click *OK*.

To configure an inline CASB profile with safe search in the CLI:

1. Configure the inline CASB profile:

```
config casb profile
    edit "google_test"
        config saas-application
            edit "google"
                set safe-search enable
                set safe-search-control "strict"
            next
        end
    next
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 7
        set name "casb_test_google"
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "ssl"
        set casb-profile "google_test"
        set nat enable
    next
end
```

To test the configuration:

1. Open a browser and attempt to search in Google for content that is considered mature or explicit.
2. The sensitive content is filtered out in the search results.

Sample log:

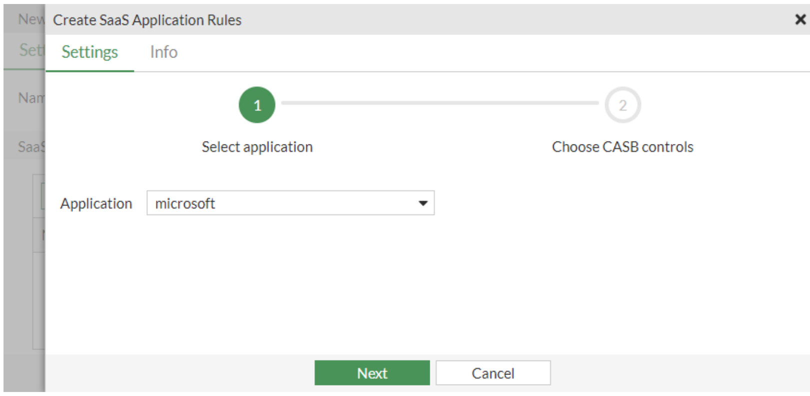
```
1: date=2023-08-18 time=17:01:36 eventtime=1692403295962385271 tz="-0700" logid="2500010002"
type="utm" subtype="casb" eventtype="casb" level="information" vd="vdom1" msg="CASB access
was monitored because it contained activity." policyid=7 sessionid=63774 srcip=10.1.100.195
dstip=142.250.217.98 srcport=61065 dstport=443 srcintf="port1" srcintfrole="undefined"
dstintf="port3" dstintfrole="undefined" proto=6 action="monitor" profile="google_test"
saasapp="google" useractivity="google-safe-search" activitycategory="safe-search-control"
```

Example 3: tenant control

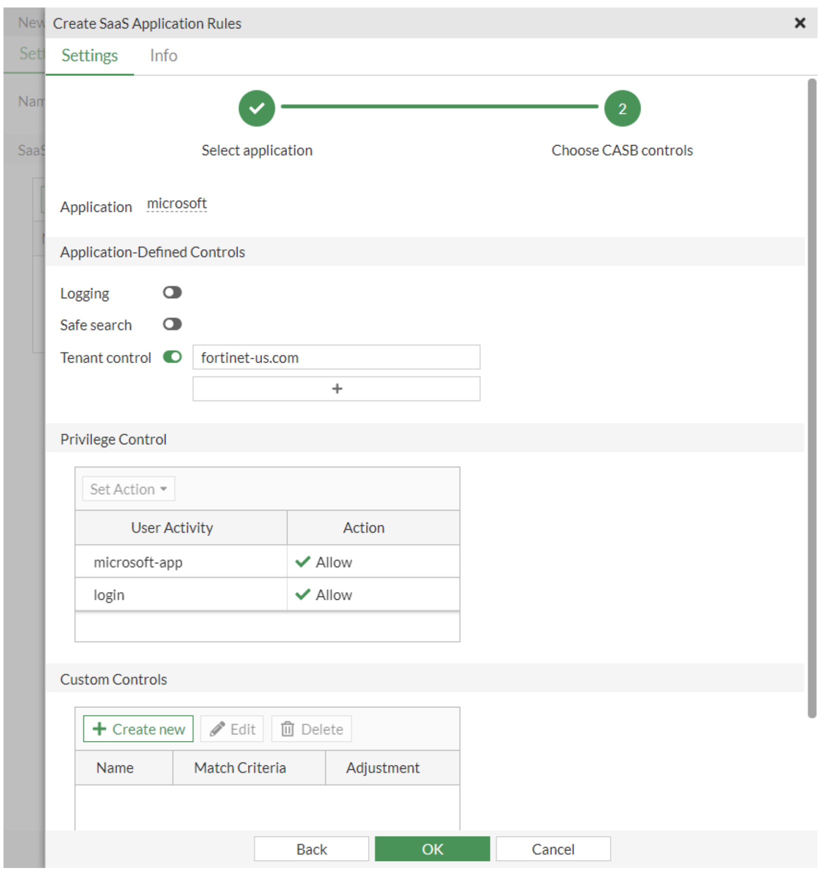
In this example, tenant control is configured for Microsoft in the inline CASB profile for the fortinet-us.com domain.

To configure an inline CASB profile with tenant control in the GUI:

1. Configure the inline CASB profile:
   a. Go to *Security Profiles > Inline-CASB* and click *Create new*.
   b. Enter a *Name*, such as *microsoft_test*.
   c. In the *SaaS Applications* table, click *Create new*. The *Create SaaS Application Rules* pane opens.
   d. Set the *Application* to *microsoft*, then click *Next*.



   e. Enable *Tenant control*. Click the + and enter *fortinet-us.com*.



   f. Click *OK*.
2. Configure the firewall policy:

a. Go to *Policy & Objects* > *Firewall Policy*. Edit an existing policy, or create a new one.
b. Set the *Inspection Mode* to *Proxy-based*.
c. In the *Security Profiles* section, enable *Inline-CASB* and select the *microsoft_test* profile.
d. Set the *SSL Inspection* profile to one that uses deep inspection.
e. Configure the other settings as needed.
f. Click *OK*.

To configure an inline CASB profile with tenant control in the CLI:

1. Configure the inline CASB profile:

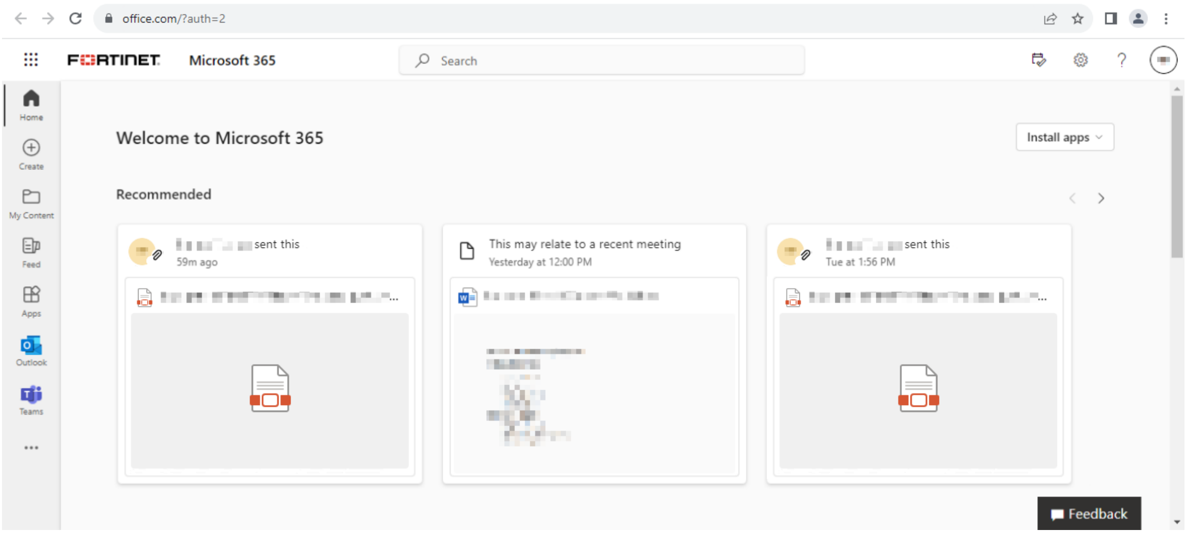
```
config casb profile
    edit "microsoft_test"
        config saas-application
            edit "microsoft"
                set tenant-control enable
                set tenant-control-tenants "fortinet-us.com"
            next
        end
    next
end
```

2. Configure the firewall policy:

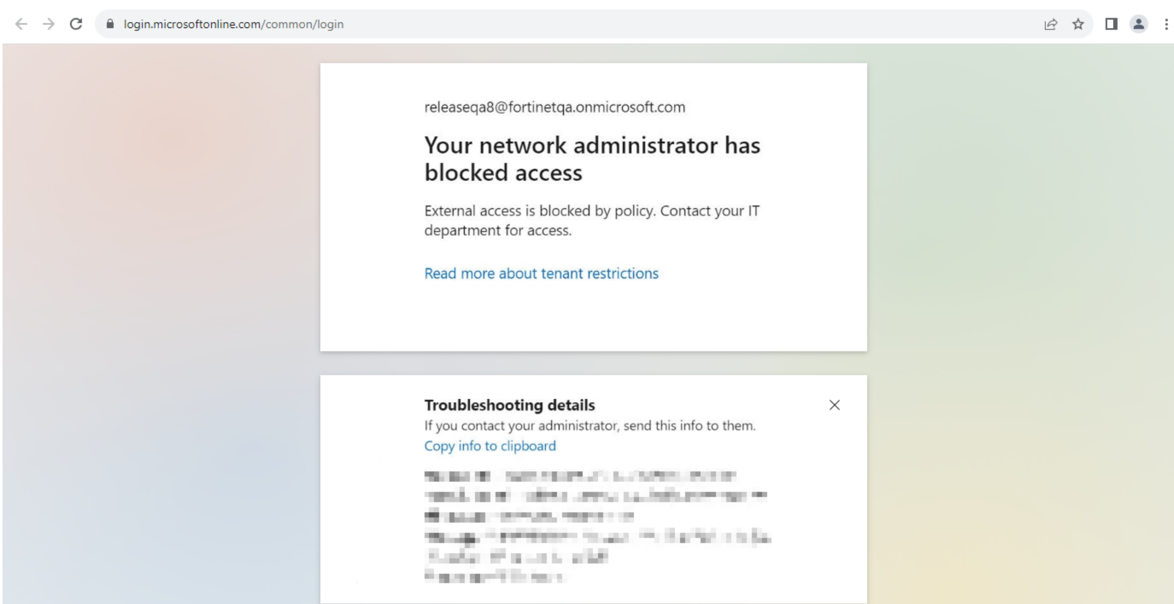
```
config firewall policy
    edit 8
        set name "casb_test_microsoft"
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "ssl"
        set casb-profile "microsoft_test"
        set nat enable
    next
end
```

To test the configuration:

1. Open a browser and attempt to log in to Microsoft Office 365 with a fortinet-us.com account.
2. Since the domain is valid, the user can log in successfully.



3. Attempt to log in to Microsoft Office 365 with another account with a different domain.
4. The domain is invalid. The user is unable to log in, and an error message appears: *Your network administrator has blocked access*.



Sample log:

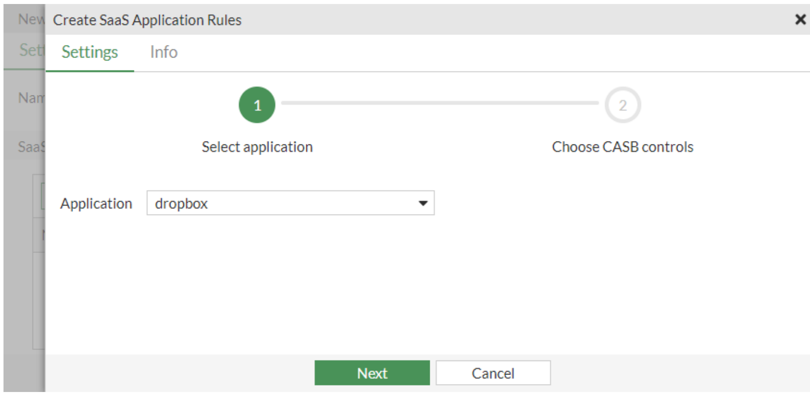
```
1: date=2023-08-18 time=17:09:25 eventtime=1692403765238967943 tz="-0700" logid="2500010002"
type="utm" subtype="casb" eventtype="casb" level="information" vd="vdom1" msg="CASB access
was monitored because it contained activity." policyid=8 sessionid=65108 srcip=10.1.100.195
dstip=20.189.173.15 srcport=61912 dstport=443 srcintf="port1" srcintfrole="undefined"
dstintf="port3" dstintfrole="undefined" proto=6 action="monitor" profile="microsoft_test"
saasapp="microsoft" useractivity="ms-tenant-control" activitycategory="tenant-control"
```

Example 4: UTM bypass

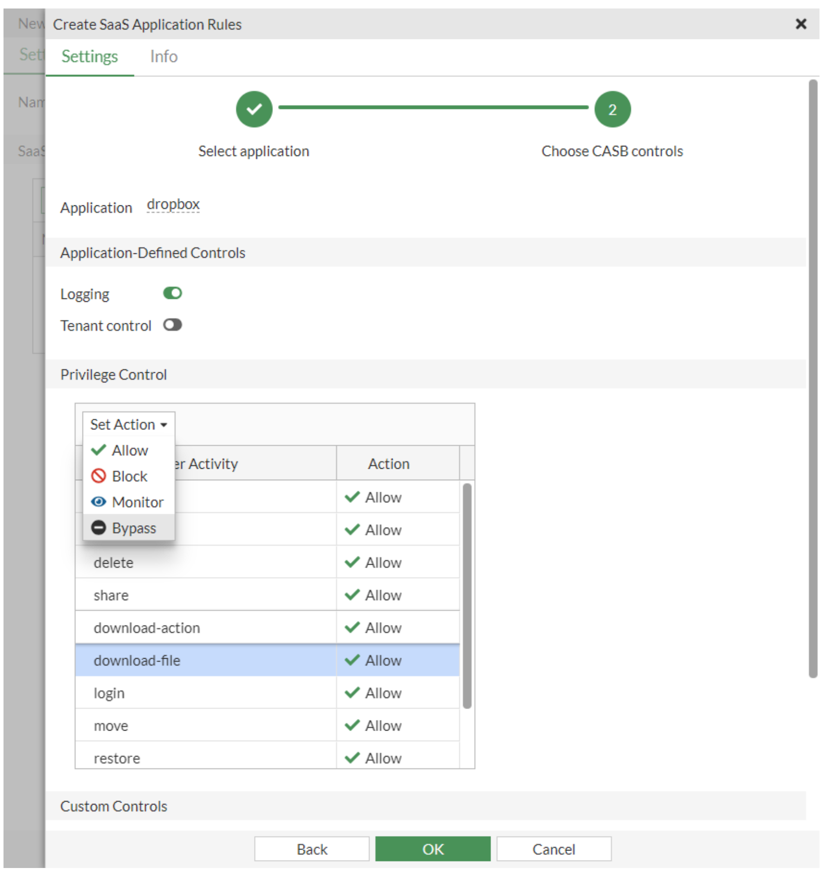
In this example, UTM bypass is configured for Dropbox file downloading in the inline CASB profile.

To configure an inline CASB profile with UTM bypass in the GUI:

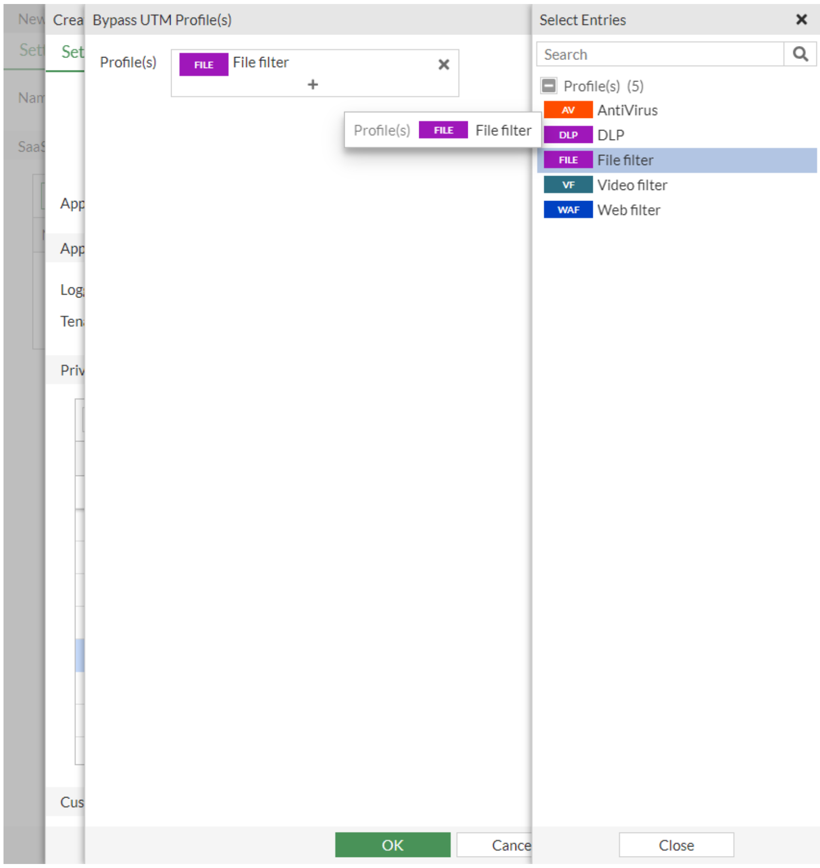
1. Configure the inline CASB profile:
   a. Go to *Security Profiles > Inline-CASB* and click *Create new*.
   b. Enter a *Name*, such as *dropbox_test*.
   c. In the *SaaS Applications* table, click *Create new*. The *Create SaaS Application Rules* pane opens.
   d. Set the *Application* to *dropbox*, then click *Next*.



   e. Enable *Logging*.
   f. In the *Privilege Control* table, select *download-file* and from the *Set Action* dropdown, select *Bypass*.



   The *Bypass UTM Profile(s)* pane opens.
   g. Click the + and set *Profile(s)* to *File Filter*.



h. Click *OK* to save the bypass UTM profile.
i. Click *OK* to save the inline CASB profile
2. Configure the firewall policy:
   a. Go to *Policy & Objects > Firewall Policy*. Edit an existing policy, or create a new one.
   b. Set the *Inspection Mode* to *Proxy-based*.
   c. In the *Security Profiles* section, enable *Inline-CASB* and select the *dropbox_test* profile.
   d. Set the *SSL Inspection* profile to one that uses deep inspection.
   e. Configure the other settings as needed.
   f. Click *OK*.

To configure an inline CASB profile with UTM bypass in the CLI:

1. Configure the inline CASB profile:

```
config casb profile
    edit "dropbox_test"
        config saas-application
            edit "dropbox"
                config access-rule
                    edit "dropbox-download-file"
                        set bypass file-filter
                        set action bypass
                    next
                end
```

```
            next
        end
    next
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 9
        set name "casb_test_dropbox"
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "ssl"
        set casb-profile "dropbox_test"
        set nat enable
    next
end
```

To test the configuration:

1. Open a browser and log in to Dropbox.
2. Attempt to download a file, such as a PDF. The download is successful.

Sample log:

```
1: date=2023-08-18 time=17:15:29 eventtime=1692404129378193492 tz="-0700" logid="2500010001"
type="utm" subtype="casb" eventtype="casb" level="information" vd="vdom1" msg="CASB access
was allowed although it contained activity." policyid=9 sessionid=65452 srcip=10.1.100.195
dstip=162.125.1.15 srcport=62110 dstport=443 srcintf="port1" srcintfrole="undefined"
dstintf="port3" dstintfrole="undefined" proto=6 action="bypass" profile="dropbox_test"
saasapp="dropbox" useractivity="dropbox-download-file" activitycategory="activity-control"
```

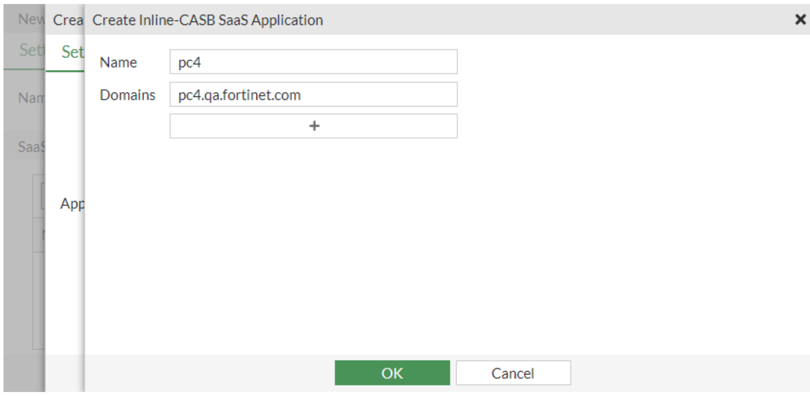
Example 5: customized SaaS application and user activity

In this example, a custom SaaS application is created (pc4) with a custom user action. When a user accesses pc4.qa.fortinet.com/virus, they are redirected to pc4.qa.fortinet.com/testweb/testweb.htm.

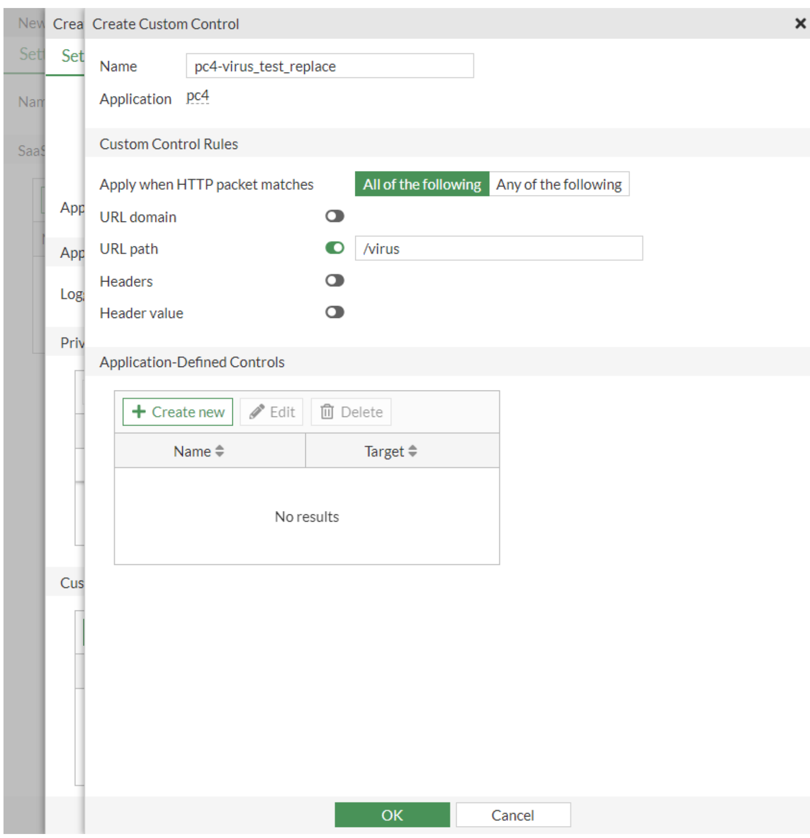
To configure a customized inline CASB profile in the GUI:

1. Configure the inline CASB profile:
    a. Go to *Security Profiles* > *Inline-CASB* and click *Create new*.
    b. Enter a *Name*, such as *custom_test*.
    c. In the *SaaS Applications* table, click *Create new*. The *Create SaaS Application Rules* pane opens.
    d. In the *Application* dropdown, click the + to create a custom entry. The *Create Inline-CASB SaaS Application* pane opens.

- **e.** Enter the *Name* (*pc4*) and *Domains* (*pc4.qa.fortinet.com*), then click *OK*.



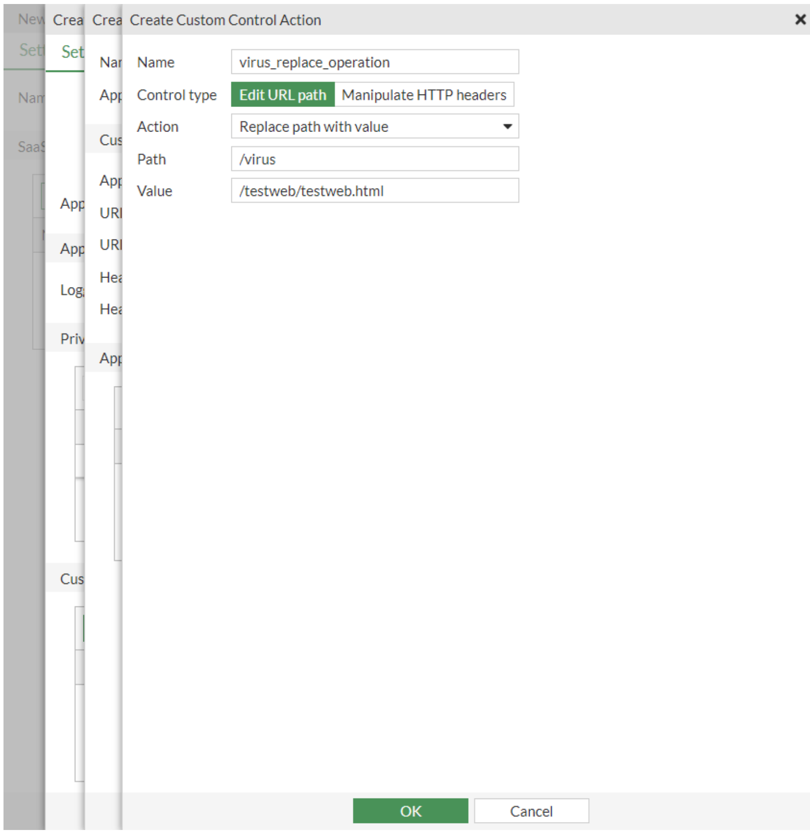
- **f.** Select *pc4* and click *Next*.
- **g.** Configure the custom control and action:
  - **i.** In the *Custom Controls* table, *Create new*. The *Create Custom Control* pane opens.
  - **ii.** Enter a *Name*, such as *pc4-virus_test_replace*.
  - **iii.** Set *Apply when HTTP packet matches* to *All of the following*.
  - **iv.** Enable *URL path* and enter */virus*.



  - **v.** In the *Application-Defined Controls* table, *Create new*. The *Create Custom Control Action* pane opens.
  - **vi.** Enter a *Name*, such as *virus_replace_operation*.
  - **vii.** Set the *Control Type* to *Edit URL path*.
  - **viii.** Set the *Action* to *Replace path with value*.

ix. Set the *Path* to */virus*.

x. Set the *Value* to */testweb/testweb.html*.



xi. Click *OK* to save the custom action.

xii. Click *OK* to save the custom control.

h. Click *OK* to save the application rule.

i. Click *OK* to save the inline CASB profile.

2. Configure the firewall policy:

   a. Go to *Policy & Objects* > *Firewall Policy*. Edit an existing policy, or create a new one.

   b. Set the *Inspection Mode* to *Proxy-based*.

   c. In the *Security Profiles* section, enable *Inline-CASB* and select the *custom_test* profile.

   d. Set the *SSL Inspection* profile to one that uses deep inspection.

   e. Configure the other settings as needed.

   f. Click *OK*.

To configure a customized inline CASB profile in the CLI:

1. Configure the CASB SaaS application:

```
config casb saas-application
    edit "pc4"
        set domains "pc4.qa.fortinet.com"
    next
end
```

2. Configure the CASB user activity:

```
config casb user-activity
    edit "pc4-virus_test_replace"
        set application "pc4"
        set category other
        config match
            edit 1
                config rules
                    edit 1
                        set type path
                        set match-value "/virus"
                    next
                end
            next
        end
        config control-options
            edit "virus_replace_operation"
                config operations
                    edit "virus_replace_operation"
                        set target path
                        set action replace
                        set search-key "/virus"
                        set values "/testweb/testweb.html"
                    next
                end
            next
        end
    next
end
```

3. Configure the inline CASB profile:

```
config casb profile
    edit "custom_test"
        config saas-application
            edit "pc4"
                config custom-control
                    edit "pc4-virus_test_replace"
                        config option
                            edit "virus_replace_operation"
                            next
                        end
                    next
                end
            next
        end
    next
end
```

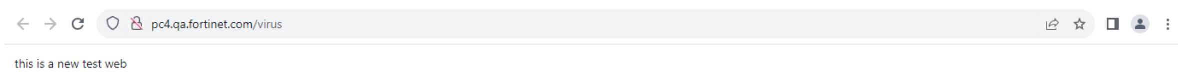
4. Configure the firewall policy:

```
config firewall policy
    edit 10
        set name "casb_test_custom"
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
```

```
            set dstaddr "all"
            set schedule "always"
            set service "ALL"
            set utm-status enable
            set inspection-mode proxy
            set ssl-ssh-profile "ssl"
            set casb-profile "custom_test"
            set nat enable
        next
    end
```

To test the configuration:

1. Open a browser and go to pc4.qa.fortinet.com/virus.
2. Access is redirected to pc4.qa.fortinet.com/testweb/testweb.htm.



Sample log:

```
1: date=2023-08-21 time=08:31:06 eventtime=1692631866382806917 tz="-0700" logid="2500010001"
type="utm" subtype="casb" eventtype="casb" level="information" vd="vdom1" msg="CASB access
was allowed although it contained activity." policyid=10 sessionid=3139 srcip=10.1.100.195
dstip=172.16.200.44 srcport=56774 dstport=80 srcintf="port1" srcintfrole="undefined"
dstintf="port3" dstintfrole="undefined" proto=6
url="http://pc4.qa.fortinet.com/testweb/testweb.html" action="bypass" profile="custom_test"
saasapp="pc4" useractivity="pc4-virus_test_replace" activitycategory="other"
```

Support domain name in XFF with ICAP - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Domain name in XFF with ICAP

The FortiGate can forward additional domain-related information to the ICAP server. Once domain information is gathered from an external authentication server (such as LDAP or an FSSO collector agent), FortiOS incorporates this domain information in `WinNT://DOMAIN/Username` format and forwards it to the ICAP server.

Basic ICAP configuration

The ICAP server and profile are configured on the FortiGate. The ICAP profile's header settings uses the `WinNT://$domain/$user` variable for the user information provided by the remote authentication server.

To configure the ICAP settings:

1. Configure the ICAP server:

```
config icap server
    edit "content-filtration-server4"
```

```
        set ip-address 10.1.100.41
        set max-connections 200
    next
end
```

2. Configure the ICAP profile:

```
config icap profile
    edit "Prop-Content-Filtration"
        set request enable
        set response enable
        set streaming-content-bypass enable
        set request-server "content-filtration-server4"
        set response-server "content-filtration-server4"
        set request-path "/proprietary_code/content-filter/"
        set response-path "/proprietary_code/content-filter/"
        set methods delete get head options post put trace other
        config icap-headers
            edit 1
                set name "X-Authenticated-User"
                set content "WinNT://$domain/$user"
            next
        end
    next
end
```

3. Configure the firewall policy:

```
config firewall policy
    edit 4
        set name "icap_filter3"
        set srcintf "port10"
        set dstintf "port9"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set icap-profile "Prop-Content-Filtration"
        set logtraffic all
        set nat enable
        set groups "ldap group" "AD-group"
    next
end
```

LDAP example

In this example, an AD LDAP server and remote user group are configured. When successful user authentication occurs, FortiOS retrieves all the user information (such as the domain name) from the UserPrincipalName attribute. A packet capture is used to compare the user and domain information before and after authentication in the ICAP REQMOD message.

To configure the LDAP authentication:

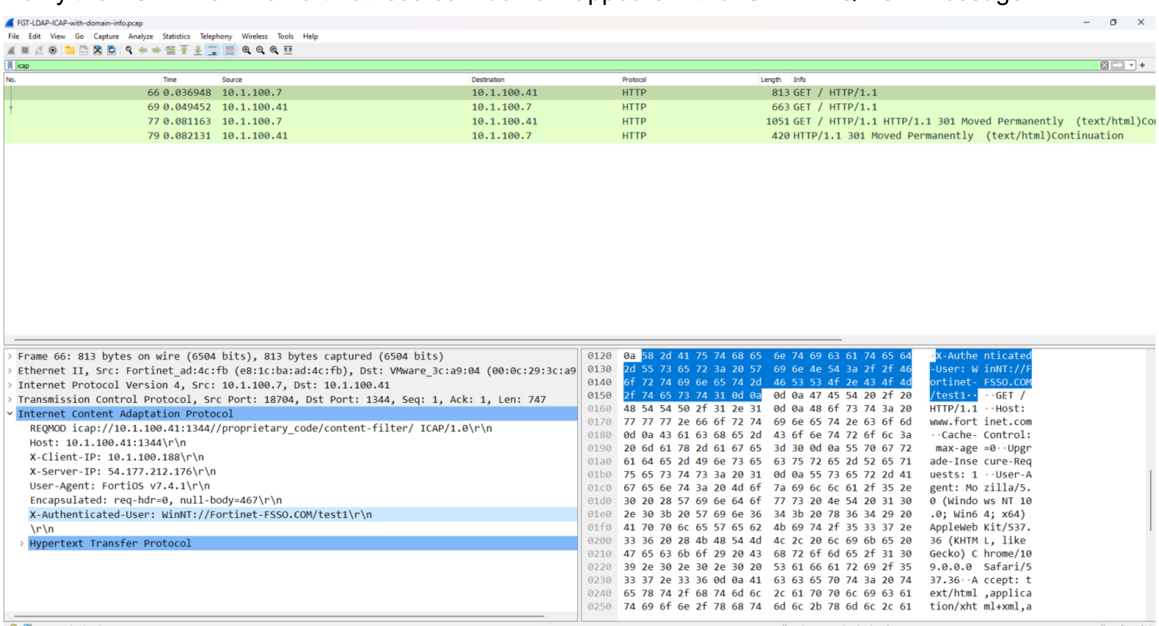
1. Configure the LDAP server:

```
config user ldap
    edit "AD-ldap"
        set server "10.1.100.131"
        set cnid "cn"
        set dn "dc=fortinet-fsso,dc=com"
        set type regular
        set username "cn=Administrator,cn=users,dc=fortinet-fsso,dc=com"
        set password **********
    next
end
```

2. Configure the LDAP user group:

```
config user group
    edit "ldap group"
        set member "AD-ldap"
        config match
            edit 1
                set server-name "AD-ldap"
                set group-name "CN=group1,OU=Testing,DC=Fortinet-FSSO,DC=COM"
            next
            edit 2
                set server-name "AD-ldap"
                set group-name "CN=group2,OU=Testing,DC=Fortinet-FSSO,DC=COM"
            next
        end
    next
end
```

3. Start local traffic dump between the FortiGate and ICAP server before a user authenticates and save it in a PCAP file.
4. Verify the PCAP file. The Fortinet-fsso.com domain appears in the ICAP REQMOD message.



5. Optionally, run the following command to verify WAD debugs:

```
# diagnose wad debug enable category icap
```

FSSO example

In this example, a local FSSO agent and remote user group are configured. When successful user authentication occurs, FortiOS retrieves all the user information (such as the domain name). A packet capture is used to compare the user and domain information before and after authentication in the ICAP REQMOD message.

To configure the FSSO authentication:

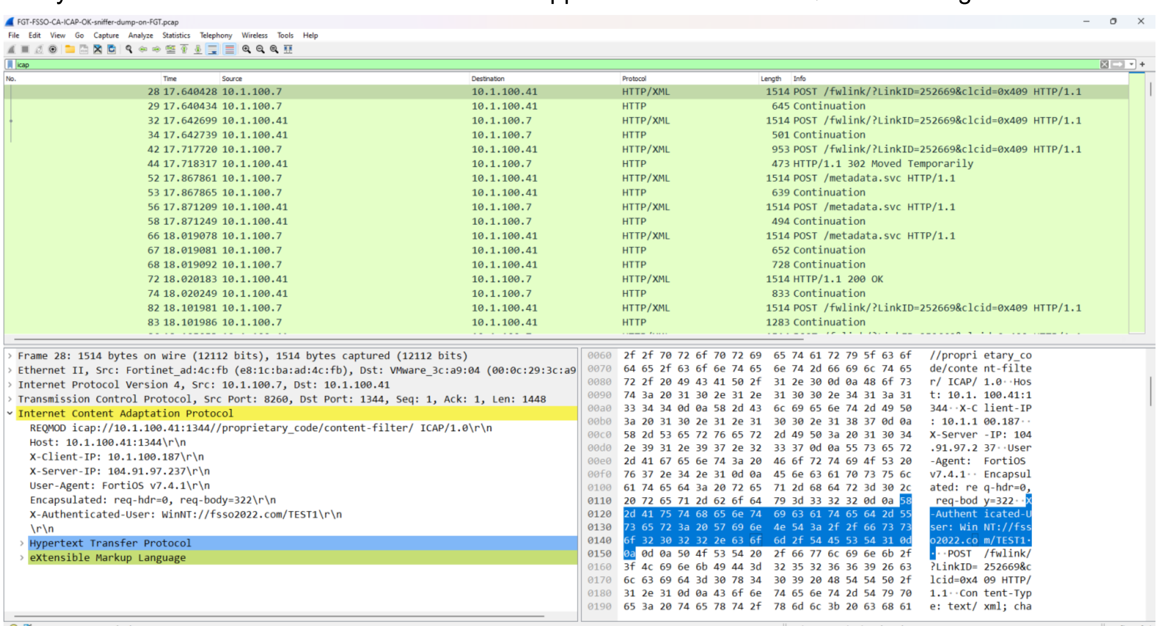
1. Configure the FSSO agent:

```
config user fsso
    edit "AD-fsso"
        set server "10.1.100.199"
        set password **********
    next
end
```

2. Configure the FSSO user group:

```
config user group
    edit "AD-group"
        set group-type fsso-service
        set member "FORTINET-FSSO/GROUP1" "FORTINET-FSSO/GROUP2"
    next
end
```

3. Start local traffic dump between the FortiGate and ICAP server before a user authenticates and save it in a PCAP file.
4. Verify the PCAP file. The fsso2022.com domain appears in the ICAP REQMOD message.



5. Optionally, verify the FSSO log file and search for the `get_dns_domain` lines:

```
...
06/20/2023 14:58:58 [ 1484] FortiGate connection accepted, auth OK.
06/20/2023 14:58:58 [ 1484] FortiGate:FG4H1E5819900343-root connected on socket (2004).
06/20/2023 14:58:58 [ 1484] send AUTH, len:26
06/20/2023 14:58:58 [ 1484] ready to read from socket
06/20/2023 14:58:58 [ 1484] Bytes received from FortiGate: 26
06/20/2023 14:58:58 [ 1484] process AD_INFO
06/20/2023 14:58:58 [ 1484] group filter received from FortiGate: len:26
06/20/2023 14:58:58 [ 1484] packet seq:2
06/20/2023 14:58:58 [ 1484] ad info flag:1
06/20/2023 14:58:58 [ 1484] FGT sends empty group list
06/20/2023 14:58:58 [ 1484] ready to read from socket
06/20/2023 14:58:58 [ 1484] Bytes received from FortiGate: 36
06/20/2023 14:58:58 [ 1484] packet seq:3
06/20/2023 14:58:58 [ 1484] option:00000001 ref point:00000000
06/20/2023 14:58:58 [ 1484] toFGT set to:1
06/20/2023 14:58:58 [ 1484] get_dns_domain_name:177 enable_dns_domain_name:1, netbios_
domain_name:FSSO2022
06/20/2023 14:58:58 [ 1484] get_dns_domain_name:185 dns_domain_name:FSSO2022.com
06/20/2023 14:58:58 [ 1484] send LOGON_INFO, len:187
06/20/2023 14:58:58 [ 1484] send_to_FGT() called:sock:2004 sendbuf:198f4498 sendlen:187
```

Enhance the video filter profile with a new level of customization and control - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- Filtering based on title
- Filtering based on description
- Configuring a video filter keyword list

Video filter profiles include a new level of customization and control with two keyword-based filters for video titles and descriptions that offer AND/OR logic options. Users can prioritize configured filters, and manage all categories and channels that match the filters using the new *Any* option.

By default, video filter profiles include an implicit rule set to allow the video. If a video does not match any of the other filters, it is subject to this implicit rule and passes through.

Unicode emoji character code is currently not supported for the title and description filters.

The YouTube API key must be configured to use this feature. Otherwise, the title and description filters will not retrieve the video information and bypass the traffic.

To configure the YouTube API key in the GUI:

1. Go to *Security Profiles > Video Filter* and select the *Video Filter Settings* tab.
2. Click the + to add an API key.
3. Click *OK*.

To configure the YouTube API key in the CLI:

```
config videofilter youtube-key
    edit <id>
        set key <string>
    next
end
```

This topic contains five video filter use cases:

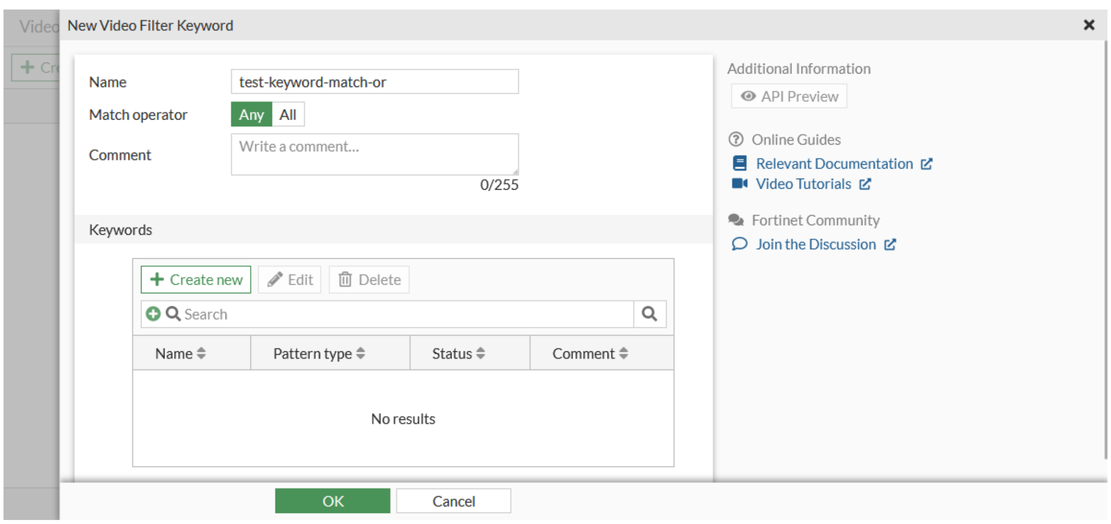
- Example 1: blocking a video with a title containing the keywords 'API' or 'game'
- Example 2: blocking a video with a description containing the keywords 'API' and 'testing'
- Example 3: allowing a specific video by filtering the title while blocking others
- Example 4: allowing a specific video by filtering the description while blocking others
- Example 5: disabling a keyword in the keyword list

Example 1: blocking a video with a title containing the keywords 'API' or 'game'

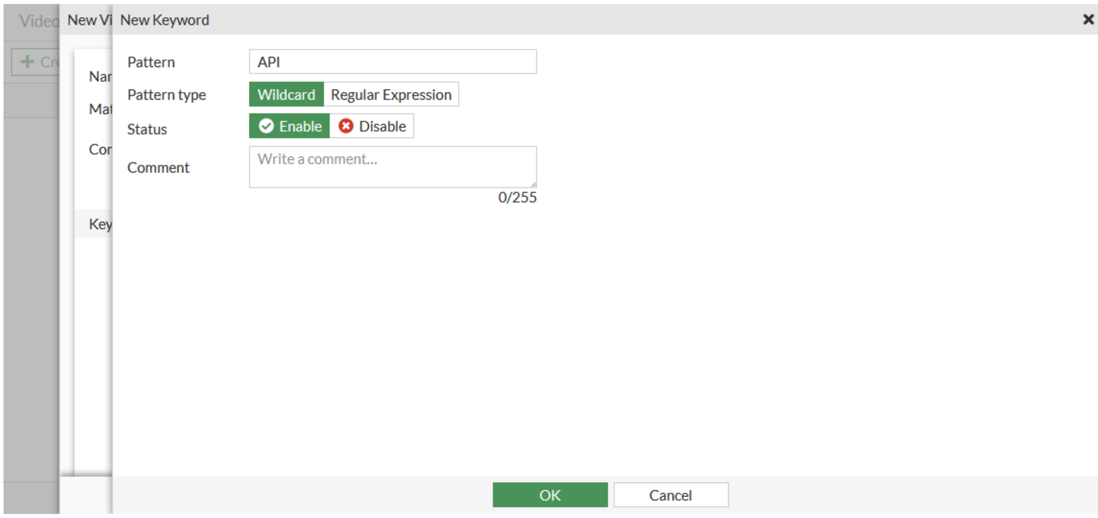
In this example, videos are blocked that contain the keywords 'API' or 'game', so the keywords filter uses the *Any* match operator.

To configure the video filter profile in the GUI:

1. Configure the video filter keyword list:
   a. Go to *Security Profiles > Video Filter*, select the *Video Filter Keyword* tab, and click *Create new*.
   b. Enter a name (*test-keyword-match-or*) and set *Match operator* to *Any*.



   c. In the *Keywords* table, click *Create new*.
   d. Configure the API keyword with the following settings:
      i. In the *Pattern* field, enter *API*.
      ii. Set the *Pattern type* to *Wildcard*.



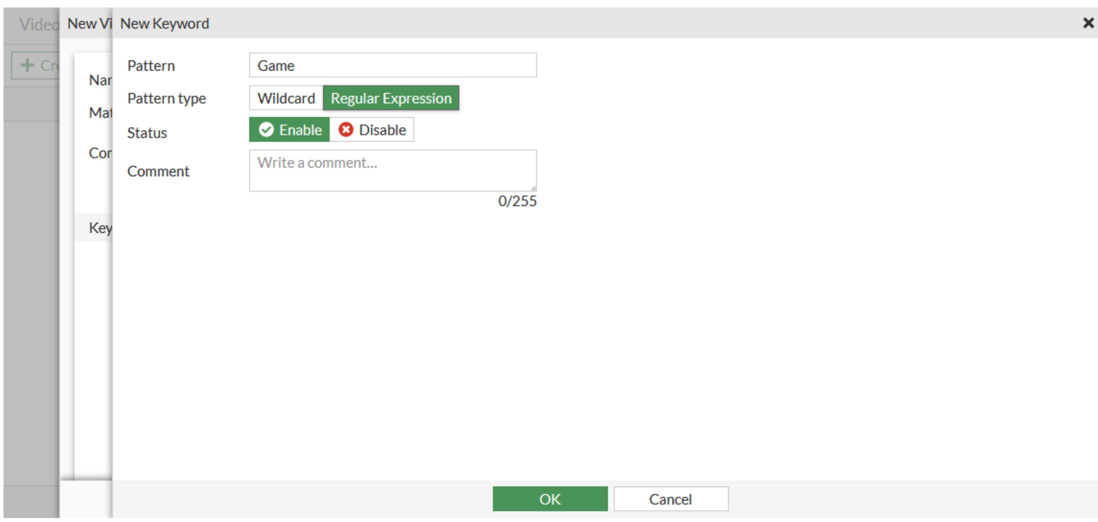
iii. Click *OK*.

e. Click *Create new*.

f. Configure the game keyword with the following settings:

  i. In the *Pattern* field, enter *Game*.

  ii. Set the *Pattern type* to *Regular Expression*.



  iii. Click *OK*.

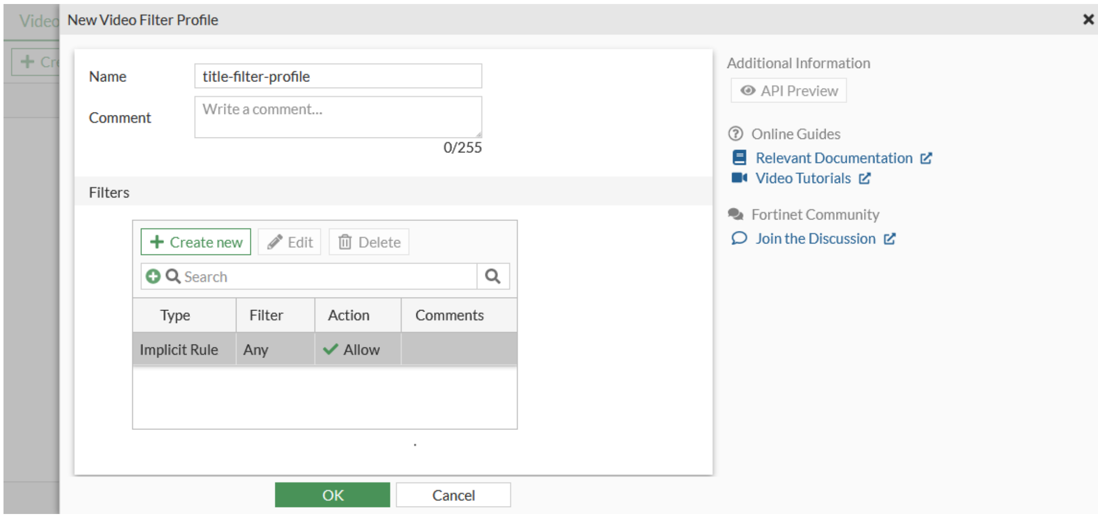
g. Click *OK* to save the keyword list.

2. Configure the video filter profile:

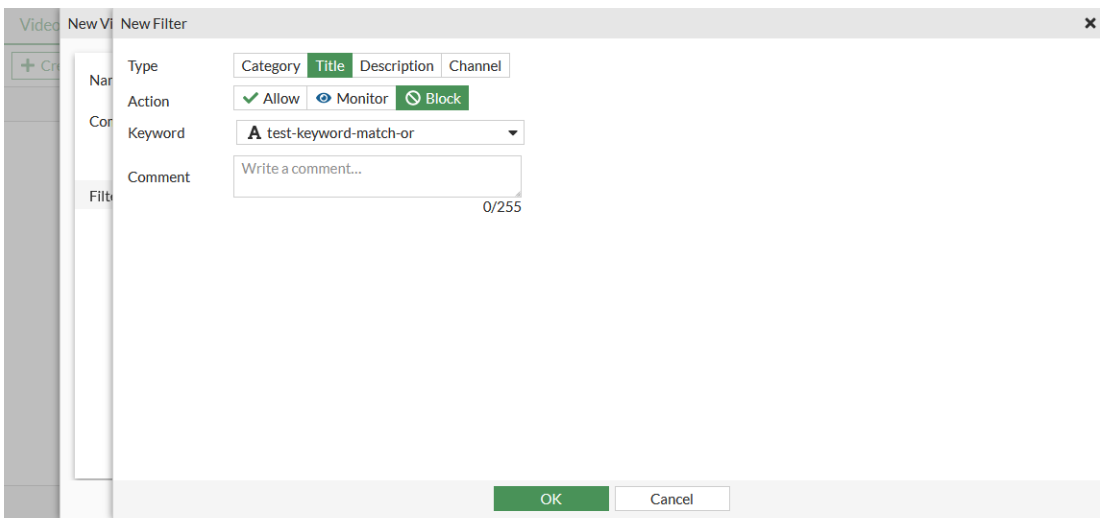
  a. Go to *Security Profiles > Video Filter*, select the *Video Filter Profile* tab, and click *Create new*.

  b. Enter a name (*title-filter-profile*).

  c. In the *Filters* table, click *Create new*.



d. Configure the filter with the following settings:
   i. Set the *Type* to *Title*.
   ii. Set the *Action* to *Block*.
   iii. Set the *Keyword* to *test-keyword-match-or*.



   iv. Click *OK*.

e. Click *OK* to save the video filter profile.

3. Apply the video filter in a firewall policy.

To configure the video filter profile in the CLI:

1. Configure the video filter keyword list:

```
config videofilter keyword
    edit 1
        set name "test-keyword-match-or"
        set match or
        config word
            edit "API"
                set pattern-type wildcard
```

```
                set status enable
            next
            edit "Game"
                set pattern-type regex
                set status enable
            next
        end
    next
end
```

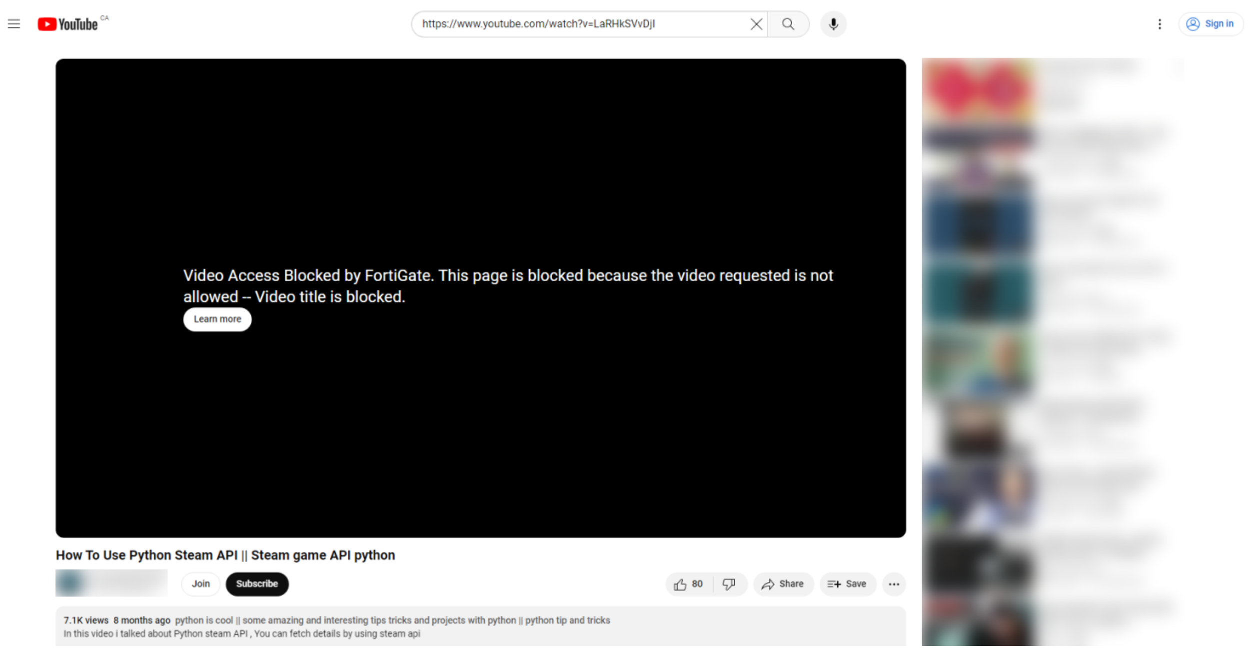
2. Configure the video filter profile:

```
config videofilter profile
    edit "title-filter-profile"
        config filters
            edit 1
                set type title
                set keyword 1
                set action block
                set log enable
            next
        end
    next
end
```

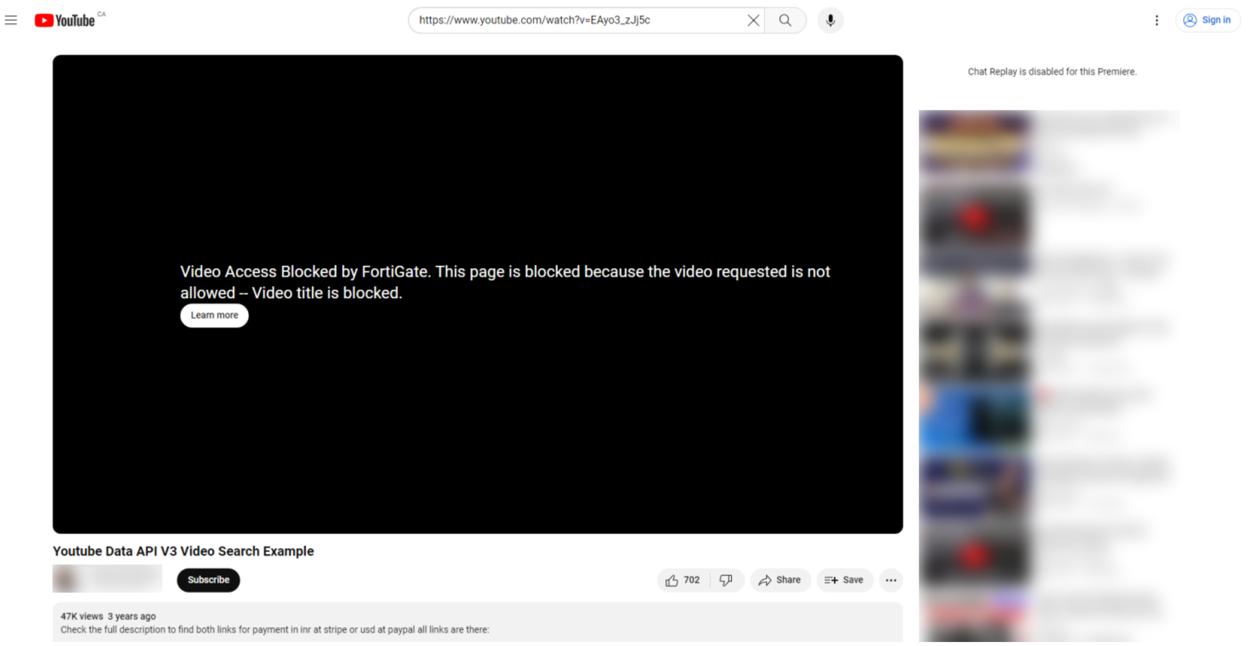
3. Apply the video filter in a firewall policy.

To verify the configuration:

1. From a client, search for a video in YouTube named "How To Use Python Steam API || Steam game API python". The video is blocked.



2. Search for a video in YouTube named "Youtube Data API V3 Video Search Example". The video is blocked.



Sample logs:

```
6: date=2023-11-24 time=09:51:30 eventtime=1700848289598975941 tz="-0800" logid="0350013712"
type="utm" subtype="webfilter" eventtype="unknown" level="warning" vd="vdom1" msg="Video
title is blocked." policyid=1 poluuid="19841eb8-841c-51ee-7047-6a6860eb3522"
sessionid=384813810 srcip=10.1.100.141 dstip=142.251.33.110 srcport=21473 dstport=443
srcintf="port2" srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" proto=6
httpmethod="GET" service="HTTPS" action="blocked" videoinfosource="API" profile="title-
filter-profile" videoid="LaRHkSVvDjI" videotitle="How To Use Python Steam API  || Steam game
API python" hostname="www.youtube.com" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KH" url="https://www.youtube.com/watch?v=LaRHkSVvDjI"
```

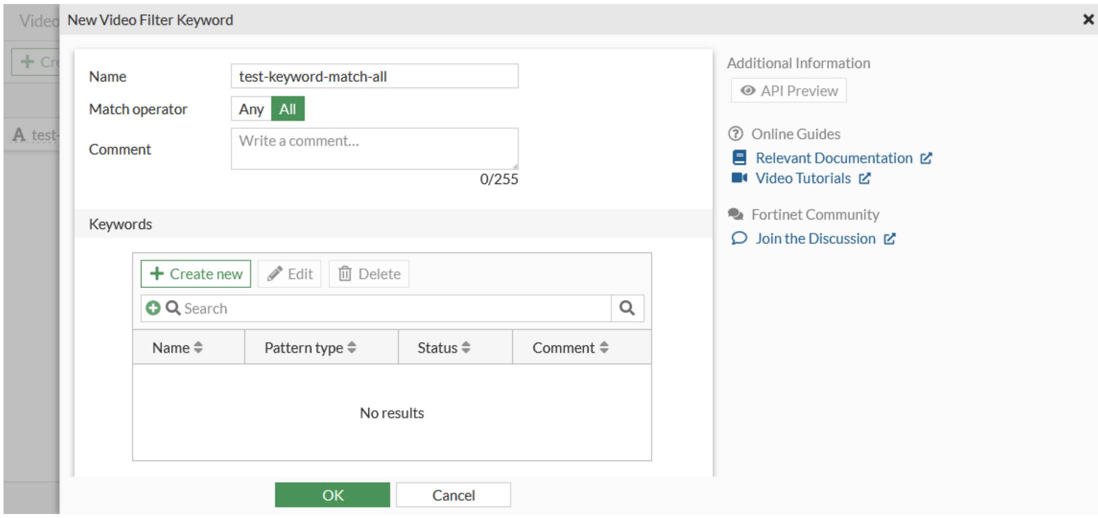
```
17: date=2023-11-23 time=19:30:59 eventtime=1700796659106881476 tz="-0800"
logid="0350013712" type="utm" subtype="webfilter" eventtype="unknown" level="warning"
vd="vdom1" msg="Video title is blocked." policyid=1 poluuid="19841eb8-841c-51ee-7047-
6a6860eb3522" sessionid=384811679 srcip=10.1.100.141 dstip=142.251.215.238 srcport=15058
dstport=443 srcintf="port2" srcintfrole="undefined" dstintf="port1" dstintfrole="undefined"
proto=6 httpmethod="GET" service="HTTPS" action="blocked" videoinfosource="API"
profile="title-filter-profile" videoid="EAyo3_zJj5c" videotitle="Youtube Data API V3 Video
Search Example" hostname="www.youtube.com" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KH" url="https://www.youtube.com/watch?v=EAyo3_zJj5c"
```

Example 2: blocking a video with a description containing the keywords 'API' and 'testing'

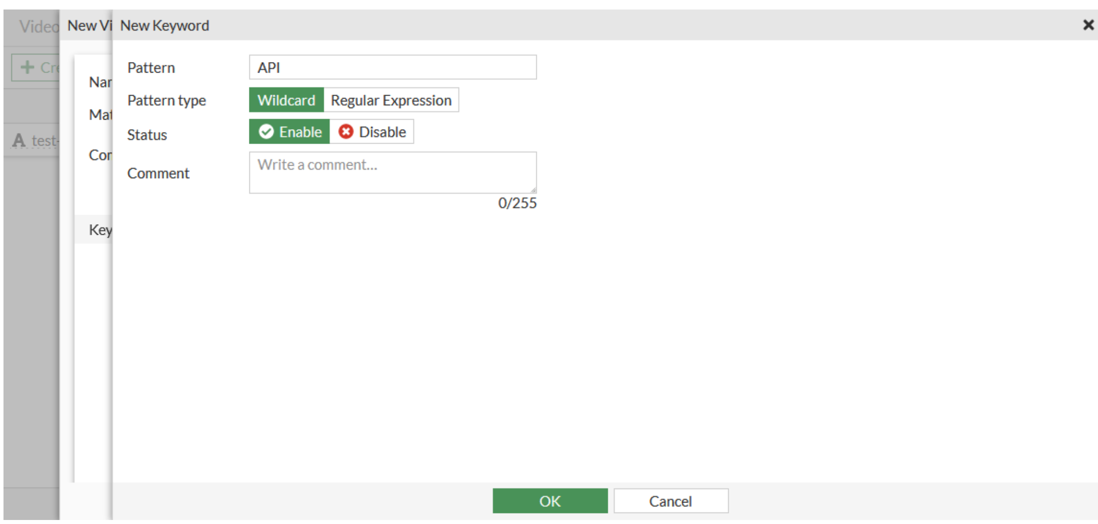
In this example, videos are blocked where the description contains the keywords 'API' and 'testing', so the keywords filter uses the *All* match operator. The description filter supports the first 100 characters of the video description.

To configure the video filter profile in the GUI:

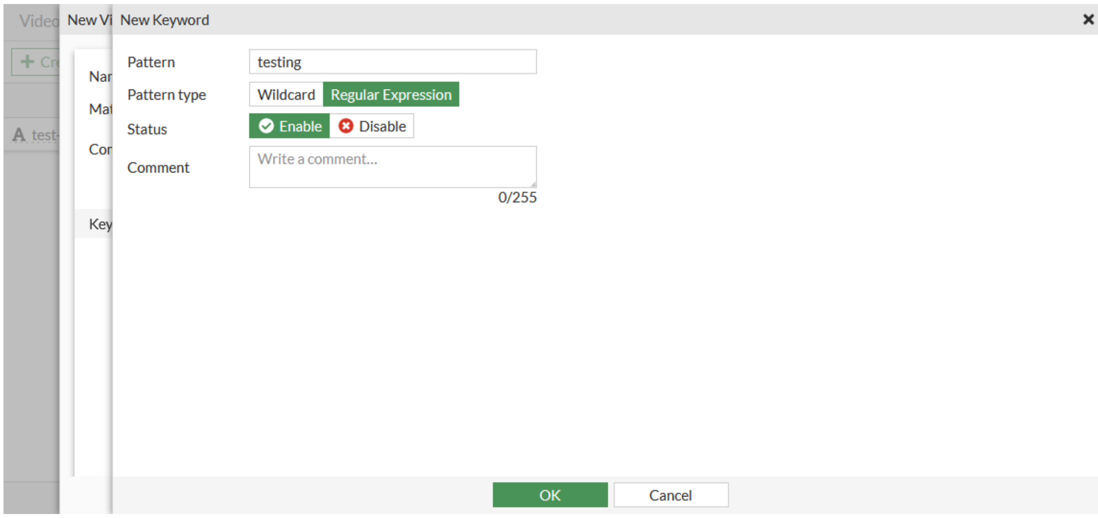
1. Configure the video filter keyword list:
   a. Go to *Security Profiles > Video Filter*, select the *Video Filter Keyword* tab, and click *Create new*.
   b. Enter a name (*test-keyword-match-all*) and set *Match operator* to *All*.



   c. In the *Keywords* table, click *Create new*.
   d. Configure the API keyword with the following settings:
      i. In the *Pattern* field, enter *API*.
      ii. Set the *Pattern type* to *Wildcard*.



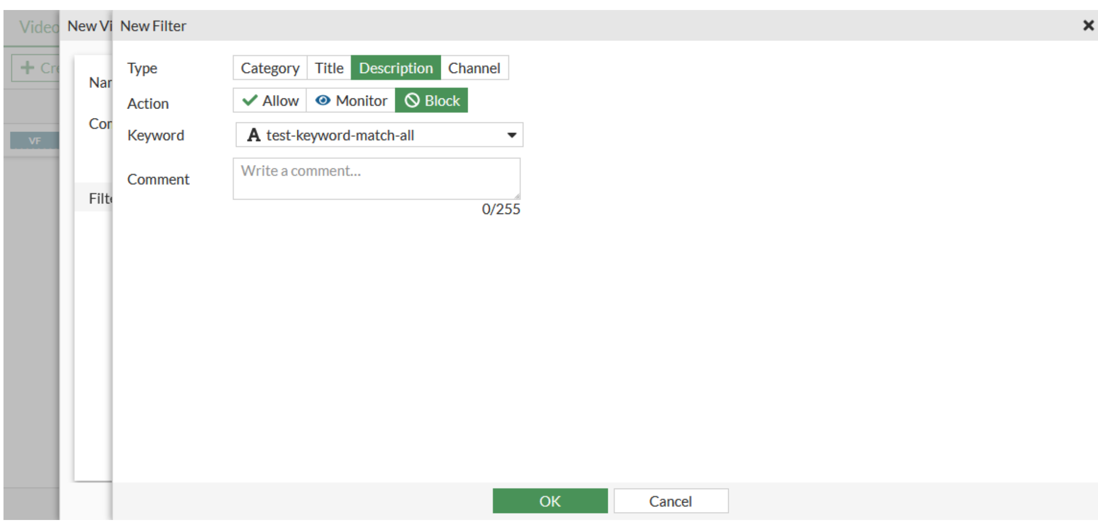
      iii. Click *OK*.
   e. Click *Create new*.
   f. Configure the testing keyword with the following settings:
      i. In the *Pattern* field, enter *testing*.
      ii. Set the *Pattern type* to *Regular Expression*.



iii. Click *OK*.

g. Click *OK* to save the keyword list.

2. Configure the video filter profile:
   a. Go to *Security Profiles* > *Video Filter*, select the *Video Filter Profile* tab, and click *Create new*.
   b. Enter a name (*test-description-filter*).
   c. In the *Filters* table, click *Create new*.
   d. Configure the filter with the following settings:
      i. Set the *Type* to *Description*.
      ii. Set the *Action* to *Block*.
      iii. Set the *Keyword* to *test-keyword-match-all*.



      iv. Click *OK*.
   e. Click *OK* to save the video filter profile.
3. Apply the video filter in a firewall policy.

To configure the video filter profile in the CLI:

1. Configure the video filter keyword list:

```
config videofilter keyword
    edit 2
        set name "test-keyword-match-all"
        set match and
        config word
            edit "API"
                set pattern-type wildcard
                set status enable
            next
            edit "testing"
                set pattern-type regex
                set status enable
            next
        end
    next
end
```

2. Configure the video filter profile:

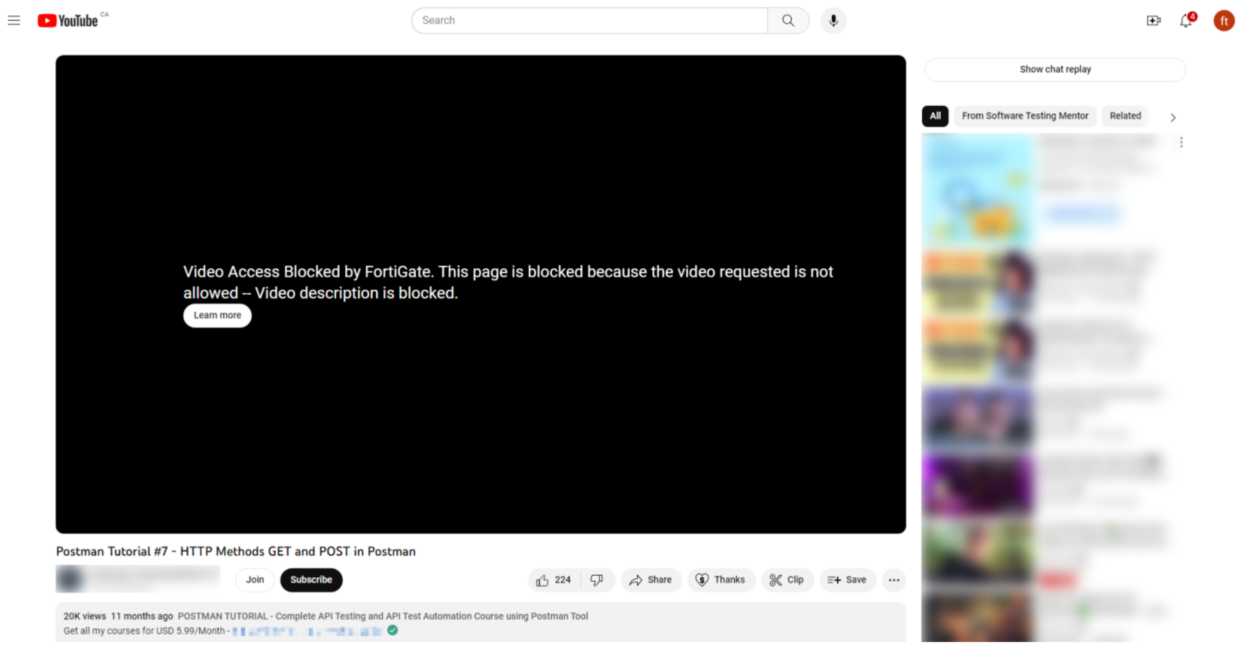
```
config videofilter profile
    edit "test-description-filter"
        config filters
            edit 1
                set type description
                set keyword 2
                set action block
                set log enable
            next
        end
    next
end
```

3. Apply the video filter in a firewall policy.

To verify the configuration:

1. From a client, search for a video in YouTube named "Postman Tutorial #7 - HTTP Methods GET and POST in Postman". The description contains the text, "POSTMAN TUTORIAL - Complete API Testing and API Test

Automation Course using Postman Tool...", so the video is blocked.



Sample log:

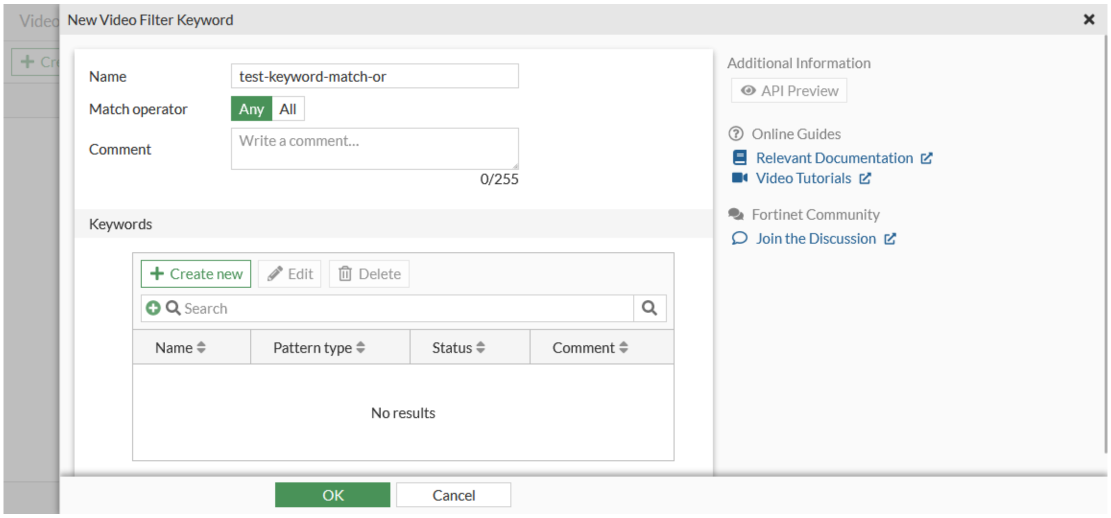
4: date=2023-11-24 time=16:08:51 eventtime=1700870931146681788 tz="-0800" logid="0351013728" type="utm" subtype="webfilter" eventtype="unknown" level="warning" vd="vdom1" **msg="Video description is blocked."** policyid=1 poluuid="090ca600-83e4-51ee-158a-a920fcf8f892" sessionid=100211 srcip=10.1.100.141 dstip=142.250.69.206 srcport=24948 dstport=443 srcintf="port2" srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" proto=6 httpmethod="GET" service="HTTPS" **action="blocked"** videoinfosource="API" profile="test-description-filter" videoid="pUGmhtqVJRk" videodesc="Get all my courses for USD 5.99/Month - https://bit.ly/all-c..." hostname="www.youtube.com" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KH" url="https://www.youtube.com/watch?v=pUGmhtqVJRk"

Example 3: allowing a specific video by filtering the title while blocking others

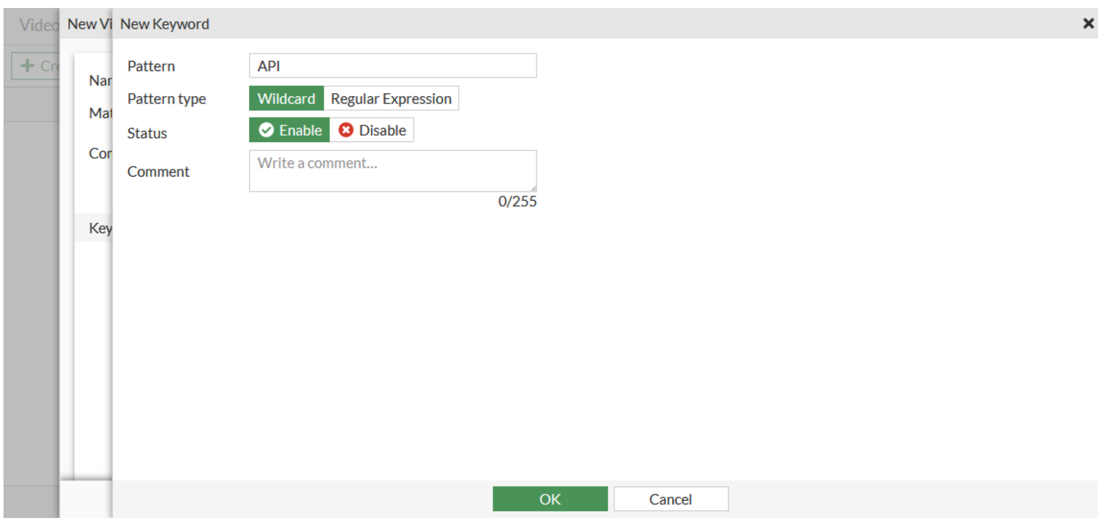
In this example, specific videos are allowed using title filtering while blocking others. The video filter profile contains two filters. The first filter uses a keyword list (monitor and allow the keywords 'API' or 'game' with the *Any* match operator). The second filter uses a category filter to block any unmatched videos by title filtering.

To configure the video filter profile in the GUI:

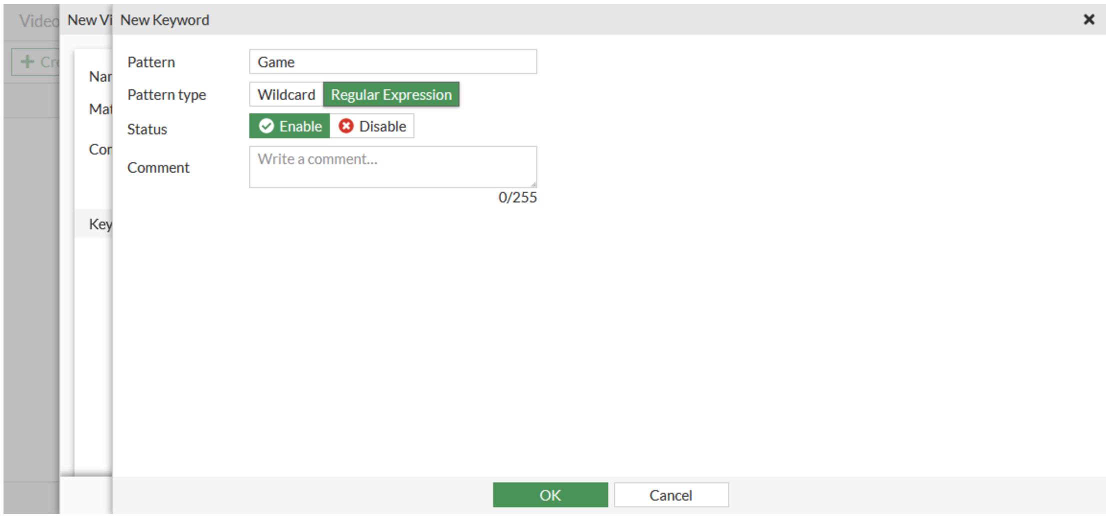
1. Configure the video filter keyword list:
   a. Go to *Security Profiles > Video Filter*, select the *Video Filter Keyword* tab, and click *Create new*.
   b. Enter a name (*test-keyword-match-or*) and set *Match operator* to *Any*.



c. In the *Keywords* table, click *Create new*.
d. Configure the API keyword with the following settings:
   i. In the *Pattern* field, enter *API*.
   ii. Set the *Pattern type* to *Wildcard*.



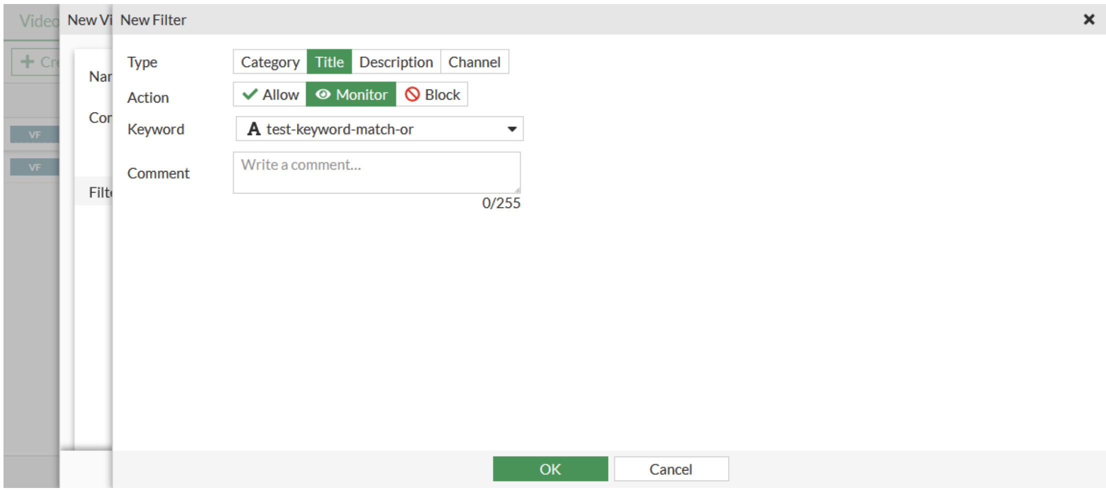
   iii. Click *OK*.
e. Click *Create new*.
f. Configure the game keyword with the following settings:
   i. In the *Pattern* field, enter *Game*.
   ii. Set the *Pattern type* to *Regular Expression*.



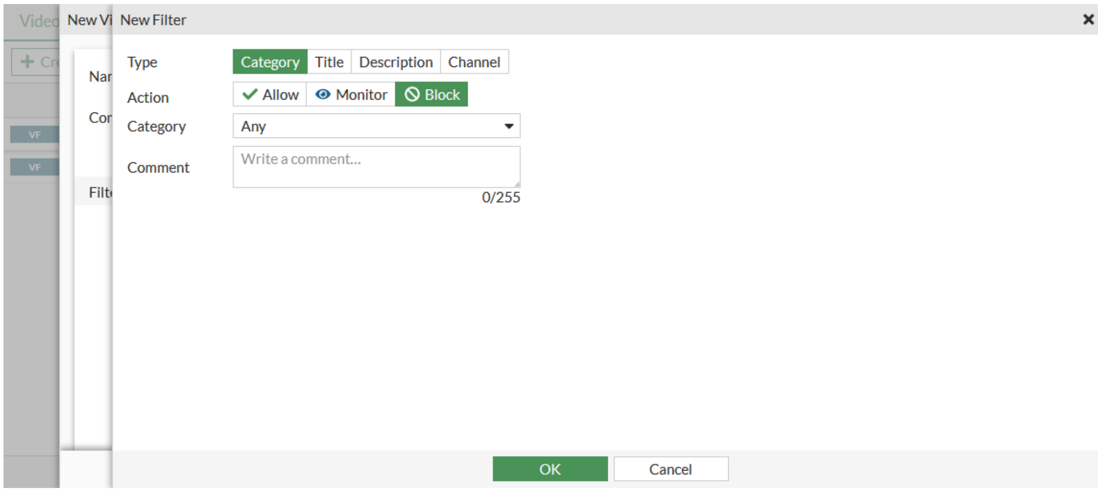
   iii. Click *OK*.

g. Click *OK* to save the keyword list.

2. Configure the video filter profile:
   a. Go to *Security Profiles* > *Video Filter*, select the *Video Filter Profile* tab, and click *Create new*.
   b. Enter a name (*allow-specific-title*).
   c. In the *Filters* table, click *Create new*.
   d. Configure the first filter with the following settings:
      i. Set the *Type* to *Title*.
      ii. Set the *Action* to *Monitor*.
      iii. Set the *Keyword* to *test-keyword-match-or*.



      iv. Click *OK*.
   e. Configure the second filter with the following settings:
      i. Set the *Type* to *Category*.
      ii. Set the *Action* to *Block*.
      iii. Set the *Category* to *Any*.



iv. Click *OK*.

f. Click *OK* to save the video filter profile.

3. Apply the video filter in a firewall policy.

To configure the video filter profile in the CLI:

1. Configure the video filter keyword list:

```
config videofilter keyword
    edit 1
        set name "test-keyword-match-or"
        set match or
        config word
            edit "API"
                set pattern-type wildcard
                set status enable
            next
            edit "Game"
                set pattern-type regex
                set status enable
            next
        end
    next
end
```

2. Configure the video filter profile:

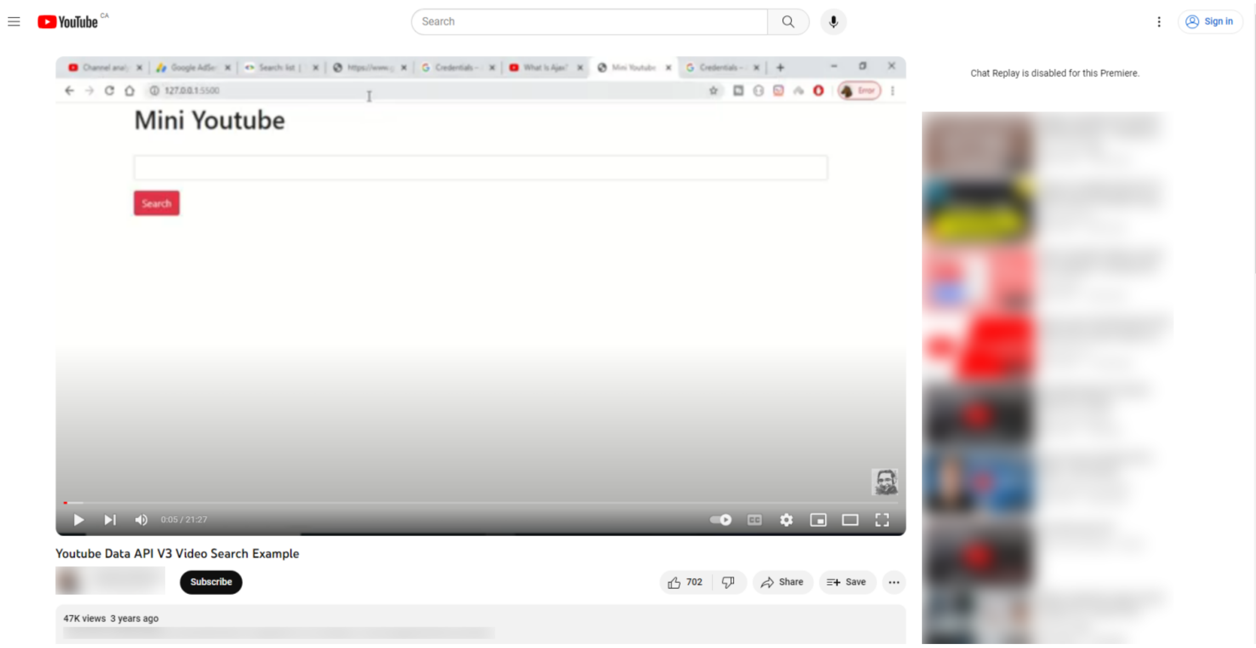
```
config videofilter profile
    edit "allow-specific-title"
        config filters
            edit 1
                set type title
                set keyword 1
                set action monitor
                set log enable
            next
            edit 2
                set type category
                set category "any"
```

```
                set action block
                set log enable
            next
        end
    next
end
```

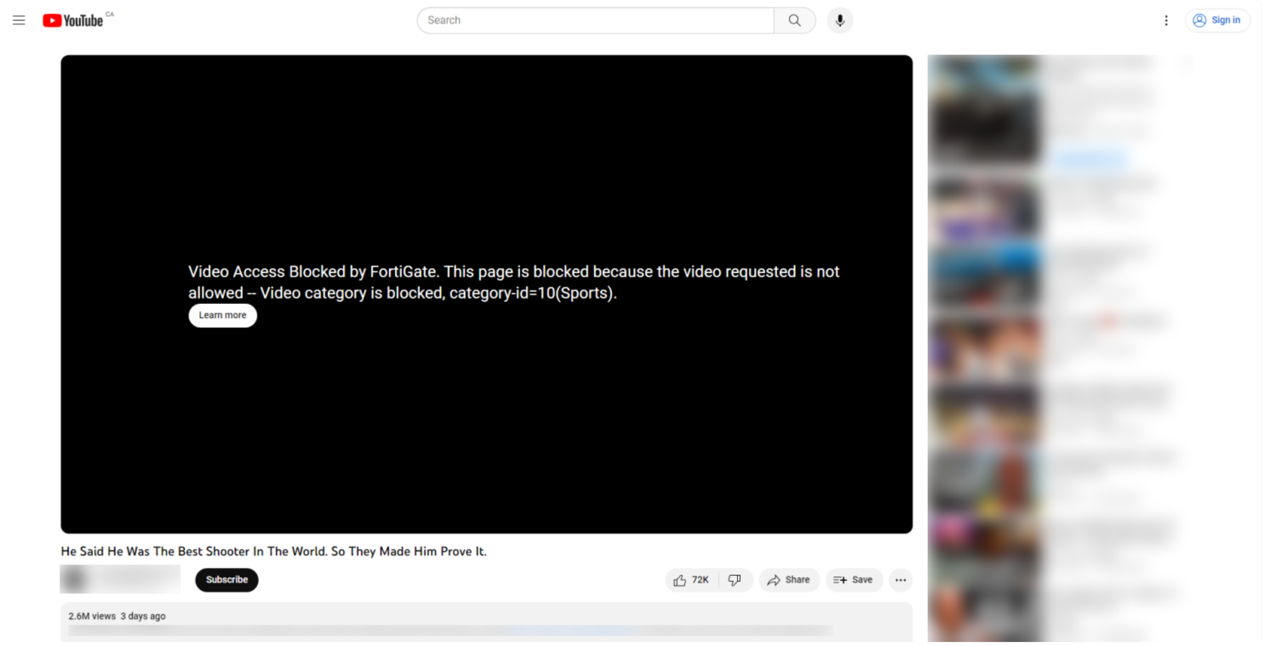
3. Apply the video filter in a firewall policy.

To verify the configuration:

1. From a client, search for a video in YouTube with a title containing the word "API" or "game". The video is allowed.



2. Search for another video without "API" or "game" in the title. The video is blocked.



Sample logs:

```
5: date=2023-11-24 time=17:37:45 eventtime=1700876265256758209 tz="-0800" logid="0350013713"
type="utm" subtype="webfilter" eventtype="unknown" level="notice" vd="vdom1" msg="Video
title is monitored." policyid=1 poluuid="090ca600-83e4-51ee-158a-a920fcf8f892"
sessionid=106912 srcip=10.1.100.141 dstip=142.250.217.110 srcport=25224 dstport=443
srcintf="port2" srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" proto=6
httpmethod="GET" service="HTTPS" action="passthrough" videoinfosource="API" profile="allow-
specific-title" videoid="EAyo3_zJj5c" videotitle="Youtube Data API V3 Video Search Example"
hostname="www.youtube.com" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KH" url="https://www.youtube.com/watch?v=EAyo3_zJj5c"
```

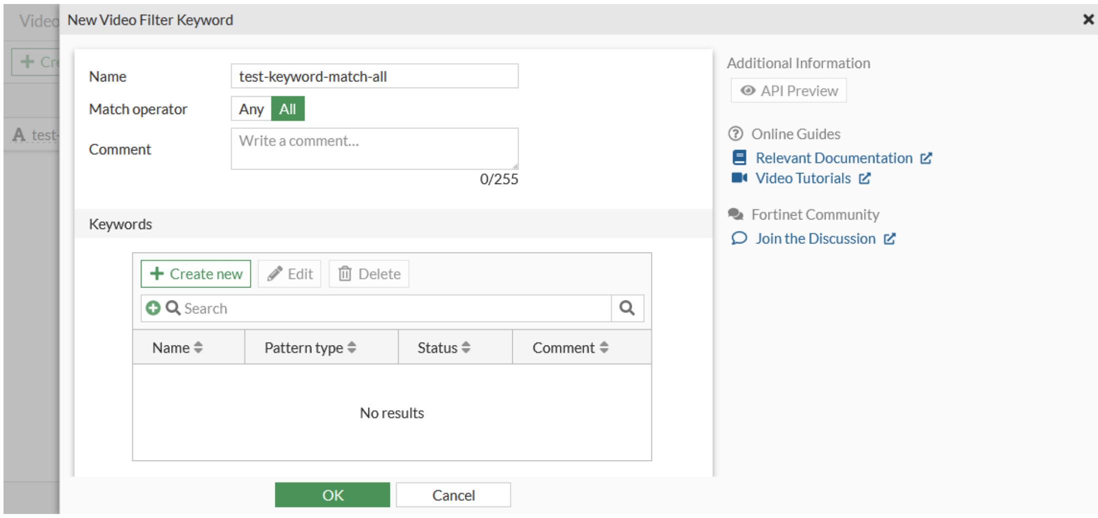
```
32: date=2023-11-24 time=17:38:58 eventtime=1700876338000614028 tz="-0800"
logid="0347013664" type="utm" subtype="webfilter" eventtype="videofilter-category"
level="warning" vd="vdom1" msg="Video category is blocked." policyid=1 poluuid="090ca600-
83e4-51ee-158a-a920fcf8f892" sessionid=107051 srcip=10.1.100.141 dstip=142.250.217.110
srcport=25260 dstport=443 srcintf="port2" srcintfrole="undefined" dstintf="port1"
dstintfrole="undefined" proto=6 httpmethod="POST" service="HTTPS" action="blocked"
videoinfosource="API" profile="allow-specific-title" videoid="7JhBGWSO108"
videocategoryid=10 videocategoryname="Sports" hostname="www.youtube.com" agent="Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KH" referralurl="https://www.youtube.com/"
url="https://www.youtube.com/youtubei/v1/player?key=AIzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_
11qcW8&prettyPrint=false"
```

Example 4: allowing a specific video by filtering the description while blocking others

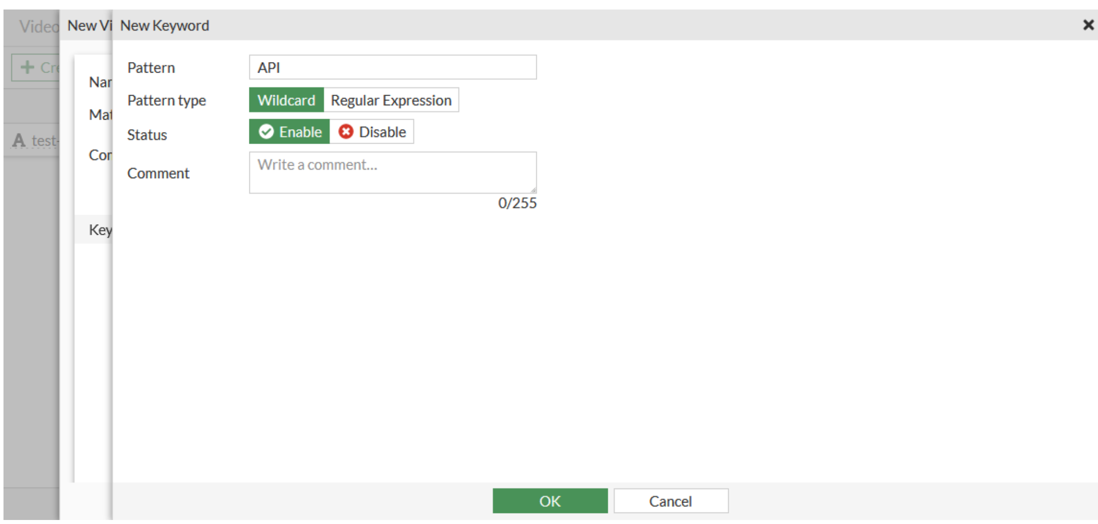
In this example, specific videos are allowed using description filtering while blocking others. The video filter profile contains two filters. The first filter uses a keyword list (monitor and allow the keywords 'API' and 'testing' with the *All* match operator). The second filter uses a channel filter to block any unmatched videos.

To configure the video filter profile in the GUI:

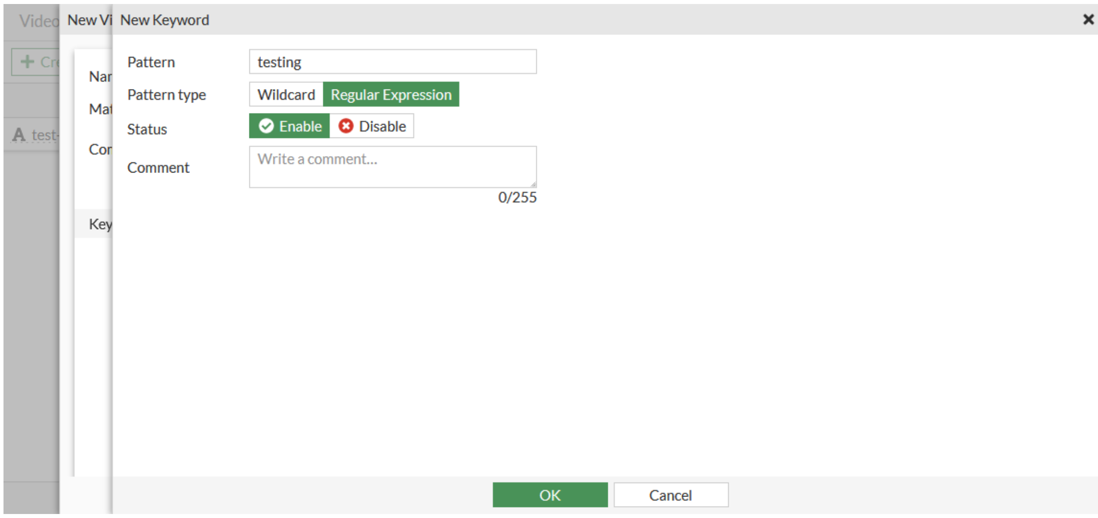
1. Configure the video filter keyword list:
   a. Go to *Security Profiles > Video Filter*, select the *Video Filter Keyword* tab, and click *Create new*.
   b. Enter a name (*test-keyword-match-all*) and set *Match operator* to *All*.



   c. In the *Keywords* table, click *Create new*.
   d. Configure the API keyword with the following settings:
      i. In the *Pattern* field, enter *API*.
      ii. Set the *Pattern type* to *Wildcard*.



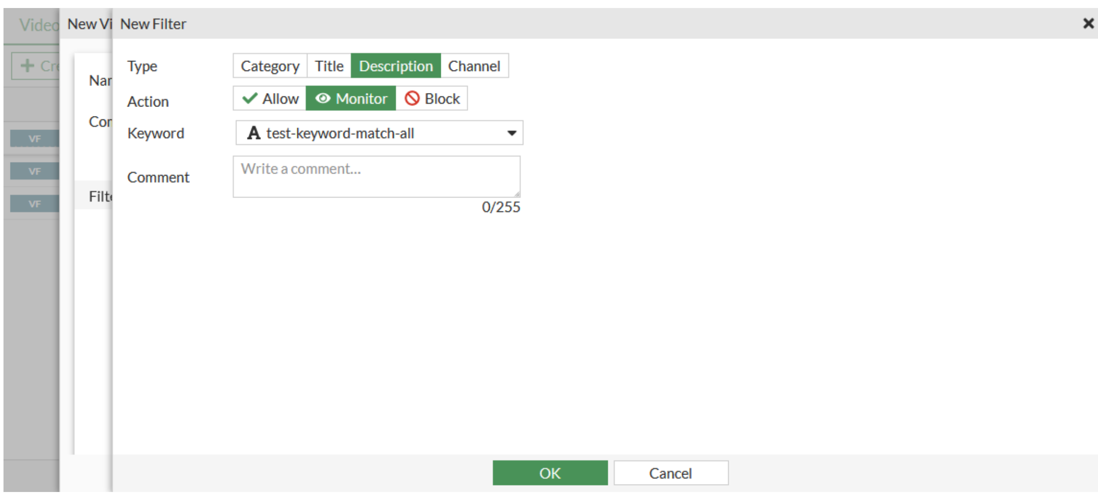
      iii. Click *OK*.
   e. Click *Create new*.
   f. Configure the testing keyword with the following settings:
      i. In the *Pattern* field, enter *testing*.
      ii. Set the *Pattern type* to *Regular Expression*.



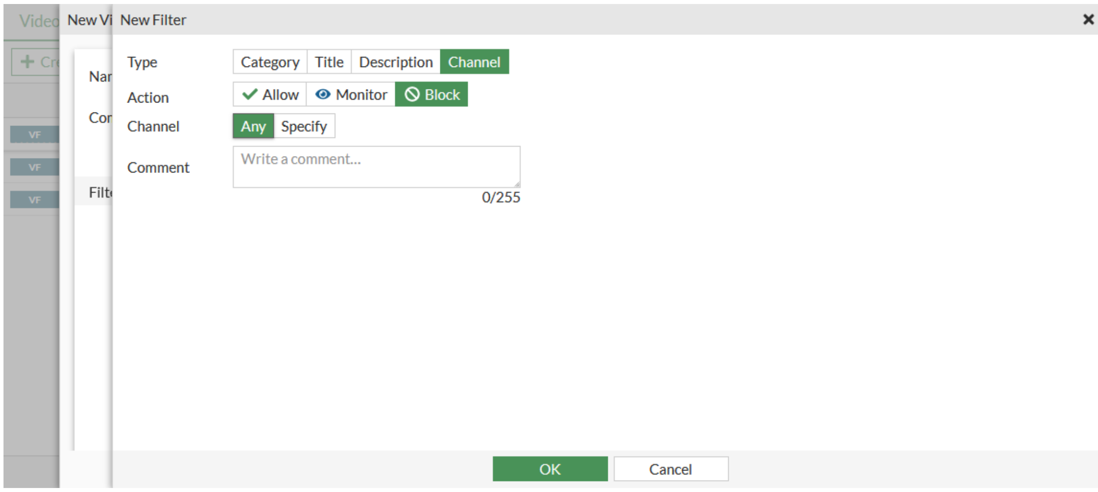
iii. Click *OK*.

g. Click *OK* to save the keyword list.

2. Configure the video filter profile:
   a. Go to *Security Profiles* > *Video Filter*, select the *Video Filter Profile* tab, and click *Create new*.
   b. Enter a name (*test-allow-specific-description*).
   c. In the *Filters* table, click *Create new*.
   d. Configure the first filter with the following settings:
      i. Set the *Type* to *Description*.
      ii. Set the *Action* to *Monitor*.
      iii. Set the *Keyword* to *test-keyword-match-all*.



      iv. Click *OK*.
   e. Configure the second filter with the following settings:
      i. Set the *Type* to *Channel*.
      ii. Set the *Action* to *Block*.
      iii. Set the *Channel* to *Any*.



   - iv. Click *OK*.
  - f. Click *OK* to save the video filter profile.
3. Apply the video filter in a firewall policy.

To configure the video filter profile in the CLI:

1. Configure the video filter keyword list:

```
config videofilter keyword
    edit 1
        set name "test-keyword-match-all"
        set match and
        config word
            edit "API"
                set pattern-type wildcard
                set status enable
            next
            edit "testing"
                set pattern-type regex
                set status enable
            next
        end
    next
end
```

2. Configure the video filter profile:

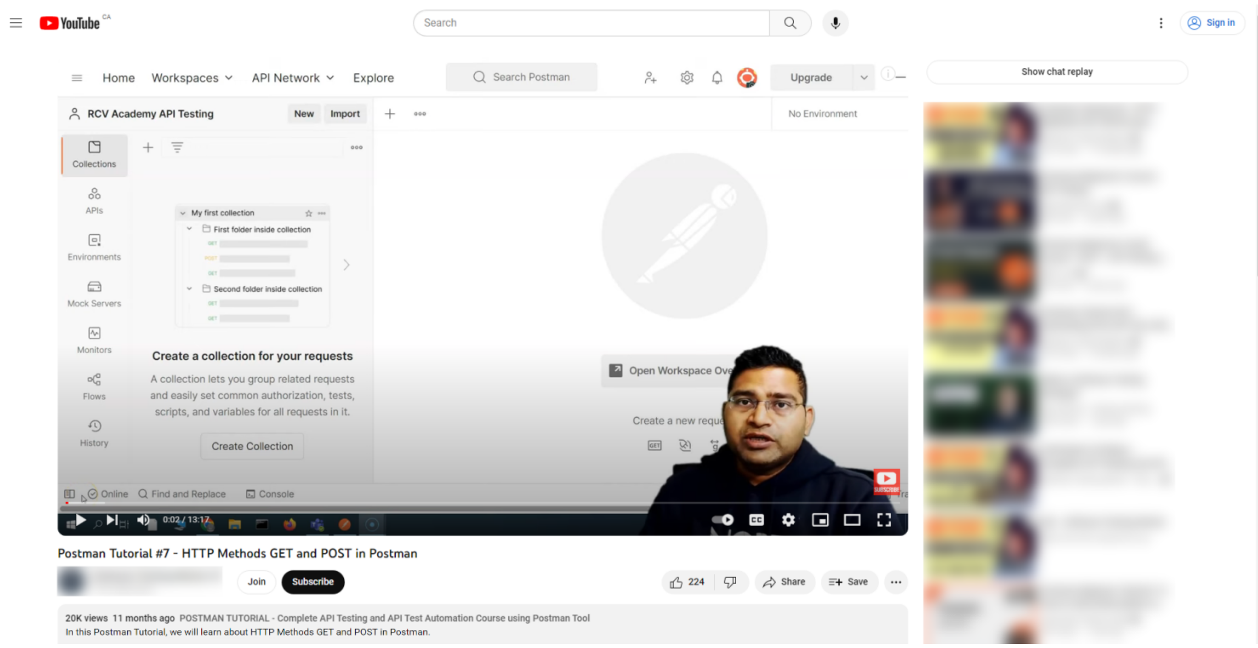
```
config videofilter profile
    edit "test-allow-specific-description"
        config filters
            edit 1
                set type description
                set keyword 1
                set action monitor
                set log enable
            next
            edit 2
                set type channel
                set channel "any"
```

```
                    set action block
                    set log enable
                next
            end
        next
    end
```

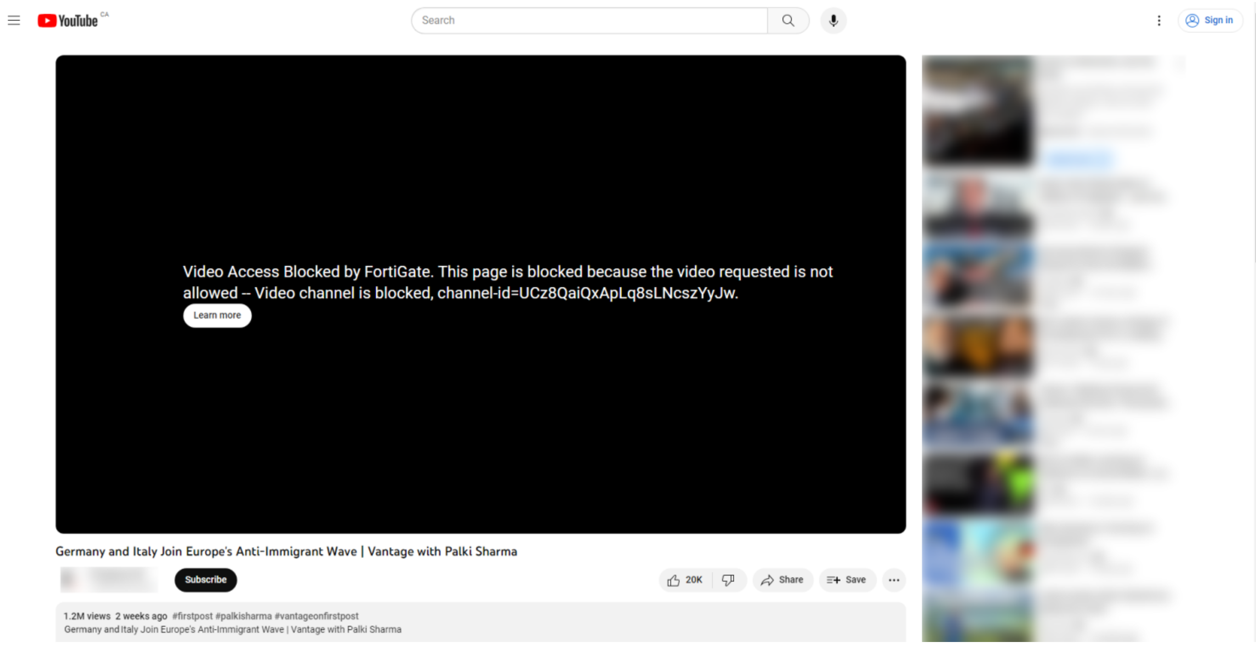
3. Apply the video filter in a firewall policy.

To verify the configuration:

1. From a client, search for a video in YouTube with a description containing the words "API" and "testing". The video is allowed since the video description contains "In this Postman Tutorial, we will learn about HTTP Methods GET and POST in Postman."



2. Search for another video without "API" and "testing" in the description. The video is blocked.



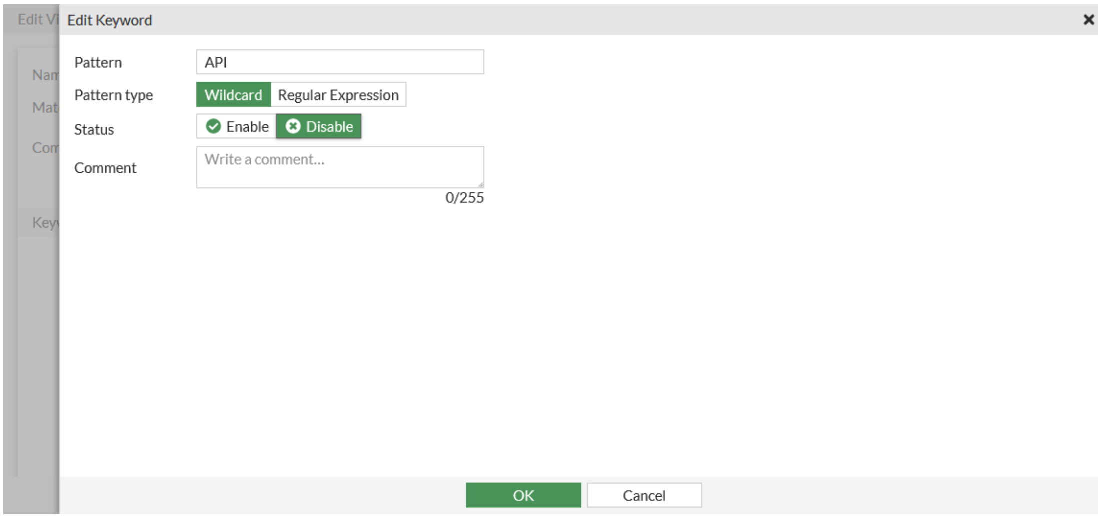
Sample logs:

```
1: date=2023-11-24 time=18:07:46 eventtime=1700878066675991798 tz="-0800" logid="0351013729"
type="utm" subtype="webfilter" eventtype="unknown" level="notice" vd="vdom1" msg="Video
description is monitored." policyid=1 poluuid="090ca600-83e4-51ee-158a-a920fcf8f892"
sessionid=109384 srcip=10.1.100.141 dstip=142.250.217.110 srcport=25452 dstport=443
srcintf="port2" srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" proto=6
httpmethod="GET" service="HTTPS" action="passthrough" videoinfosource="API" profile="test-
allow-specific-description" videoid="pUGmhtqVJRk" videodesc="Get all my courses for USD
5.99/Month - https://bit.ly/all-c..." hostname="www.youtube.com" agent="Mozilla/5.0 (Windows
NT 10.0; Win64; x64) AppleWebKit/537.36 (KH"
url="https://www.youtube.com/watch?v=pUGmhtqVJRk"

32: date=2023-11-24 time=18:08:54 eventtime=1700878134354460846 tz="-0800"
logid="0348013680" type="utm" subtype="webfilter" eventtype="videofilter-channel"
level="warning" vd="vdom1" msg="Video channel is blocked." policyid=1 poluuid="090ca600-
83e4-51ee-158a-a920fcf8f892" sessionid=109532 srcip=10.1.100.141 dstip=142.250.217.110
srcport=25498 dstport=443 srcintf="port2" srcintfrole="undefined" dstintf="port1"
dstintfrole="undefined" proto=6 httpmethod="POST" service="HTTPS" action="blocked"
videoinfosource="Cache" profile="test-allow-specific-description" videoid="uB0AcaxR-eM"
videochannelid="UCz8QaiQxApLq8sLNcszYyJw" hostname="www.youtube.com" agent="Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KH"
referralurl="https://www.youtube.com/watch?v=uB0AcaxR-eM"
url="https://www.youtube.com/youtubei/v1/player?key=AIzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_
11qcW8&prettyPrint=false"
```

Example 5: disabling a keyword in the keyword list

To disable a keyword in the keyword list in the GUI:

1. Go to *Security Profiles > Video Filter*, select the *Video Filter Keyword* tab, and edit an entry.
2. In the *Keywords* table, select an entry and click *Edit*.
3. Set the *Status* to *Disable*.



4. Click *OK*.
5. Click *OK* to save the keyword list.

To disable a keyword in the keyword list in the CLI:

```
config videofilter keyword
    edit 1
        set name "test-keyword-match-all"
        set match and
        config word
            edit "API"
                set status disable
            next
        end
    next
end
```

VPN

This section includes information about VPN related new features:

IPsec and SSL VPN

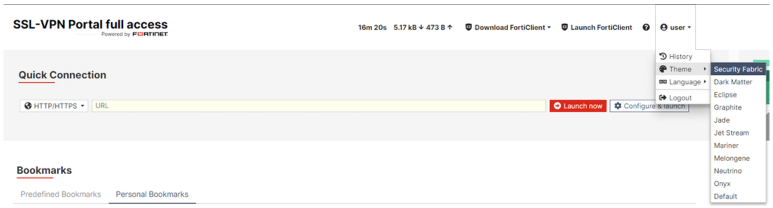
This section includes information about IPsec and SSL VPN related new features:

Update the SSL VPN web portal layout using Neutrino

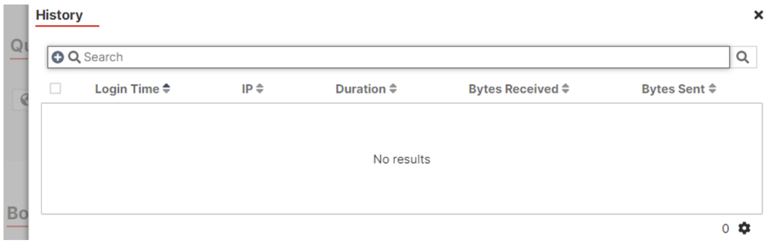
Using Neutrino styling, the SSL VPN web portal layout has been updated. Users logging into the portal can find the following changes to the layout:

- The top navigation bar is updated. Users can now download and launch FortiClient from the navigation bar.

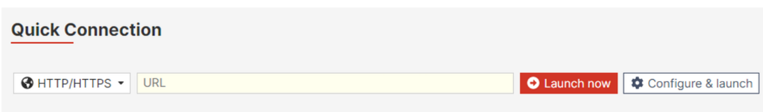
- *History*, *Theme*, and *Language* can be accessed from the user menu.



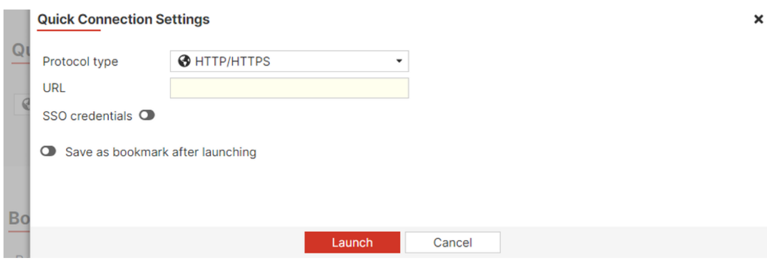
The *History* pane has been updated.



- *Quick Connection* has been updated and is displayed prominently at the top with the *HTTP/HTTPS* type set as the default.

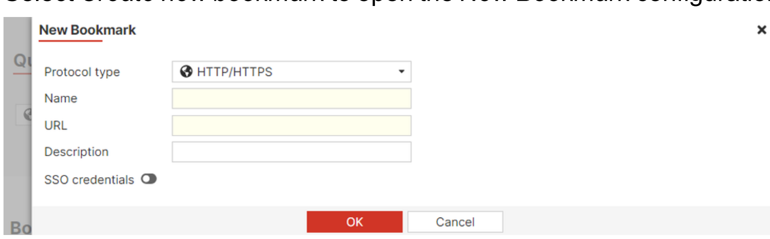


Users can configure advanced options and save the connection as a bookmark after launch from the *Quick Connection Settings* page.

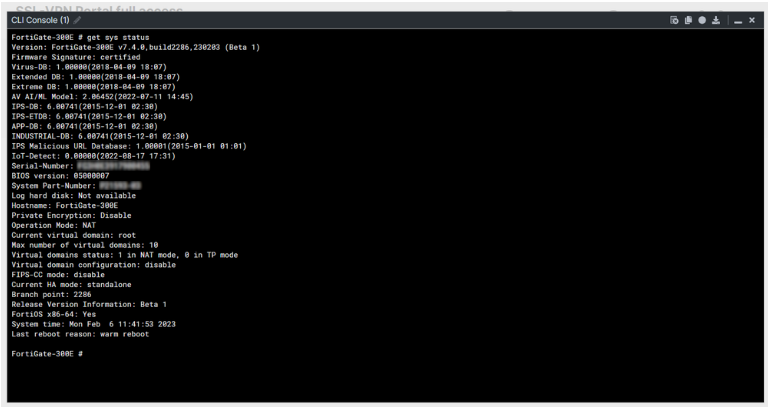


- *Bookmarks* are split into *Predefined Bookmarks* and *Personal Bookmarks* tabs. Users can search through their bookmarks using the *Search* field.

Select *Create new bookmark* to open the *New Bookmark* configuration pane.



- A CLI console is available for SSH and Telnet sessions.

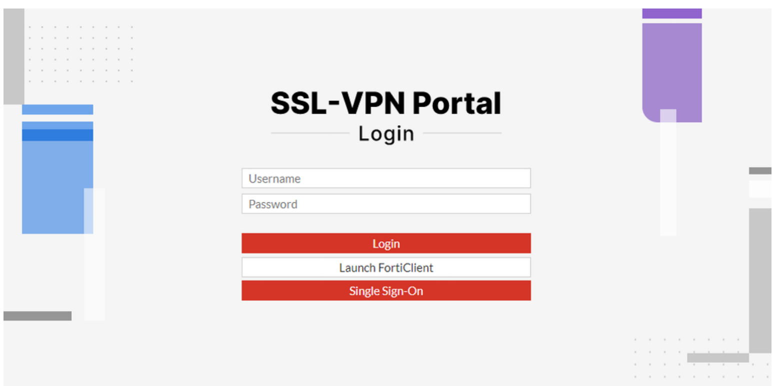


Improve the styling of the SSL VPN landing page

The styling of the SSL VPN web login page and portal have been updated with Fortinet Inc. corporate styling. Fortinet Inc. branding elements are incorporated into each theme.

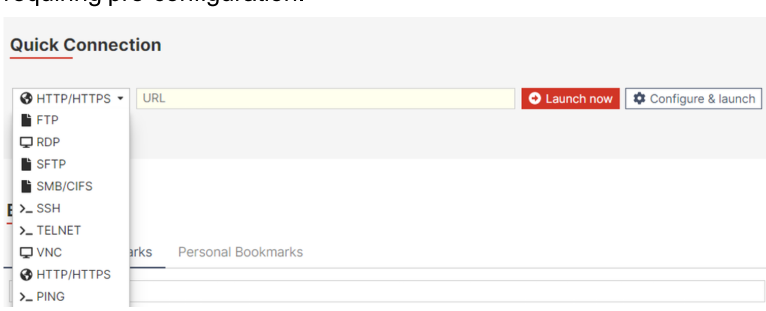
The styling updates include the following changes:

- An updated login page.



- The header displays the title of the portal along with a new, static *Powered by Fortinet* subheader.

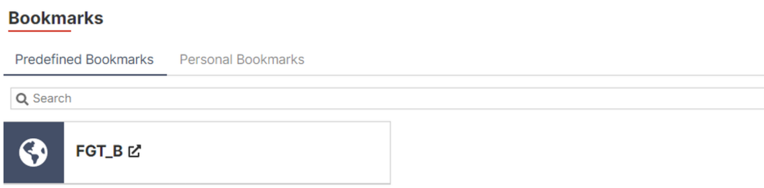
- *Quick Connection* now supports toggling between different protocols. Launching from *Quick Connection* also provides quick access to RDP and VNC directly and prompts users for their username and password without requiring pre-configuration.



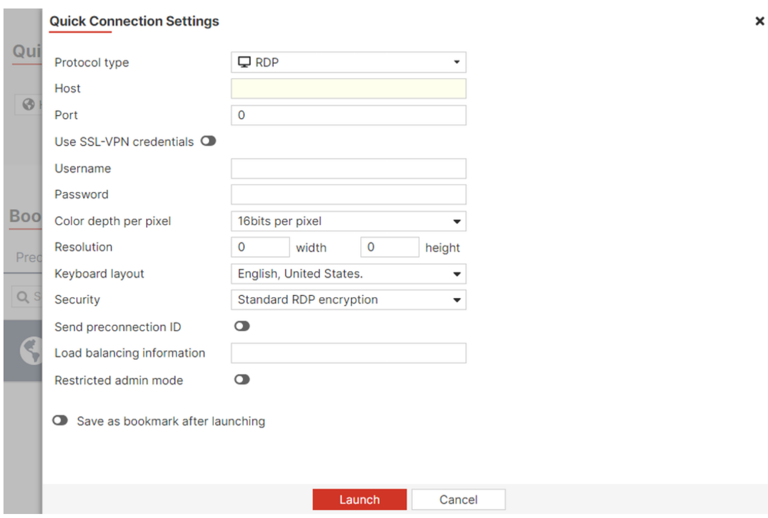
- *Bookmarks* now display at most three bookmark entries per row.

- Some elements and entries have been renamed:
  - In *Quick Connection*, *Configure & launch* and *Launch now* were previously *More Options* and *Launch*, respectively.

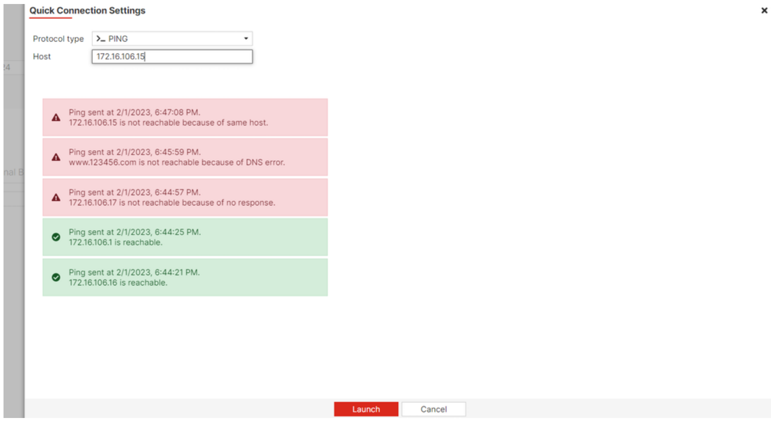
  - *Predefined Bookmarks* was previously *Shared Bookmarks*.



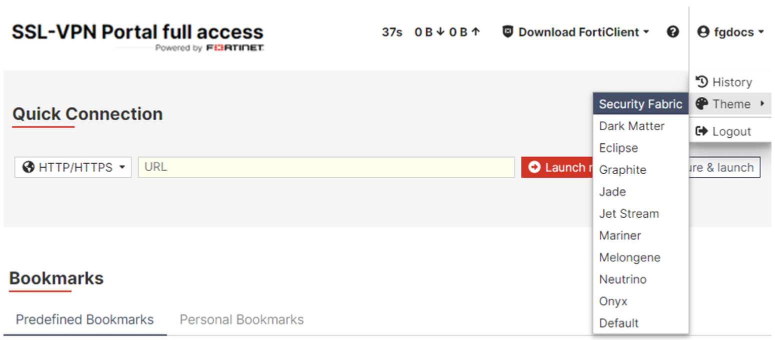
  - In the *Quick Connection Settings* pane, *Protocol type* was previously *Type*, and *Resolution*, *width*, and *height* were previously *Screen width* and *Screen height*.



- The five latest ping results are displayed as notifications.



- The *Security Fabric* and *Jet Stream* themes have been added with *Security Fabric* set as the default theme.



Allow SSL VPN login to be redirected to a custom landing page

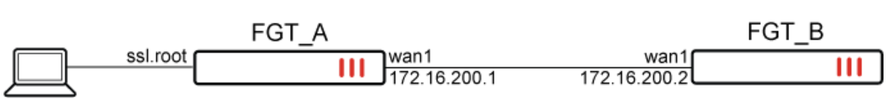
This information is also available in the FortiOS 7.4 Administration Guide:
- SSL VPN custom landing page

This enhancement allows a SSL VPN web mode user to log in to the web portal and be redirected to a custom landing page. The new landing page can accept SSO credentials as well as SSO from form data. This allows administrators to streamline web application access for their users. The custom redirected portal has a logout button so that when users log out from the web application, they are also logged out from the SSL VPN web connection.

The custom landing page can be configured in *VPN > SSL-VPN Portals* by setting the portal *Landing page* to *Custom* or by using the command `config landing-page`.

Example

In the following example, the SSL VPN web portal settings are configured so that the URL of the custom landing page of FGT_A is set to the FGT_B login page. Therefore, when a web user is logging into FGT_A's SSL VPN web portal, they will automatically be redirected to FGT_B, where the SSO username and password are passed into the username and password input fields. This allows for single sign on of the connecting user into FGT_B through the SSL VPN.



To configure a custom landing page from the CLI:

1. Configure the user and user group:

```
config user local
    edit "custom_landing_user"
        set type password
        set passwd ********
    next
end
config user group
    edit "ssl-web-group"
        set member "custom_landing_user"
    next
end
```

2. Configure the SSL VPN web portal:

```
config vpn ssl web portal
    edit "custom_landing"
        set web-mode enable
        set landing-page-mode enable
        config landing-page
            set url "https://172.16.200.2/login"
            set sso static
            config form-data
                edit "username"
                    set value "admin"
                next
                edit "secretkey"
                    set value "1"
                next
            end
            set sso-credential alternative
            set sso-username "admin"
            set sso-password ********
        end
    next
end
```

3. Configure the SSL VPN settings:

```
config vpn ssl settings
    set servercert "fgt_gui_automation"
    set tunnel-ip-pools "SSLVPN_TUNNEL_ADDR1"
    set tunnel-ipv6-pools "SSLVPN_TUNNEL_IPv6_ADDR1"
    set port 1443
    set source-interface "port1"
    set source-address "all"
    set source-address6 "all"
    set default-portal "full-access"
    config authentication-rule
        edit 2
            set users "custom_landing_user"
            set portal "custom_landing"
        next
    end
```

```
    set encrypt-and-store-password enable
end
```

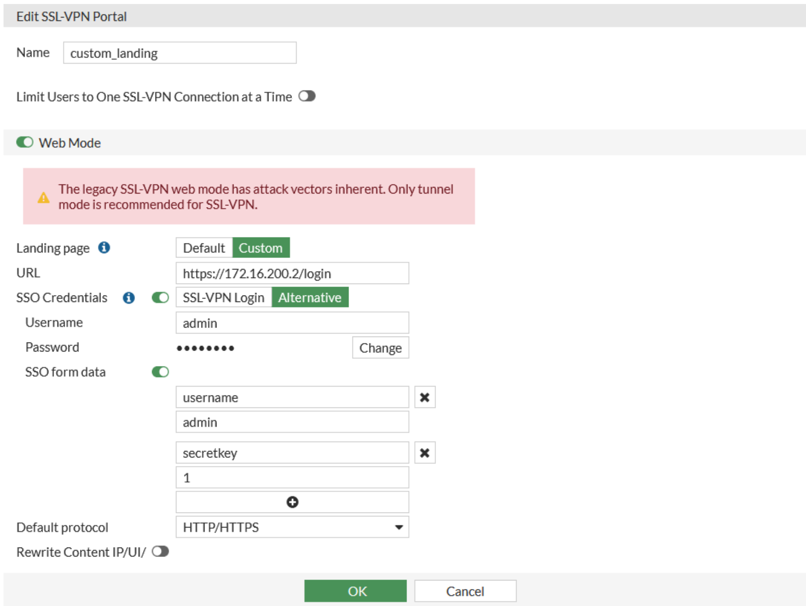
4. Configure the firewall policy:

```
config firewall policy
    edit 1
        set name "testpolicy"
        set srcintf "ssl.root"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
        set groups "ssl-web-group"
        set users "custom_landing_user"
    next
end
```

To configure a custom landing page from the GUI:

1. Configure the user and user group:
   a. Go to *User & Authentication* > *User Definition* to create the `custom_landing_user` user.
   b. Go to *User & Authentication* > *User Groups* to create the `ssl-web-group` user group with the member `custom_landing_user`.
2. Configure the SSL VPN web portal:
   a. Go to *VPN* > *SSL-VPN Portals*.
   b. Click *Create New*.
   c. Enter `custom_landing` as the *Name*.
   d. Enable custom *Web Mode* features:
      i. Enable *Web Mode*.
      ii. Set *Landing Page* to *Custom*.
      iii. Enter the FGT_B login page *URL*.
      iv. Enable *SSO Credentials* and select *Alternative*.

v. Enable *SSO form data* and enter the form keys and values.



e. Click *OK*.

3. Configure the SSL VPN settings:
   a. Go to *VPN > SSL-VPN Settings*.
   b. Set *Listen on Interface(s)* to *port1*.
   c. Set *Listen on Port* to *1443*.
   d. Set *Server Certificate* to *fgt_gui_automation*.
   e. Create a new *Authentication/Portal Mapping* for group *ssl-web-group* mapping the portal *custom-landing*.
   f. Click *Apply*.
4. Configure the firewall policy:
   a. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
   b. Configure the following settings:

Name	testpolicy
Incoming Interface	ssl.root
Outgoing Interface	wan1
Source	all custom_landing_user ssl-web-group
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

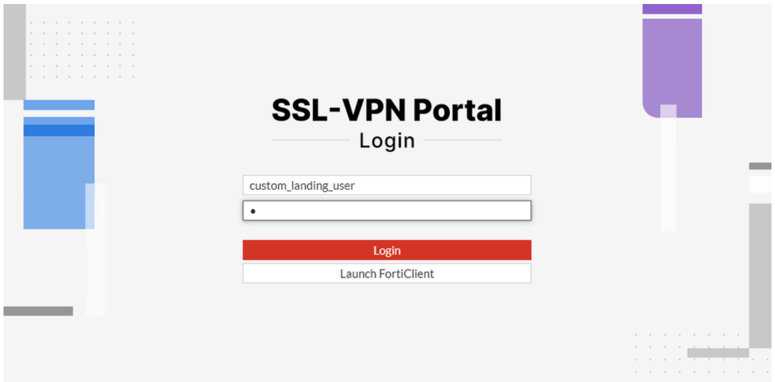
   c. Enable *NAT*.

d. Enable *Log Allowed Traffic* and set it to *All Sessions*.
e. Click *OK*.

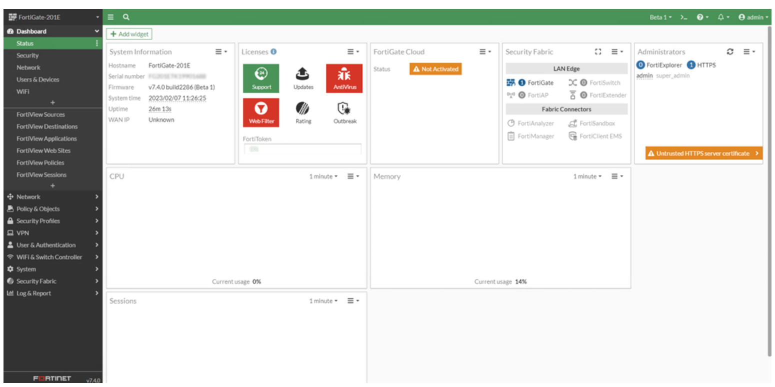
Once the SSL VPN web portal is configured, the connected user can access FGT_B through the FGT_A SSL VPN web portal.

To access FGT_B through the FGT_A SSL VPN web portal:

1. Enter your SSO credentials in the SSL VPN login fields.



The landing page is redirected to the FGT_B GUI automatically.



IPsec SA key retrieval from a KMS server using KMIP

This information is also available in the FortiOS 7.4 Administration Guide:

- IPsec SA key retrieval from a KMS server using KMIP

In environments that require centralized management of cryptographic keys where no key derivations or algorithmic operations are allowed on edge devices (such as the FortiGate), they will deploy a Key Management Services (KMS) server cluster to generate and manage all cryptographic keys. Then, the Key Management Interoperability Protocol (KMIP) is used on the edge devices to locate the KMS server, create keys if they do not exist, and retrieve keys to be used for securing these edge devices.

FortiGates have a KMIP client that sends KMIP requests to locate the Key Management Services (KMS) server, creates keys if they do not exist on the KMS server, and retrieves keys from the KMS server to use as IPsec security association (SA) keys for IKEv2 only.

This feature allows the FortiGate to offload the task of generating IPsec SA keys to a KMS server, regardless of specific IPsec VPN topologies with a FortiGate, when the administrator has the requirement to centralize cryptographic keys management in a KMS server.

The FortiGate's integrated KMIP client also supports the following:

- If the KMS server is unavailable, then the FortiGate continues to use the previous keys to avoid a network blackout.
- ADVPN configurations for the hub and spoke, so that shortcuts between two spokes will use their own encryption keys retrieved from the KMS server.
- Multiple tunnels between the same tunnel endpoints using multiple VRFs.

To configure the KMIP server:

```
config vpn kmip-server
    edit <KMS_server_ID>
        config server-list
            edit <ID>
                set server <server_IP>
                set cert <string>
            next
        end
        set username <username_defined_on_KMS_server>
        set password <password>
    next
end
```

To apply the KMS server in the phase 1 interface settings:

```
config vpn ipsec phase1-interface
    edit <name>
        set kms <KMS_server_ID>
    next
end
```

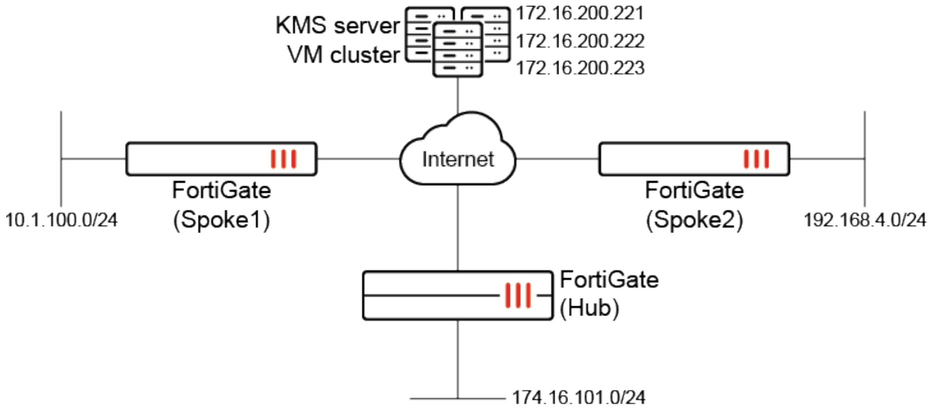
IPsec tunnels will not be established if a FortiGate VPN peer does not support KMS, or has not configured `kms <KMS_server_ID>` in `config vpn ipsec phase1-interface`.

The following diagnostic commands have been added:

- `get vpn ike kms-keys`
- `diagnose debug application kmipd -1`
- `execute kmip {create | destroy | get | locate | rekey} <parameter>`

Example

In this example, there is a topology with an ADVPN hub FortiGate and two spoke FortiGates. There is a cluster or three KMS server VMs (172.16.200.221, 172.16.200.222, and 172.16.200.223) that operates in round-robin mode. The `testuser1_Cert` certificate is issued by the KMS server, and the `testuser1` user is defined on the KMS server. Authentication to the KMS server by the KMIP client requires both a certificate and a password.



The Hub FortiGate acting as the responder will try to locate keys on the KMS server first. If they do not exist, the FortiGate requests to create new keys on KMS server. The responder sends the keys' names to the Spoke1 and Spoke2 FortiGates acting as the initiators using IKE messages, and these initiators locate and retrieve keys from KMS server using the keys' names. The `keylifeseconds` parameter in phase 2 defines how often the FortiGate will try to synchronize local keys to those on the KMS server.

The keys are retrieved from the KMS server and used as IPsec SA keys in IPsec tunnels. The key format used is: [IDi/r]-[IDr/i]-[phase2name]-ENC/AUTH-[keyalg]-[keylen].

First, this example focuses on the Hub FortiGate and the IPsec VPN connection between the Spoke1 and Hub FortiGate. Second, this example focuses on the spoke-to-spoke tunnel, also known as a shortcut tunnel or shortcut, which is established when traffic flows between the Spoke1 and Spoke2 FortiGates.

To configure IPsec SA key retrieval from a KMS server on the Hub FortiGate:

1. Configure the KMIP server:

```
config vpn kmip-server
    edit "KMS_server"
        config server-list
            edit 1
                set server "172.16.200.221"
                set cert "testuser1_Cert"
            next
            edit 2
                set server "172.16.200.222"
                set cert "testuser1_Cert"
            next
            edit 3
                set server "172.16.200.223"
                set cert "testuser1_Cert"
            next
        end
        set username "testuser1"
        set password **********
    next
end
```

2. Configure the IPsec VPN phase 1 settings:

```
config vpn ipsec phase1-interface
    edit "hub"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set authmethod signature
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dpd on-idle
        set auto-discovery-sender enable
        set kms "KMS_server"
        set certificate "Fortinet_Factory_Backup"
        set dpd-retryinterval 60
    next
end
```

This feature is only supported in IKEv2. The `localid` is required in the phase 1 settings when using the PSK authentication method.

3. Configure the IPsec VPN phase 2 settings:

```
config vpn ipsec phase2-interface
    edit "hub"
        set phase1name "hub"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set keylifeseconds 7200
    next
end
```

To verify the IPsec configuration and tunnel between the Hub and Spoke1 FortiGates:

1. Verify the tunnel state on the Hub:

```
Hub # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=hub ver=2 serial=1 172.16.200.4:0->0.0.0.0:0 tun_id=10.0.0.1 tun_id6=::10.0.0.1
dst_mtu=0 dpd-link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-
rfc  role=primary accept_traffic=1 overlay_id=0

proxyid_num=0 child_num=2 refcnt=4 ilast=42965007 olast=42965007 ad=/0
stat: rxp=980 txp=1980 rxb=125003 txb=123108
dpd: mode=on-idle on=0 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
run_tally=0
------------------------------------------------------
name=hub_0 ver=2 serial=10 172.16.200.4:0->172.16.200.1:0 tun_id=10.10.10.2 tun_
id6=::10.0.0.16 dst_mtu=1500 dpd-link=on weight=1
```

```
bound_if=10 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74408 options[122a8]=npu
rgwy-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

parent=hub index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=6 olast=6 ad=s/1
stat: rxp=21 txp=39 rxb=2644 txb=2389
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=hub proto=0 sa=1 ref=3 serial=1 ads
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=826 type=00 soft=0 mtu=1438 expire=6673/0B replaywin=2048
       seqno=15 esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=7191/7200
  dec: spi=628d1814 esp=aes key=16 5dad0d8d3568eab7c3f259349dc64039
       ah=sha1 key=20 e660f491b80b2cfdcdb0d737942bea2e853dac8d
  enc: spi=471dfe2e esp=aes key=16 1de4b8e8accaa792e0934fbd9f933a6a
       ah=sha1 key=20 1fa244d3971b4d4df59b8d7b3655a1b77f8e65af
  dec:pkts/bytes=22/2696, enc:pkts/bytes=59/4949
  npu_flag=03 npu_rgwy=172.16.200.1 npu_lgwy=172.16.200.4 npu_selid=e dec_npuid=1 enc_
npuid=0
------------------------------------------------------
name=hub_1 ver=2 serial=f 172.16.200.4:0->172.16.200.3:0 tun_id=10.10.10.3 tun_
id6=::10.0.0.15 dst_mtu=1500 dpd-link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74408 options[122a8]=npu
rgwy-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

parent=hub index=1
proxyid_num=1 child_num=0 refcnt=5 ilast=2 olast=2 ad=s/1
stat: rxp=21 txp=43 rxb=2615 txb=2718
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=hub proto=0 sa=1 ref=3 serial=1 ads
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=826 type=00 soft=0 mtu=1438 expire=6665/0B replaywin=2048
       seqno=17 esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=7187/7200
  dec: spi=628d1813 esp=aes key=16 5fcca9194ced21b0a586a8fd7a27cbf7
       ah=sha1 key=20 6d6d9dc77d5af89f062927c4d4695d404df1ffe3
  enc: spi=8d568113 esp=aes key=16 2006f323b760238048fcd6f7783b0a04
       ah=sha1 key=20 bd6db68ee035088f35174b2b5c58a51fbbe3f5b5
  dec:pkts/bytes=22/2686, enc:pkts/bytes=65/5566
  npu_flag=03 npu_rgwy=172.16.200.3 npu_lgwy=172.16.200.4 npu_selid=d dec_npuid=1 enc_
npuid=0
```

2. Verify the KMS keys for the VPN tunnel between the Hub and Spoke1:

```
Hub # get vpn ike  kms-keys

vd: root/0
name: hub_1
addr: 172.16.200.4:500 -> 172.16.200.3:500

  phase2
```

```
    name: hub
    server: "KMS_server"
    spi: 628d1813
      enc
        keyname: "Spoke2-hub-hub-ENC-AES-16"
        key: 5fcca9194ced21b0a586a8fd7a27cbf7
      auth
        keyname: "Spoke2-hub-hub-AUTH-SHA1-20"
        key: 6d6d9dc77d5af89f062927c4d4695d404df1ffe3
    spi: 8d568113
      enc
        keyname: "hub-Spoke2-hub-ENC-AES-16"
        key: 2006f323b760238048fcd6f7783b0a04
      auth
        keyname: "hub-Spoke2-hub-AUTH-SHA1-20"
        key: bd6db68ee035088f35174b2b5c58a51fbbe3f5b5


  vd: root/0
  name: hub_0
  addr: 172.16.200.4:500 -> 172.16.200.1:500

    phase2
    name: hub
    server: "KMS_server"
    spi: 628d1814
      enc
        keyname: "Spoke1-hub-hub-ENC-AES-16"
        key: 5dad0d8d3568eab7c3f259349dc64039
      auth
        keyname: "Spoke1-hub-hub-AUTH-SHA1-20"
        key: e660f491b80b2cfdcdb0d737942bea2e853dac8d
    spi: 471dfe2e
      enc
        keyname: "hub-Spoke1-hub-ENC-AES-16"
        key: 1de4b8e8accaa792e0934fbd9f933a6a
      auth
        keyname: "hub-Spoke1-hub-AUTH-SHA1-20"
        key: 1fa244d3971b4d4df59b8d7b3655a1b77f8e65af
```

To verify the IPsec configuration and tunnel between the Spoke1 and Spoke2 FortiGates:

1. Verify the tunnel state on Spoke1:

```
Spoke1 # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=spoke1 ver=2 serial=1 172.16.200.1:0->172.16.200.4:0 tun_id=172.16.200.4 tun_
id6=::172.16.200.4 dst_mtu=1500 dpd-link=on weight=1
bound_if=19 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc
 run_state=0 role=primary accept_traffic=1 overlay_id=0 proxyid_num=1 child_num=1
refcnt=5 ilast=35 olast=35 ad=r/2
stat: rxp=1 txp=11 rxb=71 txb=699
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=5
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
```

```
proxyid=spoke1 proto=0 sa=1 ref=3 serial=2 adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=12026 type=00 soft=0 mtu=1438 expire=6621/0B replaywin=2048
       seqno=c esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=6903/7200
  dec: spi=471dfe2e esp=aes key=16 1de4b8e8accaa792e0934fbd9f933a6a
       ah=sha1 key=20 1fa244d3971b4d4df59b8d7b3655a1b77f8e65af
  enc: spi=628d1814 esp=aes key=16 5dad0d8d3568eab7c3f259349dc64039
       ah=sha1 key=20 e660f491b80b2cfdcdb0d737942bea2e853dac8d
  dec:pkts/bytes=2/142, enc:pkts/bytes=22/2131
  npu_flag=03 npu_rgwy=172.16.200.4 npu_lgwy=172.16.200.1 npu_selid=1 dec_npuid=2 enc_
npuid=2
run_tally=0
------------------------------------------------------
name=spoke1_0 ver=2 serial=4 172.16.200.1:0->172.16.200.3:0 tun_id=172.16.200.3 tun_
id6=::172.16.200.3 dst_mtu=1500 dpd-link=on weight=1
bound_if=19 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/66216 options[102a8]=npu
rgwy-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0 parent=spoke1
index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=10 olast=10 ad=r/2
stat: rxp=1 txp=5 rxb=84 txb=420
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=spoke1 proto=0 sa=1 ref=3 serial=1 adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=12026 type=00 soft=0 mtu=1438 expire=6947/0B replaywin=2048
       seqno=6 esn=0 replaywin_lastseq=00000402 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=7190/7200
  dec: spi=471dfe2f esp=aes key=16 a6d6a25cd986860bcc502d58f32e99de
       ah=sha1 key=20 07d712156eaca28439fbe944e3a8c9af4c45166a
  enc: spi=8d568114 esp=aes key=16 b01c534b11792b856c1b95c78c4cad91
       ah=sha1 key=20 fe6a82177db6911b3203d1306969e5ddec8fd039
  dec:pkts/bytes=2/168, enc:pkts/bytes=10/1180
  npu_flag=03 npu_rgwy=172.16.200.3 npu_lgwy=172.16.200.1 npu_selid=4 dec_npuid=2 enc_
npuid=2
```

2. Verify the KMS keys for the VPN tunnel between Spoke1 and Spoke2:

```
Spoke1 # get vpn ike kms-keys

vd: root/0
name: spoke1
addr: 172.16.200.1:500 -> 172.16.200.4:500

  phase2
  name: spoke1
  server: "KMS_server"
  spi: 628d1814
    enc
      keyname: "Spoke1-hub-hub-ENC-AES-16"
      key: 5dad0d8d3568eab7c3f259349dc64039
    auth
      keyname: "Spoke1-hub-hub-AUTH-SHA1-20"
      key: e660f491b80b2cfdcdb0d737942bea2e853dac8d
```

```
    spi: 471dfe2e
      enc
        keyname: "hub-Spoke1-hub-ENC-AES-16"
        key: 1de4b8e8accaa792e0934fbd9f933a6a
      auth
        keyname: "hub-Spoke1-hub-AUTH-SHA1-20"
        key: 1fa244d3971b4d4df59b8d7b3655a1b77f8e65af


  vd: root/0
  name: spoke1_0
  addr: 172.16.200.1:500 -> 172.16.200.3:500

    phase2
    name: spoke1
    server: "KMS_server"
    spi: 8d568114
      enc
        keyname: "Spoke1-Spoke2-spoke2-ENC-AES-16"
        key: b01c534b11792b856c1b95c78c4cad91
      auth
        keyname: "Spoke1-Spoke2-spoke2-AUTH-SHA1-20"
        key: fe6a82177db6911b3203d1306969e5ddec8fd039
    spi: 471dfe2f
      enc
        keyname: "Spoke2-Spoke1-spoke2-ENC-AES-16"
        key: a6d6a25cd986860bcc502d58f32e99de
      auth
        keyname: "Spoke2-Spoke1-spoke2-AUTH-SHA1-20"
        key: 07d712156eaca28439fbe944e3a8c9af4c45166a
```

3. Verify the FortiGate (KMIP client) connection to the KMS server:

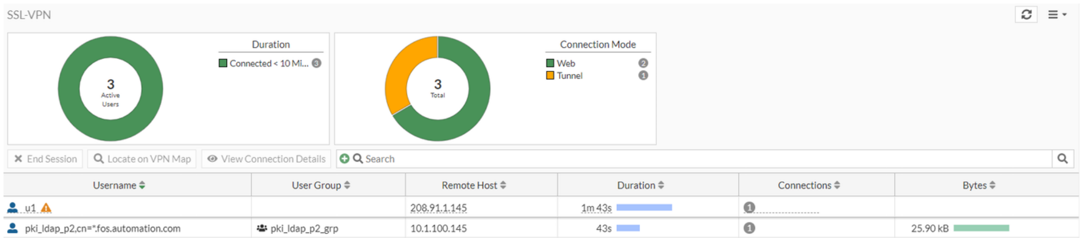
```
Spoke1 # execute kmip locate KMS_server hub-Spoke1-hub-AUTH-SHA1-20
Locating key 'hub-Spoke1-hub-AUTH-SHA1-20', jobid=1935521133
Ret=0, jobid=1935521133
        Key ID: 2ba130bff7174ba7a237d7ea53611121383b132cf18a4fd183890ca196296cb4
```

Add user group information to the SSL-VPN monitor

This information is also available in the FortiOS 7.4 Administration Guide:

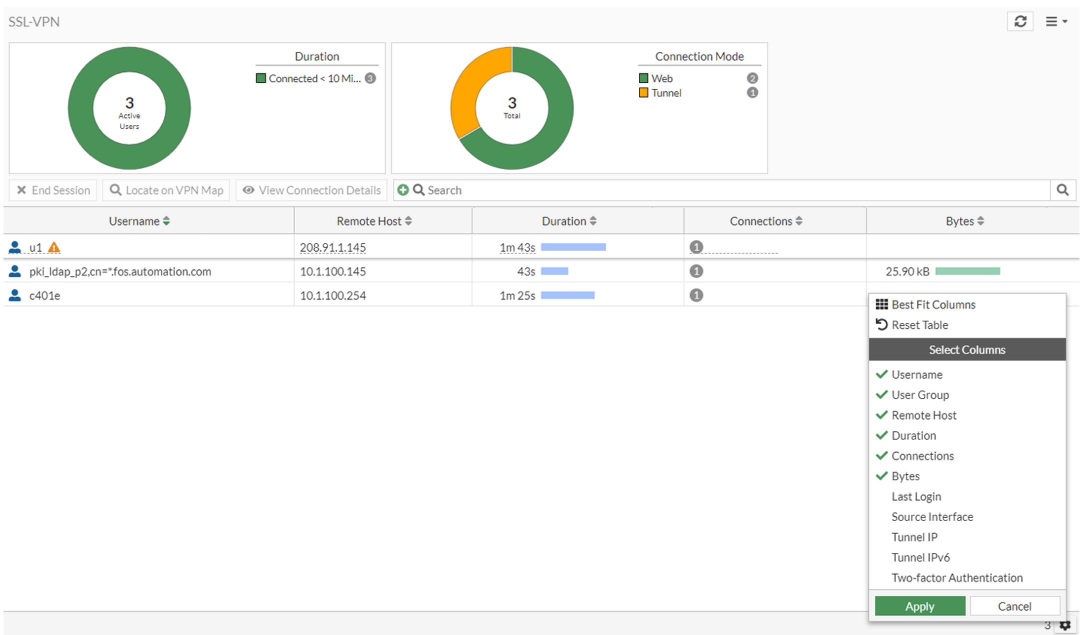
- SSL-VPN monitor

User group information can be viewed on the SSL-VPN monitor in the *User Group* column. Therefore, it is not necessary to navigate to *User & Authentication* > *User Groups* to locate the group information.

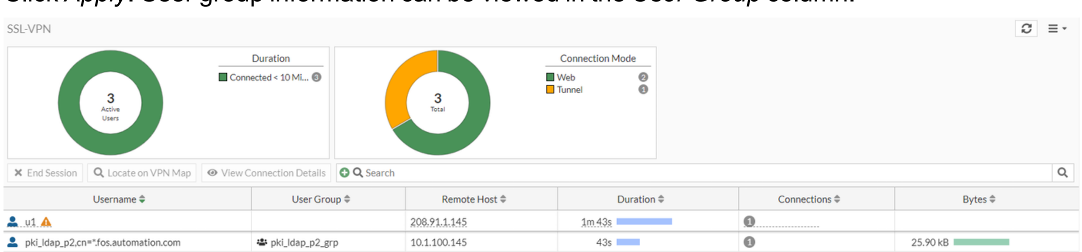


To view SSL-VPN user group information:

1. Go to *Dashboard > Network*.
2. Hover over the *SSL-VPN* widget and click *Click to expand*.
3. Click the gear icon. Available columns are listed.
4. Select *User Group*.



5. Click *Apply*. User group information can be viewed in the *User Group* column.



IPsec IKE load balancing based on FortiSASE account information

This information is also available in the FortiOS 7.4 Administration Guide:

- IPsec IKE load balancing based on FortiSASE account information

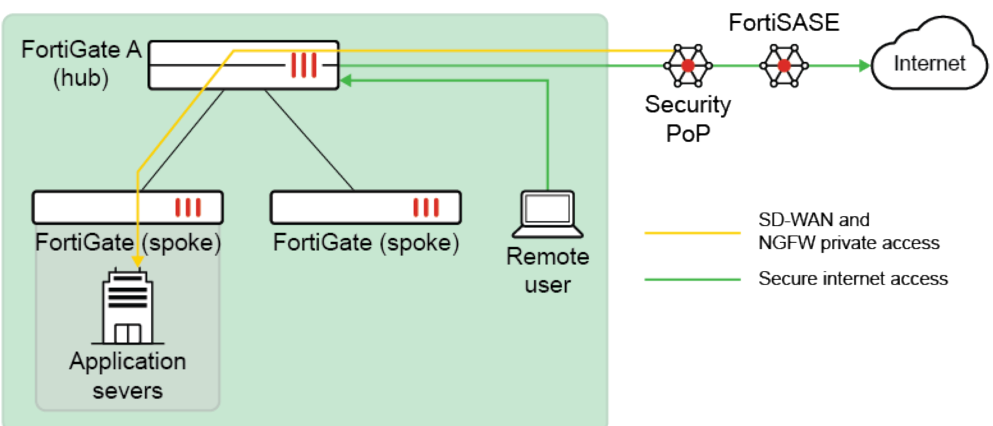
The FortiGate device ID is carried by the IKEv2 message NOTIFY payload when it is configured.

```
config vpn ipsec phase1-interface
    edit <name>
        set dev-id-notification enable
        set dev-id <string>
    next
end
```

This device ID configuration is required when the FortiGate is configured as a secure edge LAN extension for FortiSASE. It allows FortiSASE to distribute IKE/IPsec traffic according to the FortiGate device ID to achieve load balancing.

Example

In this example, a FortiGate SD-WAN is configured, which acts as a secure edge. FortiSASE ensures secure internet access for users in the local network behind the FortiGate and allows other FortiSASE remote users with secure private access to private resources behind the FortiGate.



To configure FortiGate A (FGT-A):

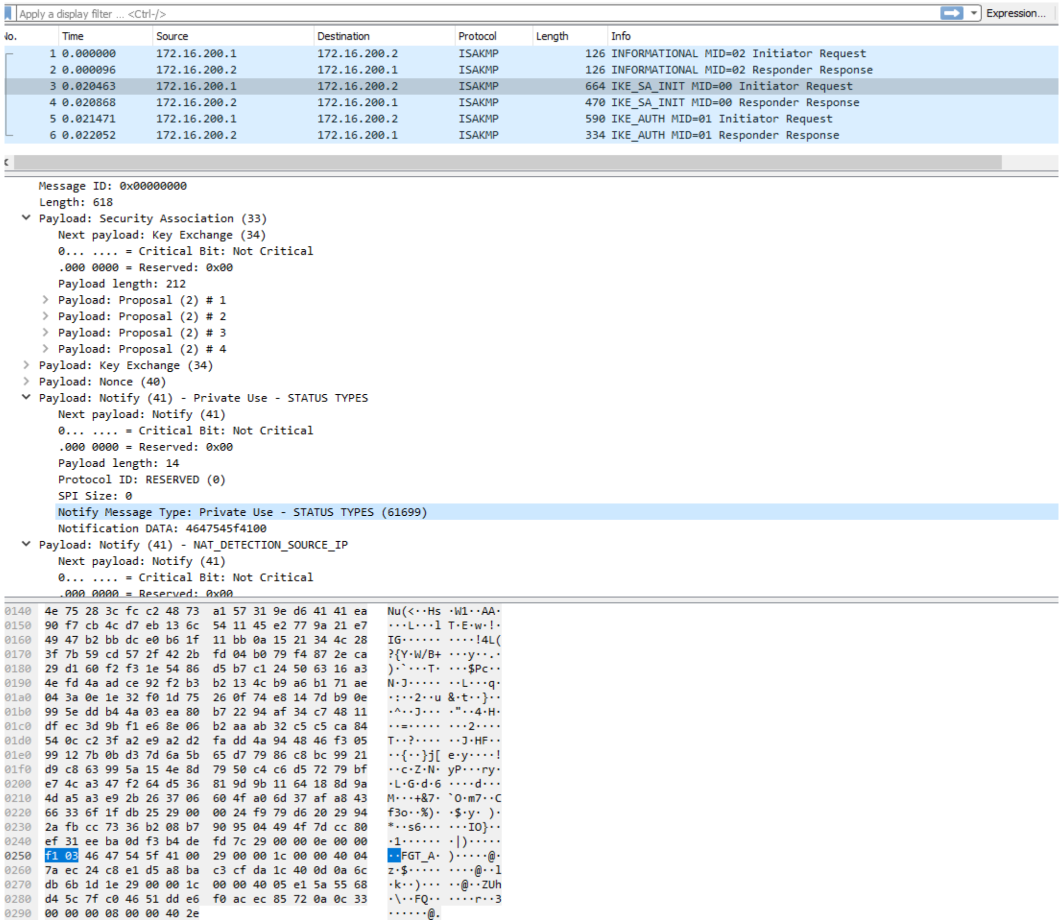
1. Configure the IPsec phase 1 settings:

```
config vpn ipsec phase1-interface
    edit "ul-port1"
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set localid "peerid-UNshTWcLQ22UNWqk0UwYtCQNtVhujrxAdyMG0qRsGVkx9mM8ksdaRZOF"
        set dpd on-idle
        set comments "[FGCONN] Do NOT edit. Automatically generated by extension 
controller."
        set dev-id-notification enable
        set dev-id "FGT_A"
        set remote-gw 172.16.200.2
        set psksecret ********
    next
end
```

2. Verify that the IPsec tunnel is established:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 3
------------------------------------------------------
name=ul-port1 ver=2 serial=3 172.16.200.1:0->172.16.200.2:0 tun_id=172.16.200.2 tun_
id6=::172.16.200.2 dst_mtu=1500 dpd-link=on weight=1
bound_if=19 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc
 run_state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=0 olast=0 ad=/0
stat: rxp=2689 txp=7115 rxb=278520 txb=617095
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=ul-port1 proto=0 sa=1 ref=3 serial=1
  src: 0:10.252.0.2-10.252.0.2:0
  dst: 0:10.252.0.1-10.252.0.1:0
  SA:  ref=6 options=10226 type=00 soft=0 mtu=1438 expire=41281/0B replaywin=2048
       seqno=1bca esn=0 replaywin_lastseq=00000a80 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42897/43200
  dec: spi=acf1f0fc esp=aes key=16 97d75ba10fbc904f14ce4a4caf8b4148
       ah=sha1 key=20 4ab706602068f9590314c4b16f53130a8011f410
  enc: spi=ca8de50b esp=aes key=16 8185ec9d2ecbb1d157663a6c199fc998
       ah=sha1 key=20 9430df55054152ab88e7372a322aad8f87688614
  dec:pkts/bytes=2690/278560, enc:pkts/bytes=14227/1632503
  npu_flag=03 npu_rgwy=172.16.200.2 npu_lgwy=172.16.200.1 npu_selid=2 dec_npuid=2 enc_
npuid=2
run_tally=0
```

3. Perform a packet capture of IPsec traffic (Wireshark is used in this example) and locate the initiator request IKE packet's NOTIFY message (type 61699).



Adjust DTLS heartbeat parameter for SSL VPN

This information is also available in the FortiOS 7.4 Administration Guide:

- Configuring the DTLS heartbeat parameters

The DTLS heartbeat parameters for SSL VPN can be adjusted. This improves the success rate of establishing a DTLS tunnel in networks with congestion or jitter.

```
config vpn ssl settings
    set dtls-heartbeat-idle-timeout <integer>
    set dtls-heartbeat-interval <integer>
    set dtls-heartbeat-fail-count <integer>
end
```

`dtls-heartbeat-idle-timeout <integer>`	Set the idle timeout before the DTLS heartbeat is sent, in seconds (3 - 10, default = 3).
`dtls-heartbeat-interval <integer>`	Set the interval between DTLS heartbeats, in seconds (3 - 10, default = 3).

`dtls-heartbeat-fail-count <integer>`	Set the number of missing heartbeats before the connection is considered dropped, in seconds (3 - 10, default = 3).

To configure the DTLS heartbeat parameters:

```
config vpn ssl settings
    set dtls-heartbeat-idle-timeout 3
    set dtls-heartbeat-interval 3
    set dtls-heartbeat-fail-count 3
end
```

To verify the configuration:

1. Run diagnostics on the client side:

```
# diagnose debug application sslvpn -1
# diagnose debug enable
00:00:03 S:058.000B(000.000B/s) R:000.000B(000.000B/s) Sd: 0 0.0000%DEBUG fsv_tun_send_
clt_hb:812 send heartbeat.
00:00:06 S:135.000B(000.000B/s) R:000.000B(000.000B/s) Sd: 0 0.0000%DEBUG fsv_client_on_
read:575 got type heartbeat
00:00:07 S:135.000B(000.000B/s) R:019.000B(019.000B/s) Sd: 0 0.0000%DEBUG fsv_tun_send_
clt_hb:812 send heartbeat.
00:00:09 S:154.000B(000.000B/s) R:019.000B(000.000B/s) Sd: 0 0.0000%DEBUG fsv_client_on_
read:575 got type heartbeat
00:00:10 S:154.000B(000.000B/s) R:038.000B(019.000B/s) Sd: 0 0.0000%DEBUG fsv_tun_send_
clt_hb:812 send heartbeat.
00:00:13 S:173.000B(000.000B/s) R:038.000B(000.000B/s) Sd: 0 0.0000%DEBUG fsv_tun_send_
clt_hb:812 send heartbeat.
```

   The heartbeat starts being sent after the idle timeout, and the heartbeat is sent every three seconds.

2. Run diagnostics on the server side:

```
root@auto-pc147:/home/auto# ./sslvpn/perf_test/fsvc-0.90/build/fsvc -s 10.1.100.2 -n
1443 -u u1 -p 1 --dtls -d debug
...
2023-04-26 12:01:40 [304:vdom1:5]sslvpn_send_ctrl_msg:987 0x7f1f2743e800 message:
heartbeat  10.1.100.147
2023-04-26 12:01:41 [304:vdom1:5]sslvpn_dtls_handle_client_data:758 0x7f1f2743e800 got
heartbeat
2023-04-26 12:01:44 [304:vdom1:5]sslvpn_dtls_handle_client_data:758 0x7f1f2743e800 got
heartbeat
2023-04-26 12:01:46 [304:vdom1:5]sslvpn_send_ctrl_msg:987 0x7f1f2743e800 message:
heartbeat  10.1.100.147
2023-04-26 12:01:50 [304:vdom1:5]sslvpn_send_ctrl_msg:987 0x7f1f2743e800 message:
heartbeat  10.1.100.147
2023-04-26 12:01:54 [304:vdom1:5]sslvpn_send_ctrl_msg:987 0x7f1f2743e800 message:
heartbeat  10.1.100.147
2023-04-26 12:01:54 [304:vdom1:5]sslvpn_dtls_timeout_check:358 no heartbeat received for
9 seconds.
2023-04-26 12:01:54 [304:vdom1:5]fsv_disassociate_fd_to_ipaddr:2367 deassociate
10.11.12.1 from tun (ssl.vdom1:12)
2023-04-26 12:01:54 [304:vdom1:5]dtls_tun_link_down:1884 tunnel device (12) closed
2023-04-26 12:01:54 [304:vdom1:5]tunnel is down, wait for next connection.
2023-04-26 12:01:54 [304:vdom1:5]sslvpn_release_dynip:1597 free app session, idx[0]
```

```
2023-04-26 12:01:54 [304:vdom1:5]release dyip
2023-04-26 12:01:54 [304:vdom1:5]Destroy sconn 0x7f1f2743e800, connSize=0. (vdom1)
```

The tunnel is disconnected once the `dtls-heartbeat-fail-count` is reached.

3. Use a Linux traffic control (tc) utility to introduce packet loss of 30% on the interface connected to the FortiGate (ens192):

```
root@auto-pc147:~# tc qdisc add dev ens192 root netem loss 30%
```

4. Run a ping test. The results show that the network has jitter/congestion as 33% of packets are being lost:

```
root@auto-pc147:~# ping 10.1.100.2 -c 100
PING 10.1.100.2 (10.1.100.2) 56(84) bytes of data.
64 bytes from 10.1.100.2: icmp_seq=1 ttl=255 time=0.111 ms
64 bytes from 10.1.100.2: icmp_seq=2 ttl=255 time=0.106 ms
...
64 bytes from 10.1.100.2: icmp_seq=99 ttl=255 time=0.103 ms
64 bytes from 10.1.100.2: icmp_seq=100 ttl=255 time=0.097 ms

--- 10.1.100.2 ping statistics ---
100 packets transmitted, 67 received, 33% packet loss, time 101382ms
rtt min/avg/max/mdev = 0.088/0.104/0.141/0.009 ms
```

5. Run diagnostics again on the server side to verify that the DTLS tunnel is established:

```
# diagnose debug application sslvpn -1
# diagnose debug enable
[307:vdom1:9]form_ipv4_pol_split_tunnel_addr:113 Matched policy (id = 14) to add ipv4
split tunnel routing address
[307:vdom1:9]SSL state:warning close notify (10.1.100.147)
[307:vdom1:9]sslConnGotoNextState:311 error (last state: 1, closeOp: 0)
[307:vdom1:9]Destroy sconn 0x7f1f27454800, connSize=0. (vdom1)
[307:vdom1:9]SSL state:warning close notify (10.1.100.147)
[304:vdom1:7]allocSSLConn:310 sconn 0x7f1f2743e800 (1:vdom1)
[304:vdom1:7]DTLS established: DTLSv1 ECDHE-RSA-AES256-GCM-SHA384 from 10.1.100.147
[304:vdom1:7]sslvpn_dtls_handle_client_data:693 got type clthello-tun
[304:vdom1:7]sslvpn_dtls_handle_client_data:780 unrecognized key: id=565b74d7
[304:vdom1:7]sslvpn_dtls_handle_client_data:703 got cookie:
kKi9WXUQfKg4Mxld66IQDr3/8krPAAiA/SvxcoKfnSfDOXvKKPOgMikJZGtBaSUX1lgPK6ke73XKF43o7FYzM7MV
VBY5CIRhLLnVtFP0DmqCqOz0uVtqQlUZWgtUtGz7hTl8O6VqPlnNgKX4PAY1Y+4GBqA2wG/giITeJ1Q1O7qmGzw0
UwNao27C2AJBul+ugbN44C60H+XMBcd2ggXjjdFSQfQrt4Jhnbn3hhnvQImEVypv/0t1S6D0H+z5DmYZEf9nCPux
0JICfGBhv6w1VXMhsasjSR3Jye049MM6xA9eCiqmUZW9DZfe
[304:vdom1:7]deconstruct_session_id:716 decode session id ok, user=[u1], group=[all_
groups],authserver=[],portal=[split_tunnel_portal],host[10.1.100.147],realm=[],csrf_
token=
[D840486CC92FEFC2B7F4EA46D8A455],idx=0,auth=1,sid=1db3f5f5,login=1682614961,access=16826
14961,saml_logout_url=no,pip=no,grp_info=[uwiuNn],rmt_grp_info=[]
[304:vdom1:7]tun dev (ssl.vdom1) opened (12)
[304:vdom1:7]fsv_associate_fd_to_ipaddr:2333 associate 10.11.12.1 to tun (ssl.vdom1:12)
[304:vdom1:7]proxy arp: scanning 26 interfaces for IP 10.11.12.1
[304:vdom1:7]no ethernet address for proxy ARP
[304:vdom1:7]sslvpn_user_match:1170 add user u1 in group all_groups
[304:vdom1:7]Will add auth policy for policy 14
[304:vdom1:7]Add auth logon for user u1:all_groups, matched group number 2
[304:vdom1:7]sslvpn_send_ctrl_msg:987 0x7f1f2743e800 message: svrhello-tun ok
10.1.100.147
[304:vdom1:7]sslvpn_dtls_handle_client_data:758 0x7f1f2743e800 got heartbeat
```

```
[304:vdom1:7]sslvpn_send_ctrl_msg:987 0x7f1f2743e800 message: heartbeat  10.1.100.147
[304:vdom1:7]sslvpn_dtls_handle_client_data:758 0x7f1f2743e800 got heartbeat
[304:vdom1:7]sslvpn_dtls_handle_client_data:758 0x7f1f2743e800 got heartbeat
[304:vdom1:7]sslvpn_send_ctrl_msg:987 0x7f1f2743e800 message: heartbeat  10.1.100.147
```

SAML-based authentication for FortiClient remote access dialup IPsec VPN clients

SAML-based authentication for FortiClient remote access dialup IPsec VPN clients is now supported.

The FortiGate authd daemon has been enhanced to support SAML authentication and accepts local-in traffic from the FortiClient by the TCP port number configured in the `auth-ike-saml-port` setting.

The `ike-saml-server` setting enables a configured SAML server to listen on a FortiGate interface for SAML authentication requests from FortiClient remote access IPsec VPN clients.

For more information about this feature, see SAML-based authentication for FortiClient remote access dialup IPsec VPN clients.

Multiple interface monitoring for IPsec - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Multiple interface monitoring for IPsec

IPsec can monitor multiple interfaces per tunnel, and activate a backup link only when all of the primary links are down. This can be useful if you have multiple WAN links and want to optimize your WAN link selection and performance while limiting the use of more expensive and bandwidth intensive interfaces, like 5G or LTE.

In cases where multiple primary overlays are deployed and the backup overlay is on an LTE connection, avoiding IPsec keep alive messages, BGP hellos, and SD-WAN health checks on the backup connection is required when the primary overlays are working. The backup overlay can monitor all of the primary overlays, and is not activated until the number of unhealthy primary overlays equals or surpasses the predefined threshold.

```
config vpn ipsec phase1-interface
    edit <phase-1 name>
        set monitor <overlay> <overlay> ... <overlay>
        set monitor-min <integer>
    next
end
```

`monitor`	The IPsec interfaces to monitor.
`monitor-min`	The minimum number of monitored interfaces that must become degraded before this interface is activated (0 = all interfaces, default = 0).

In this example, four primary overlays are configured, T1 - T4, on fixed broadband connections and one backup overlay, T5, is configured on an LTE connection.

The backup overlay stays down as long as the primary overlays are working normally. When all four of the primary overlays go down, the backup overlay is activated and used to forward traffic. If any of the primary overlays recover, then the backup overlay goes down.

SD-WAN can also be configured to steer traffic.

To configure the overlays:

1. Configure the VPN remote gateways:

```
config vpn ipsec phase1-interface
    edit "T1"
        set interface "dmz"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256
        set remote-gw 172.16.208.2
        set psksecret **********
    next
    edit "T2"
        set interface "agg1"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256
        set remote-gw 172.16.203.2
        set psksecret **********
    next
    edit "T3"
        set interface "vlan100"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256
        set remote-gw 172.16.206.2
        set psksecret **********
    next
    edit "T4"
        set interface "port15"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256
        set remote-gw 172.16.209.2
        set psksecret **********
    next
    edit "T5"
        set interface "vlan200"
        set ike-version 2
        set peertype any
        set monitor "T1" "T2" "T3" "T4"
        set monitor-min 4
        set net-device disable
        set proposal aes128-sha256
        set remote-gw 172.16.210.2
        set psksecret **********
    next
end
```

2. Configure the VPN tunnels:

```
config vpn ipsec phase2-interface
    edit "T1_P2"
        set phase1name "T1"
        set proposal aes256-sha256
        set auto-negotiate enable
    next
    edit "T2_P2"
        set phase1name "T2"
        set proposal aes256-sha256
        set auto-negotiate enable
    next
    edit "T3_P2"
        set phase1name "T3"
        set proposal aes256-sha256
        set auto-negotiate enable
    next
    edit "T4_P2"
        set phase1name "T4"
        set proposal aes256-sha256
        set auto-negotiate enable
    next
    edit "T5_P2"
        set phase1name "T5"
        set proposal aes256-sha256
        set auto-negotiate enable
    next
end
```

3. Configure the interfaces:

```
config system interface
    edit "T1"
        set vdom "root"
        set ip 100.1.1.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 100.1.1.2 255.255.255.0
        set snmp-index 113
        set interface "dmz"
    next
    edit "T2"
        set vdom "root"
        set ip 100.1.2.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 100.1.2.2 255.255.255.0
        set snmp-index 114
        set interface "agg1"
    next
    edit "T3"
        set vdom "root"
        set ip 100.1.3.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 100.1.3.2 255.255.255.0
        set snmp-index 115
        set interface "vlan100"
```

```
    next
    edit "T4"
        set vdom "root"
        set ip 100.1.4.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 100.1.4.2 255.255.255.0
        set snmp-index 65
        set interface "port15"
    next
    edit "T5"
        set vdom "root"
        set ip 100.1.5.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 100.1.5.2 255.255.255.0
        set snmp-index 117
        set interface "vlan200"
    next
end
```

4. Check the IPsec tunnel summary:

```
# get vpn ipsec tunnel summary
'T2' 172.16.203.2:0  selectors(total,up): 1/1  rx(pkt,err): 0/0  tx(pkt,err): 0/4
'T3' 172.16.206.2:0  selectors(total,up): 1/1  rx(pkt,err): 0/0  tx(pkt,err): 0/4
'T4' 172.16.209.2:0  selectors(total,up): 1/1  rx(pkt,err): 0/0  tx(pkt,err): 0/4
'T5' 172.16.210.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/4
'T1' 172.16.208.2:0  selectors(total,up): 1/1  rx(pkt,err): 0/0  tx(pkt,err): 0/4
```

The backup overlay, T5, is down.

To configure steering traffic with SD-WAN:

1. Configure the SD-WAN:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "T1"
        next
        edit 2
            set interface "T2"
        next
        edit 3
            set interface "T3"
        next
        edit 4
            set interface "T4"
        next
        edit 5
            set interface "T5"
```

```
        next
    end
    config service
        edit 1
            set name "1"
            set load-balance enable
            set dst "all"
            set src "172.16.205.0"
            set priority-members 1 2 3 4 5
        next
    end
end
```

2. Configure a static route:

```
config router static
    edit 5
        set dst 8.0.0.0 255.0.0.0
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
end
```

3. Check the routing table:

```
# get router info routing-table static
Routing table for VRF=0
S       8.0.0.0/8 [1/0] via T2 tunnel 172.16.203.2, [1/0]
                  [1/0] via T3 tunnel 172.16.206.2, [1/0]
                  [1/0] via T1 tunnel 172.16.208.2, [1/0]
                  [1/0] via T4 tunnel 172.16.209.2, [1/0]
```

Check the results:

- When both the T1 and T2 connections are down, T5 stays down as well, and traffic is load-balanced on T3 and T4 by the SD-WAN configuration:

```
 # get vpn ipsec tunnel summary
'T2' 172.16.203.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
'T3' 172.16.206.2:0  selectors(total,up): 1/1  rx(pkt,err): 0/0  tx(pkt,err): 0/0
'T4' 172.16.209.2:0  selectors(total,up): 1/1  rx(pkt,err): 0/0  tx(pkt,err): 0/4
'T5' 172.16.210.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/4
'T1' 172.16.208.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
```

```
# get router info routing-table static
Routing table for VRF=0
S       8.0.0.0/8 [1/0] via T3 tunnel 172.16.206.2, [1/0]
                  [1/0] via T4 tunnel 172.16.209.2, [1/0]
```

Traffic is load-balanced between the remaining tunnels:

```
# diagnose sniffer packet any 'host 8.8.8.8' 4
interfaces=[any]
filters=[host 8.8.8.8]
3.027055 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
3.027154 T4 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
3.031434 T4 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
3.031485 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
```

```
3.612818 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
3.612902 T3 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
3.617107 T3 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
3.617159 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
4.168845 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
4.168907 T4 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
4.173150 T4 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
4.173174 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
4.710907 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
4.710991 T3 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
4.715933 T3 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
4.715958 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
```

- When all of the primary overlays are down, T5 is activated and used for traffic

  ```
  # get vpn ipsec tunnel summary
  'T2' 172.16.203.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
  'T3' 172.16.206.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
  'T4' 172.16.209.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
  'T5' 172.16.210.2:0  selectors(total,up): 1/1  rx(pkt,err): 0/0  tx(pkt,err): 0/4
  'T1' 172.16.208.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
  ```

  ```
  # get router info routing-table static
  Routing table for VRF=0
  S       8.0.0.0/8 [1/0] via T5 tunnel 172.16.210.2, [1/0]
  ```

  Traffic is using the backup overlay, T5:

  ```
  # diagnose sniffer packet any 'host 8.8.8.8' 4
  interfaces=[any]
  filters=[host 8.8.8.8]
  1.907944 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
  1.908045 T5 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
  1.912283 T5 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
  1.912351 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
  2.665921 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
  2.665999 T5 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
  2.670209 T5 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
  2.670235 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
  5.269997 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
  5.270090 T5 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
  5.274275 T5 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
  5.274300 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
  5.781848 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
  5.781920 T5 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
  5.786334 T5 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
  5.786363 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
  ```

- If T4 recovers, T5 is deactivated and traffic switches to T4:

  ```
  # get vpn ipsec tunnel summary
  'T2' 172.16.203.2:0  selectors(total,up): 2/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
  'T3' 172.16.206.2:0  selectors(total,up): 2/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
  'T4' 172.16.209.2:0  selectors(total,up): 2/2  rx(pkt,err): 0/0  tx(pkt,err): 0/0
  'T5' 172.16.210.2:0  selectors(total,up): 2/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
  'T1' 172.16.208.2:0  selectors(total,up): 2/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
  ```

```
# get router info routing-table static
Routing table for VRF=0
S       8.0.0.0/8 [1/0] via T4 tunnel 172.16.209.2, [1/0]
```

The primary overlay T4 has recovered, and the backup overlay is down again:

```
# diagnose sniffer packet any 'host 8.8.8.8' 4
interfaces=[any]
filters=[host 8.8.8.8]
4.555685 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
4.555790 T4 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
4.560428 T4 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
4.560478 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
5.163223 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
5.163332 T4 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
5.167590 T4 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
5.167620 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
5.650089 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
5.650194 T4 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
5.654352 T4 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
5.654387 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
6.102181 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
6.102263 T4 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
6.106411 T4 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
6.106445 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
```

Update SSL VPN default behavior and visibility in the GUI - 7.4.1

SSL VPN default behavior and visibility in the GUI have been updated:

- By default, SSL VPN web mode settings are disabled and hidden from the GUI and the CLI.
- By default, SSL VPN tunnel mode settings and the *VPN > SSL-VPN* menus are hidden from the GUI.
- The CLI configuration setting for VPN GUI feature visibility has been divided into IPsec (`set gui-vpn` under `config system settings`) and SSL-VPN (`set gui-sslvpn` under `config system settings`), where IPsec is still enabled by default and SSL-VPN is now disabled by default.
- Warning messages have been added to the GUI on the *SSL-VPN Settings* page under *SSL-VPN status* and *Authentication/Portal Mapping* when either SSL VPN tunnel mode or SSL web mode is enabled.
- In *Security Fabric > Security Rating*, a new check for *Disable SSL-VPN Settings* has been added and this check fails whenever SSL VPN is enabled.

To enable SSL VPN web mode:

```
config system global
    set sslvpn-web-mode enable
end
```

To enable the *VPN > SSL-VPN* GUI menus:

```
config system settings
    set gui-sslvpn enable
end
```

If SSL VPN web mode and tunnel mode were configured in a FortiOS firmware version prior to upgrading to FortiOS 7.4.1 and above, then the *VPN > SSL-VPN* menus and *SSL VPN web mode* settings remain visible in the GUI.

In FortiOS, alternative remote access solutions are IPsec VPN and ZTNA.

Upgrading devices with SSL VPN already configured

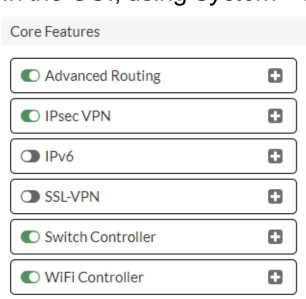
This table summarizes the SSL VPN visibility CLI configuration based on whether a device has been factory reset or has been upgraded with SSL VPN already configured:

Behavior in FortiOS 7.4.1 and above	SSL VPN web mode	SSL VPN tunnel mode	`set sslvpn-web-mode`	`set gui-sslvpn`
After factory reset	GUI and CLI disabled	GUI disabled	`disable`	`disable`
After upgrade when SSL VPN web mode and SSL VPN tunnel mode previously not enabled	GUI and CLI disabled	GUI disabled	`disable`	`disable`
After upgrade when only SSL VPN tunnel mode previously enabled	GUI and CLI disabled	GUI enabled	`disable`	`enable`
After upgrade when both SSL VPN web mode and SSL VPN tunnel mode previously enabled	GUI and CLI enabled	GUI enabled	`enable`	`enable`

SSL VPN menu visibility

By default, hide *VPN > SSL-VPN* menus for tunnel mode from the GUI, namely, *SSL-VPN Portals*, *SSL-VPN Settings*, and *SSL-VPN Clients*. This visibility is configurable.

- In the GUI, using *System > Feature Visibility*:



- In the CLI, using this configuration setting:

```
config system settings
    set gui-sslvpn disable
end
```

When SSL-VPN is enabled using either the GUI or CLI method, these VPN menus will become visible:

- *SSL-VPN Portals*
- *SSL-VPN Settings*

- *SSL-VPN Clients*

SSL VPN web mode visibility

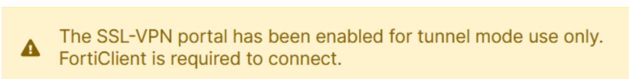
By default, hide SSL VPN web mode from the GUI using a CLI configuration setting:

```
config system global
    set sslvpn-web-mode disable
end
```

When SSL VPN web mode is hidden, the following elements are hidden:

- The *Web Mode Settings* section from the *SSL-VPN Settings* page.
- The *web-access* portal from the *SSL-VPN Portals* page.
- The *Web Mode* setting is disabled from within portals with a warning message.

If SSL VPN web mode is hidden from the GUI using the above CLI command, even though SSL VPN tunnel mode has been correctly configured, when you try to access SSL VPN web mode using the SSL VPN portal by navigating to the listening IP address, domain, and port using a web browser, you will see the following warning message:



VPN feature visibility

By default, VPN feature visibility is enabled:

```
config system settings
    set gui-vpn enable
end
```

Starting in FortiOS 7.4.1, this CLI setting no longer enables both IPsec VPN and SSL VPN feature visibility and has been updated to control IPsec VPN feature visibility only:

```
config system settings
    set gui-vpn {enable | disable}
end
```

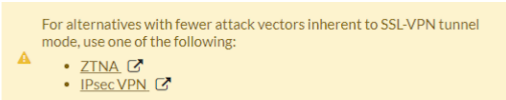
Setting	Description
`enable`	Enable the IPsec VPN settings pages on the GUI.
`disable`	Disable the IPsec VPN settings pages on the GUI.

Warning messages when SSL VPN is configured

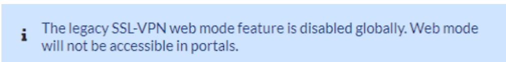
Warning messages have been added to the GUI on the *VPN > SSL-VPN Settings* page to inform the administrator of remote access alternatives.

The following warning messages are displayed with a yellow and blue banner, respectively, when SSL VPN tunnel mode is enabled and web mode is disabled:

- The yellow warning is displayed in the opening section of *VPN > SSL-VPN Settings*.

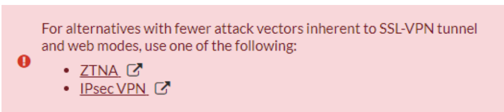


- The blue warning is displayed in the *Authentication/Portal Mapping* section of *VPN > SSL-VPN Settings*.

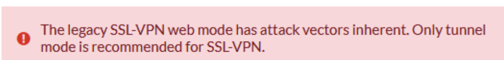


The following warning messages are displayed with red banners when SSL VPN tunnel mode and web mode are both enabled:

- The first warning is displayed in the opening section of *VPN > SSL-VPN Settings*.

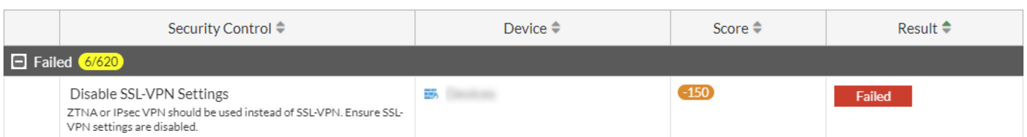


- The second warning is displayed in the *Authentication/Portal Mapping* section of *VPN > SSL-VPN Settings*.



Security Rating check for disabling SSL VPN settings

In *Security Fabric > Security Rating*, add a check for *Disable SSL-VPN Settings* and have this check fail when SSL VPN is enabled.



When SSL VPN settings are enabled, this security rating check will fail because Fortinet Inc. Security Best Practices (FSBP) suggest using ZTNA or IPsec VPN instead of SSL VPN. This page will display IPsec VPN and ZTNA help links in the *Recommendations* section.

Securely exchange serial numbers between FortiGates connected with IPsec VPN - 7.4.1

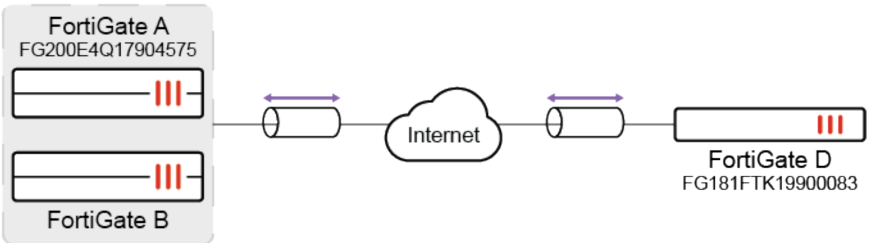
This information is also available in the FortiOS 7.4 Administration Guide:

- Securely exchange serial numbers between FortiGates connected with IPsec VPN

Serial numbers can be securely exchanged between FortiGates connected with IPsec VPN. This feature is supported in IKEv2, IKEv1 main mode, and IKEv1 aggressive mode. The exchange is only performed with participating FortiGates that have enabled the `exchange-fgt-device-id` setting under `config vpn ipsec phase1-interface`.

Example

In this example, FortiGates A and B are in an HA cluster, so the serial numbers will not exchange after failover. The cluster is connected to FortiGate D through IPsec VPN.



To securely exchange serial numbers between the FortiGates:

1. Configure the IPsec settings on FortiGate A.
   a. Configure the phase 1 interface settings:

```
config vpn ipsec phase1-interface
    edit "to_FGTD"
        set interface "port1"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set exchange-fgt-device-id enable
        set remote-gw 172.16.200.4
        set psksecret **********
    next
end
```

   b. Configure the phase 2 interface settings:

```
config vpn ipsec phase2-interface
    edit "to_FGTD"
        set phase1name "to_FGTD"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set src-addr-type name
        set dst-addr-type name
        set src-name "to_FGTD_local"
        set dst-name "to_FGTD_remote"
    next
end
```

2. Configure the IPsec settings on FortiGate D.
   a. Configure the phase 1 interface settings:

```
config vpn ipsec phase1-interface
    edit "to_FGTA"
        set interface "port2"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set exchange-fgt-device-id enable
```

```
        set remote-gw 172.16.200.1
        set psksecret **********
    next
end
```

b. Configure the phase 2 interface settings:

```
config vpn ipsec phase2-interface
    edit "to_FGTA"
        set phase1name "to_FGTA"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set src-addr-type name
        set dst-addr-type name
        set src-name "to_FGTA_local"
        set dst-name "to_FGTA_remote"
    next
end
```

3. Verify the peer serial numbers.
   a. On FortiGate A:

```
# diagnose vpn ike gateway list

vd: root/0
name: to_FGTD
version: 1
interface: port1 19
addr: 172.16.200.1:500 -> 172.16.200.4:500
tun_id: 172.16.200.4/::172.16.200.4
remote_location: 0.0.0.0
network-id: 0
created: 783s ago
peer-id: 172.16.200.4
peer-id-auth: no
peer-SN: FG181FTK19900083
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 2 a8b2df203ef134e8/955fafbd10a04fa0
  direction: initiator
  status: established 783-783s ago = 0ms
  proposal: aes128-sha256
  key: 644db099e1178d1f-119fee3141f1e2a6
  lifetime/rekey: 86400/85316
  DPD sent/recv: 00000000/00000000
  peer-id: 172.16.200.4
```

   b. On FortiGate D:

```
# diagnose vpn ike gateway list

vd: root/0
name: to_FGTA
version: 1
interface: port2 10
addr: 172.16.200.4:500 -> 172.16.200.1:500
```

```
tun_id: 172.16.200.1/::172.16.200.1
remote_location: 0.0.0.0
network-id: 0
created: 723s ago
peer-id: 172.16.200.1
peer-id-auth: no
peer-SN: FG200E4Q17904575
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 0/0

  id/spi: 7 a8b2df203ef134e8/955fafbd10a04fa0
  direction: responder
  status: established 723-723s ago = 10ms
  proposal: aes128-sha256
  key: 644db099e1178d1f-119fee3141f1e2a6
  lifetime/rekey: 86400/85406
  DPD sent/recv: 00000000/00000000
  peer-id: 172.16.200.1
```

4. After an HA failover, verify that the peer serial numbers have not changed.
   a. On FortiGate B:

```
# diagnose vpn ike gateway  list

vd: root/0
name: to_FGTD
version: 2
interface: port1 19
addr: 172.16.200.1:500 -> 172.16.200.4:500
tun_id: 172.16.200.4/::172.16.200.4
remote_location: 0.0.0.0
network-id: 0
created: 104s ago
peer-id: 172.16.200.4
peer-id-auth: no
peer-SN: FG181FTK19900083
PPK: no
IKE SA: created 1/2  established 1/2  time 0/0/0 ms
IPsec SA: created 1/2  established 1/2  time 0/0/0 ms

  id/spi: 8 3aab6778ea613bcd/e28dd0a1251a2eb1
  direction: responder
  status: established 101-101s ago = 0ms
  proposal: aes128-sha256
  child: no
  SK_ei: c05f59ac726e4c3c-0d273aa8bf5dde35
  SK_er: 5be947724fbbd85b-d1e090a757823e6a
  SK_ai: 11f85a5c896a897f-2d7a551a91d5c1e2-63394ec02414ddb2-33598a09e77c8207
  SK_ar: 4291445e00062982-f7c5a848c9ada403-6ce7e4394e3a4fd5-bf2dc03492576cfc
  PPK: no
  message-id sent/recv: 12/3
  lifetime/rekey: 86400/86028
  DPD sent/recv: 00000000/00000000
  peer-id: 172.16.200.4
```

   b. On FortiGate D:

```
# diagnose vpn ike gateway list

vd: root/0
name: to_FGTA
version: 2
interface: port2 10
addr: 172.16.200.4:500 -> 172.16.200.1:500
tun_id: 172.16.200.1/::172.16.200.1
remote_location: 0.0.0.0
network-id: 0
created: 132s ago
peer-id: 172.16.200.1
peer-id-auth: no
peer-SN: FG200E4Q17904575
PPK: no
IKE SA: created 1/2  established 1/2  time 0/10500/21000 ms
IPsec SA: created 1/2  established 1/2  time 0/10500/21000 ms

  id/spi: 9 3aab6778ea613bcd/e28dd0a1251a2eb1
  direction: initiator
  status: established 132-111s ago = 21000ms
  proposal: aes128-sha256
  child: no
  SK_ei: c05f59ac726e4c3c-0d273aa8bf5dde35
  SK_er: 5be947724fbbd85b-d1e090a757823e6a
  SK_ai: 11f85a5c896a897f-2d7a551a91d5c1e2-63394ec02414ddb2-33598a09e77c8207
  SK_ar: 4291445e00062982-f7c5a848c9ada403-6ce7e4394e3a4fd5-bf2dc03492576cfc
  PPK: no
  message-id sent/recv: 3/12
  lifetime/rekey: 86400/85988
  DPD sent/recv: 00000000/00000000
  peer-id: 172.16.200.1
```

To retrieve the peer serial number in FortiManager:

1. Add and authorize FortiGate A (see Adding online devices using Discover mode for more details).
2. Go to *Device Manager > Device & Groups* and select the FortiGate A.
3. Add the *IPsec VPN* widget (see Customizing the dashboard for more details).
4. Open the developer tools in your browser and select the *Network* tab.
5. Refresh the *IPsec VPN* widget.
6. In the *Network* tab, there should be a JSON POST request that FortiManager will proxy request to the FortiGate for the IPsec API. The response should contain the peer serial number.

IPsec split DNS - 7.4.1

This functionality empowers clients to determine whether DNS traffic should utilize the tunnel's DNS or the local DNS server for query resolution. This is achieved by letting users specify a list of FQDNs. Only FQDNs that match the specified list are directed to the tunnel for resolution, while all other queries are handled by the local DNS server.

For more information about this feature, see Enhancing IPsec security and performance.

Support IPsec tunnel to change names - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- Renaming IPsec tunnels

IPsec tunnels can be renamed. When you rename an IPsec tunnel, all references to the tunnel, such as routing and policies, are automatically updated to reflect the new name.

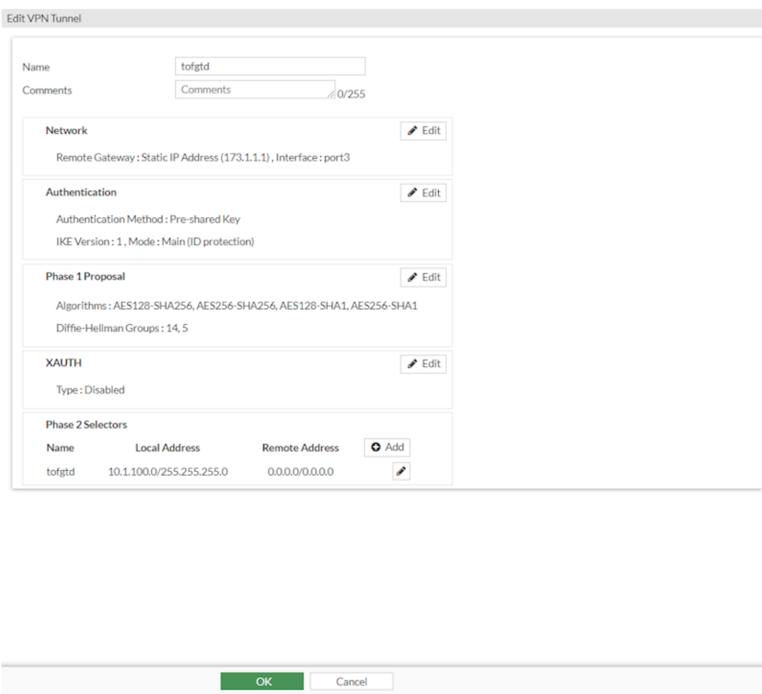
```
config vpn ipsec phase1-interface
    rename <string> to <string>
end
```

Example

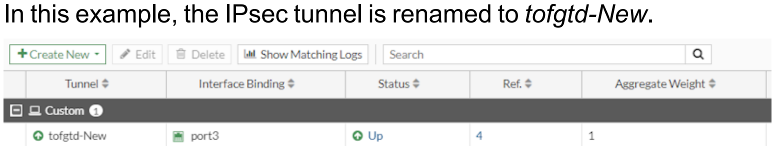
In this example, the IPsec tunnel named *tofgtd* is renamed to *tofgtd-New*, and all associated references are updated.

To rename an IPsec tunnel in the GUI:

1. Go to *VPN > IPsec Tunnels* and double-click an IPsec tunnel to open it for editing.
   In this example, the IPsec tunnel name is *tofgtd*.



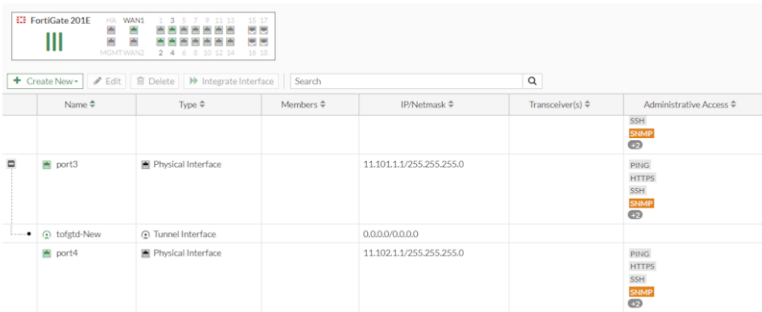
2. In the *Name* box, type a new name, and click *OK*. The IPsec tunnel is renamed, and all associated references are updated.
   In this example, the IPsec tunnel is renamed to *tofgtd-New*.



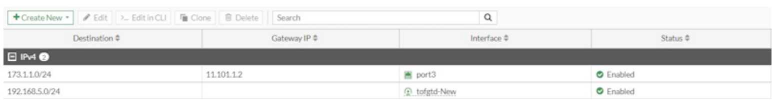
3. Check the associated references:

   In this example, all associated references show the new IPsec tunnel name of *tofgtd-New*.

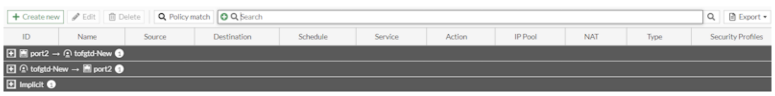
   - Go to *Network* > *Interfaces* to see that the interface references the new IPsec tunnel name.



   - Go to *Network* > *Static Routes* to see that the static route references the new IPsec tunnel name.



   - Go to *Policy & Objects* > *Firewall Policy* to see that the policy references the new IPsec tunnel name



To rename an IPsec tunnel in the CLI:

1. Rename the IPsec tunnel.

   In this example, the IPsec tunnel named *tofgtd* is renamed to *tofgtd-New*:

```
config vpn ipsec phase1-interface
    rename tofgtd to tofgtd-New
end
```

2. Show the configuration to confirm that the IPsec tunnel was renamed.

   In this example, the IPsec tunnel was renamed to *tofgtd-New*:

```
show
config vpn ipsec phase1-interface
    edit "tofgtd-New"
        set interface "port3"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd disable
        set remote-gw 173.1.1.1
        ...
    next
end
```

3. Check the associated references.

   In this example, all associated references show the new IPsec tunnel name of *tofgtd-New*.

   - Confirm that the interfaces reference the new IPsec tunnel name:

```
config router static
show
```

```
config router static
    edit 3
        set dst 192.168.5.0 255.255.255.0
        set device "tofgtd-New"
    next
end
```

- Confirm that the static route references the new IPsec tunnel name:

```
config system interface
show
    edit "tofgtd-New"
    ....
end
```

- Confirm that the policies references the new IPsec tunnel name:

```
config firewall policy
show
config firewall policy
    edit 1
        set uuid 802c6c2e-8368-51ee-bf40-6c3c32da1024
        set srcintf "port2"
        set dstintf "tofgtd-New"
        set action accept
        ...
    next
    edit 2
        set uuid 80d136aa-8368-51ee-cc52-b0b06306fb80
        set srcintf "tofgtd-New"
        set dstintf "port2"
        set action accept
        ...
    next
end
```

Encapsulate ESP packets within TCP headers - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- Encapsulate ESP packets within TCP headers

FortiOS includes a proprietary solution to support the encapsulation of Encapsulating Security Payload (ESP) packets within Transmission Control Protocol (TCP) headers. This allows ESP packets to be assigned a port number, which enables them to traverse over carrier networks where direct IPsec traffic is blocked or impeded by carrier-grade NAT.

This feature only works with IKE version 2, and it does not support ADVPN.

To configure the TCP port for IKE/IPsec traffic:

```
config system settings
    set ike-tcp-port <integer>
end
```

`ike-tcp-port <integer>`	Set the TCP port for IKE/IPsec traffic (1 - 65535, default = 4500).

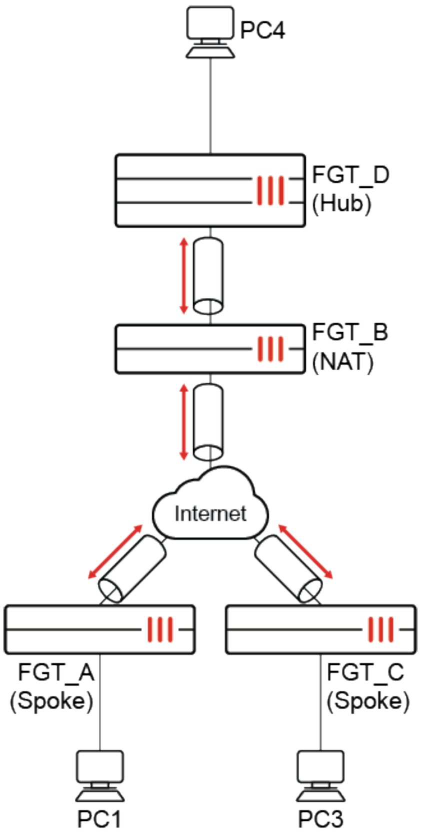
To configure ESP encapsulation on the phase 1 interface:

```
config vpn ipsec phase1-interface
    edit <name>
        set ike-version 2
        set transport {udp | udp-fallback-tcp | tcp}
        set fortinet-esp {enable | disable}
        set fallback-tcp-threshold <integer>
    next
end
```

`transport {udp | udp-fallback-tcp | tcp}`	Set the IKE transport protocol. • `udp`: use UDP transport for IKE. • `udp-fallback-tcp`: use UDP transport for IKE, with fallback to TCP transport. • `tcp`: use TCP transport for IKE.
`fortinet-esp {enable | disable}`	Enable/disable Fortinet ESP encapsulation.
`fallback-tcp-threshold <integer>`	Set the timeout before IKE/IPsec traffic falls back to TCP, in seconds (1 - 300, default = 15).

Example

In this example, IPsec VPN crosses over a carrier network and UDP packets are not allowed.



To encapsulate ESP packets within TCP headers:

1. On each FortiGate, configure the IKE TCP port setting:

```
config system settings
    set ike-tcp-port 1443
end
```

2. Disable anti-replay in the global settings on the FGT_B (NAT) FortiGate (see step 7 for more information):

```
config system global
    set anti-replay disable
    set hostname "FGT-B"
end
```

3. Configure the FGT_A (spoke) FortiGate.
   a. Configure the IPsec phase 1 settings:

```
config vpn ipsec phase1-interface
    edit "spoke"
        set interface "wan1"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-
prfsha384 chacha20poly1305-prfsha256
        set transport tcp
        set fortinet-esp enable
```

```
        set remote-gw 173.1.1.1
        set psksecret **********
    next
end
```

b. Configure the IPsec phase 2 settings:

```
config vpn ipsec phase2-interface
    edit "spoke"
        set phase1name "spoke"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set src-subnet 10.1.100.0 255.255.255.0
    next
end
```

IKE and ESP will be encapsulated into TCP, and ESP packets encapsulated into a fake TCP header.

4. Configure the FGT_C (spoke) FortiGate.
   a. Configure the IPsec phase 1 settings:

```
config vpn ipsec phase1-interface
    edit "Spoke"
        set interface "wan1"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-
prfsha384 chacha20poly1305-prfsha256
        set transport udp-fallback-tcp
        set fortinet-esp enable
        set fallback-tcp-threshold 10
        set remote-gw 173.1.1.1
        set psksecret **********
    next
end
```

   b. Configure the IPsec phase 2 settings:

```
config vpn ipsec phase2-interface
    edit "Spoke"
        set phase1name "Spoke"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set src-subnet 192.168.4.0 255.255.255.0
    next
end
```

IKE will use UDP encapsulation first. If it fails to establish in 10 seconds, it will fall back to TCP. ESP packets are encapsulated into a fake TCP header.

5. Configure the FGT_D (hub) FortiGate.
   a. Configure the IPsec phase 1 settings:

```
config vpn ipsec phase1-interface
    edit "Hub"
        set type dynamic
        set interface "port25"
        set ike-version 2
```

```
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-
prfsha384 chacha20poly1305-prfsha256
        set dpd on-idle
        set transport tcp
        set fortinet-esp enable
        set psksecret **********
        set dpd-retryinterval 60
    next
end
```

b. Configure the IPsec phase 2 settings:

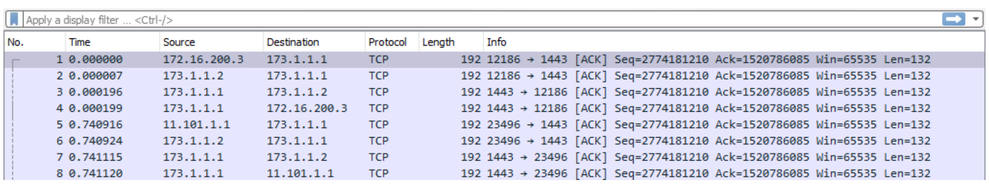
```
config vpn ipsec phase2-interface
    edit "Hub"
        set phase1name "Hub"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
    next
end
```

6. Verify the IPsec VPN tunnel state on FGT_D (hub):

```
# diagnose vpn ike gateway list

vd: root/0
name: Hub_0
version: 2
interface: port25 33
addr: 173.1.1.1:1443 -> 173.1.1.2:23496
tun_id: 173.1.1.2/::10.0.0.4
remote_location: 0.0.0.0
network-id: 0
transport: TCP
created: 733s ago
peer-id: 11.101.1.1
peer-id-auth: no
nat: peer
PPK: no
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 3 f050ac7a151a3b31/3b46b71108eea2e2
  direction: responder
  status: established 733-733s ago = 0ms
  proposal: aes128-sha256
  child: no
  SK_ei: 619dfbeb679345f7-531692a72da85727
  SK_er: 5b6a1625b2ce71cf-13b339289ca99b9d
  SK_ai: a61818128c0d5390-b6d15cf9eb58e0f6-4e8c552e6265387b-4f79dc3acdd5d092
  SK_ar: 64fb56b13ee65bd2-6ea1fb268b3ffad9-818c8e4d302a1176-c8978a8ce91d9856
  PPK: no
  message-id sent/recv: 11/2
  QKD: no
  lifetime/rekey: 86400/85396
  DPD sent/recv: 0000000c/0000000c
```

```
  peer-id: 11.101.1.1

vd: root/0
name: Hub_2
version: 2
interface: port25 33
addr: 173.1.1.1:1443 -> 173.1.1.2:12186
tun_id: 10.0.0.4/::10.0.0.6
remote_location: 0.0.0.0
network-id: 0
transport: TCP
created: 645s ago
peer-id: 172.16.200.3
peer-id-auth: no
nat: peer
PPK: no
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 17 7eb5a40cd324d2fc/f04fec6d8d77d996
  direction: responder
  status: established 645-645s ago = 0ms
  proposal: aes128-sha256
  child: no
  SK_ei: c1fe2027086b046b-0f15c6e2d25a255d
  SK_er: 3eac9a73b4dd2961-900c0af7f0e18abf
  SK_ai: e21ca3934cca7a85-af425d12baf40693-0c30e3f6d98a6a7d-273b33cc49155092
  SK_ar: 1bef95d13784e8e1-9894c1b3628e158a-3cbfe4f7a730d9de-c9150844e3ff2002
  PPK: no
  message-id sent/recv: 10/2
  QKD: no
  lifetime/rekey: 86400/85484
  DPD sent/recv: 0000000b/0000000b
  peer-id: 172.16.200.3
```

7. Verify the ESP packets sniffed on the NAT device.



Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.200.3	173.1.1.1	TCP	192	12186 → 1443 [ACK] Seq=2774181210 Ack=1520786085 Win=65535 Len=132
2	0.000007	173.1.1.2	173.1.1.1	TCP	192	12186 → 1443 [ACK] Seq=2774181210 Ack=1520786085 Win=65535 Len=132
3	0.000196	173.1.1.1	173.1.1.2	TCP	192	1443 → 12186 [ACK] Seq=2774181210 Ack=1520786085 Win=65535 Len=132
4	0.000199	173.1.1.1	172.16.200.3	TCP	192	1443 → 12186 [ACK] Seq=2774181210 Ack=1520786085 Win=65535 Len=132
5	0.740916	11.101.1.1	173.1.1.1	TCP	192	23496 → 1443 [ACK] Seq=2774181210 Ack=1520786085 Win=65535 Len=132
6	0.740924	173.1.1.2	173.1.1.1	TCP	192	23496 → 1443 [ACK] Seq=2774181210 Ack=1520786085 Win=65535 Len=132
7	0.741115	173.1.1.1	173.1.1.2	TCP	192	1443 → 23496 [ACK] Seq=2774181210 Ack=1520786085 Win=65535 Len=132
8	0.741120	173.1.1.1	11.101.1.1	TCP	192	1443 → 23496 [ACK] Seq=2774181210 Ack=1520786085 Win=65535 Len=132

In the packet capture, ESP packets are encapsulated into TCP ACK packets with the same sequence number. This is why anti-replay must be disabled on the NAT FortiGate.

IPsec key retrieval with a QKD system using the ETSI standardized API - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- IPsec key retrieval with a QKD system using the ETSI standardized API

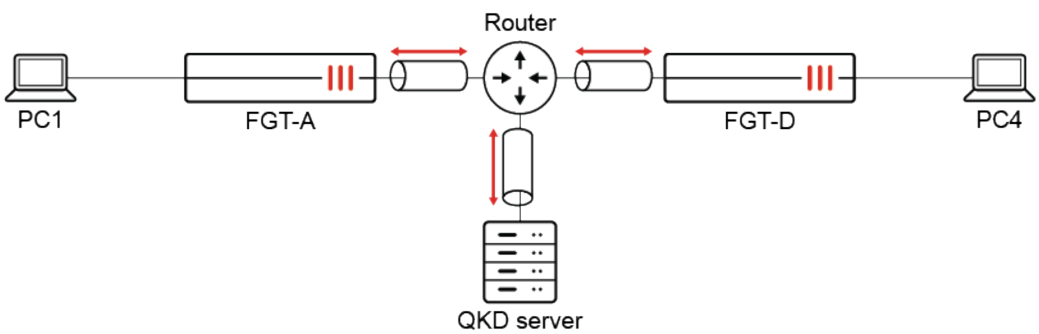
FortiGates support IPsec key retrieval with a quantum key distribution (QKD) system using the ETSI standardized API. This eliminates negotiation, simplifies the process, and enhances efficiency in IPsec key management.

```
config vpn qkd
    edit <name>
        set server <string>
        set port <integer>
        set id <string>
        set peer <string>
        set certificate <certificate_name>
    next
end
```

`server <string>`	Enter the IPv4, IPv6, or DNS address of the key management entity (KME).
`port <integer>`	Enter the port to connect to on the KME, 1 - 65535.
`id <string>`	Enter the quantum key distribution ID assigned by the KME.
`peer <string>`	Enter the peer or peer group to authenticate with the quantum key device's certificate.
`certificate <certificate_name>`	Enter the name of up to four certificates to offer to the KME.

Example

In this example, a quantum key distribution (QKD) system is deployed to perform central IPsec key management. The FortiGates installed as security gateways will terminate large amount of IPsec tunnels.



To configure IPsec key retrieval with a QKD system:

1. Configure FGT-A:
   a. Configure the QKD profile:

```
config vpn qkd
    edit "qkd_1"
        set server "172.16.200.83"
        set port 8989
        set id "FGT-A"
        set peer "qkd"
        set certificate "FGT_qkd1"
    next
end
```

   b. Configure the IPsec phase 1 interface settings:

```
config vpn ipsec phase1-interface
    edit "site1"
        set interface "wan1"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set qkd allow
        set qkd-profile "qkd_1"
        set remote-gw 173.1.1.1
        set psksecret **********
    next
end
```

c. Configure the IPsec phase 2 interface settings:

```
config vpn ipsec phase2-interface
    edit "site1"
        set phase1name "site1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
    next
end
```

2. Configure FGT-D:
   a. Configure the QKD profile:

```
config vpn qkd
    edit "qkd_1"
        set server "172.16.200.83"
        set port 8989
        set id "FGT-D"
        set peer "qkd"
        set certificate "FGT_qkd3"
    next
end
```

   b. Configure the IPsec phase 1 interface settings:

```
config vpn ipsec phase1-interface
    edit "site2"
        set interface "port25"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set qkd require
        set qkd-profile "qkd_1"
        set remote-gw 11.101.1.1
        set psksecret **********
    next
end
```

   c. Configure the IPsec phase 2 interface settings:

```
config vpn ipsec phase2-interface
    edit "site2"
        set phase1name "site2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
```

```
    next
end
```

To verify the configuration:

1. Generate traffic between PC1 and PC4.
2. Run diagnostics on FGT-A:
   a. Verify the IPsec phase 1 interface status:

```
# diagnose vpn ike gateway list

vd: root/0
name: site1
version: 1
interface: wan1 17
addr: 11.101.1.1:500 -> 173.1.1.1:500
tun_id: 172.16.200.4/::172.16.200.4
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 3s ago
peer-id: 173.1.1.1
peer-id-auth: no
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 30/30/30 ms

  id/spi: 21 ad7d995677250c7e/053f958ea7be66c8
  direction: initiator
  status: established 3-3s ago = 0ms
  proposal: aes128-sha256
  key: 5b198e1a431c20fb-c08135cf0c007704
  QKD: yes
  lifetime/rekey: 86400/86096
  DPD sent/recv: 00000000/00000000
  peer-id: 173.1.1.1
```

   b. Verify the IPsec phase 2 tunnel status:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=site1 ver=1 serial=2 11.101.1.1:0->173.1.1.1:0 tun_id=172.16.200.4 tun_
id6=::172.16.200.4 dst_mtu=1500 dpd-link=on weight=1
bound_if=17 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-
rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=12 olast=11 ad=/0
stat: rxp=1 txp=2 rxb=84 txb=168
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=site1 proto=0 sa=1 ref=3 serial=2
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=10226 type=00 soft=0 mtu=1438 expire=42883/0B replaywin=2048
       seqno=3 esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
```

```
  life: type=01 bytes=0/0 timeout=42897/43200
  dec: spi=b2af532f esp=aes key=16 c1d5d17e6bdecd5b145f672a5054cde1
       ah=sha1 key=20 084f1c0fee48994f59a125606f9c757838dc2421
  enc: spi=3d14392a esp=aes key=16 66277c8cf2bdbd2d12a9d829dde356ad
       ah=sha1 key=20 fdbaa42cca5c3a9bffb1cf0fc74ff29a643a2b9f
  dec:pkts/bytes=1/84, enc:pkts/bytes=2/304
  npu_flag=03 npu_rgwy=173.1.1.1 npu_lgwy=11.101.1.1 npu_selid=4 dec_npuid=2 enc_
npuid=2
```

The IPsec tunnel is up and traffic passes through.

c. Verify the IKE debug messages:

```
# diagnose debug application ike -1
...
ike V=root:0:site1:site1: IPsec SA connect 17 11.101.1.1->173.1.1.1:0
ike V=root:0:site1:site1: using existing connection
ike V=root:0:site1:site1: config found
ike V=root:0:site1:site1: IPsec SA connect 17 11.101.1.1->173.1.1.1:500 negotiating
ike 0:site1:20:site1:22: QKD initiator request
ike 0:site1:20:site1:22: QKD initiator key-id '4e0592fe-9568-11ee-97b8-5fb93000b0c2'
...
ike V=root:0:site1:20:site1:22: add IPsec SA: SPIs=b2af532d/3d143928
ike 0:site1:20:site1:22: IPsec SA dec spi b2af532d key
16:958EE561ABD2B6F0F4C6E042202F451E auth 20:4D694E6951ADB425A2A1C3261140957C9469A4DC
ike 0:site1:20:site1:22: IPsec SA enc spi 3d143928 key
16:6016E26398B70E55A17EF73611B30028 auth 20:357880E885F3ED23092233737B9FD0573DCB0D08
ike V=root:0:site1:20:site1:22: added IPsec SA: SPIs=b2af532d/3d143928
ike V=root:0:site1:20:site1:22: sending SNMP tunnel UP trap
```

d. Verify the statistics for qkd_1:

```
# diagnose vpn ike qkd qkd_1
client.count.fd: now 0  max 1  total 3
client.count.fp: now 0  max 1  total 3
client.count.mmap: now 2  max 2  total 9
client.event: 4
client.retry: 0
client.cmd.request.initiator: 4
client.cmd.request.responder: 0
client.cmd.reply.initiator: 4
client.cmd.reply.responder: 0
server.boot.count: 3
server.boot.last.time: 4295388395
server.boot.last.ago: 247
server.stop.budget: 0
server.stop.error: 0
server.stop.auth.count: 0
server.cmd.reading: 7
server.cmd.read: 4
server.cmd.request.initiator: 4
server.cmd.request.responder: 0
server.cmd.reply.initiator: 4
server.cmd.reply.responder: 0
server.auth.request.sending.count: 4
server.auth.request.sending.last.time: 4295389413
server.auth.request.sending.last.ago: 237
server.auth.request.sent.count: 4
```

```
server.auth.request.sent.last.time: 4295389413
server.auth.request.sent.last.ago: 237
server.auth.reply.reading.count: 4
server.auth.reply.reading.last.time: 4295389413
server.auth.reply.reading.last.ago: 237
server.auth.reply.read.count: 4
server.auth.reply.read.last.time: 4295389413
server.auth.reply.read.last.ago: 237
server.dns.addrs:
server.curl.get.count: 4
server.curl.get.last.time: 4295389413
server.curl.get.last.ago: 237
server.curl.json.parse: 4
server.curl.json.parsed: 4
```

Support for autoconnect to IPsec VPN using Microsoft Entra ID - 7.4.2

FortiOS now supports autoconnect to IPsec VPN using Microsoft Entra ID. This feature enables seamless and secure connectivity for users accessing corporate resources by automatically establishing IPsec VPN connections based on Microsoft Entra ID logon session information. See Support autoconnect to IPsec VPN using Entra ID logon session information for more information.

User and authentication

This section includes information about user and authentication related new features:

Authentication

This section includes information about authentication related new features:

Add RADSEC client support

This information is also available in the FortiOS 7.4 Administration Guide:

- Configuring a RADSEC client

FortiOS supports RADSEC clients in order to secure the communication channel over TLS for all RADIUS traffic, including RADIUS authentication and RADIUS accounting over port 2083. A FortiGate acting as a TLS client can initiate the TLS handshake with a remote RADIUS server. Administrators can specify a client certificate, perform a server identity check (enabled by default), and verify against a particular trust anchor (CA certificate). During a TLS handshake, the SNI check will use the RADIUS server FQDN if configured.

This enhancement also adds support for TCP connections, which use port 1812 for authentication and port 1813 for accounting.

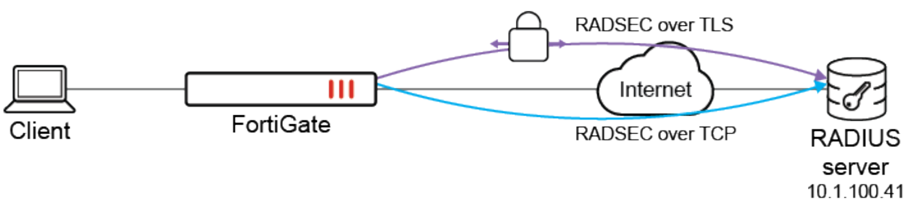
```
config user radius
    edit <name>
        set transport-protocol {udp | tcp | tls}
        set ca-cert <string>
        set client-cert <string>
        set tls-min-proto-version {default | SSLv3 | TLSv1 | TLSv1-1 | TLSv1-2}
        set server-identity-check {enable | disable}
    next
end
```

`transport-protocol {udp | tcp | tls}`	Set the type of transport protocol to use: • `udp`: use UDP (default) • `tcp`: use TCP, but no TLS security • `tls`: use TLS over TCP

`ca-cert <string>`	Set the CA certificate of server to trust under TLS.
`client-cert <string>`	Set the client certificate to use under TLS.
`tls-min-proto-version {default | SSLv3 | TLSv1 | TLSv1-1 | TLSv1-2}`	Set the minimum supported protocol version for TLS connections: • `default`: follow the system global setting • `SSLv3`: use SSLv3 • `TLSv1`: use TLSv1 • `TLSv1-1`: use TLSv1.1 • `TLSv1-2`: use TLSv1.2
`server-identity-check {enable | disable}`	Enable/disable RADIUS server identity check, which verifies the server domain name/IP address against the server certificate (default = enable).

Examples

The following topology is used to demonstrate configurations using RADSEC over TLS and RADSEC over TCP.



Example 1: RADSEC over TLS

When using TLS, FortiOS uses port 2083 for RADIUS authentication and RADIUS accounting. There is no need to configure the RADIUS accounting separately.

Before configuring RADSEC over TLS, make sure that the CA certificate (which issues the remote RADIUS server certificate) is imported into the FortiGate trusted root store. If a customized local FortiGate client certificate is used, both the certificate and private key are imported into local FortiGate certificate store.

To configure RADSEC over TLS:

1. Configure the RADIUS server:

```
config user radius
    edit "radius-tls"
        set server "10.1.100.41"
        set secret ********
        set acct-interim-interval 600
        set radius-port 2083
        set auth-type pap
        set transport-protocol tls
        set ca-cert "CA_Cert_2"
        set client-cert "portal.fortinet-fsso"
        config accounting-server
            edit 1
                set status enable
                set server "10.1.100.41"
```

```
                set secret ********
            next
        end
    next
end
```

2. Enable fnbamd debug messages on the FortiGate to verify the RADIUS authentication triggered by client traffic requesting access to external networks, which requires user authentication by the firewall policy. Note the highlighted initial RADSEC TLS authentication, successfully completed TLS handshake, and RADIUS accounting using TLS over port 2083:

```
# diagnose debug application fnbamd -1
Debug messages will be on for 30 minutes.

# diagnose debug enable
...
[629] __fnbamd_cfg_add_radius_by_user-
[1726] fnbamd_match_and_update_auth_user-Found a matching user in CMDB 'test1'
[462] fnbamd_rad_get-vfid=0, name='radius-tls'
[635] __fnbamd_cfg_add_radius_by_user-Loaded RADIUS server 'radius-tls' for user 'test1'
(16777236)
[905] fnbamd_cfg_get_radius_list-Total rad servers to try: 1
...
[806] __fnbamd_rad_get_next_addr-Next available address of rad 'radius-tls':
10.1.100.41:2083.
[981] __auth_ctx_start-Connection starts radius-tls:10.1.100.41, addr 10.1.100.41:2083
proto: TCP over TLS
[449] __rad_tcps_open-vfid 0, addr 10.1.100.41, src_ip (null), ssl_opt 1284
...
[618] create_auth_session-Total 1 server(s) to try
[1772] handle_req-r=4
[418] __rad_tcps_connect-tcps_connect(10.1.100.41) is established.
[716] __rad_rxtx-fd 10, state 1(Auth)
...
[565] fnbamd_rad_make_access_request-
[329] __create_access_request-Compose RADIUS request
[549] __create_access_request-Created RADIUS Access-Request. Len: 139.
...
[963] __auth_ctx_svr_push-Added addr 10.1.100.41:2083 from rad 'radius-tls'
[806] __fnbamd_rad_get_next_addr-Next available address of rad 'radius-tls':
10.1.100.41:2083.
[981] __auth_ctx_start-Connection starts radius-tls:10.1.100.41, addr 10.1.100.41:2083
proto: TCP over TLS
[449] __rad_tcps_open-vfid 0, addr 10.1.100.41, src_ip (null), ssl_opt 1284
[481] __rad_tcps_open-Server identity check is enabled.
[495] __rad_tcps_open-Still connecting 10.1.100.41.
...
[1393] create_acct_session-Acct type 6 session created, 0x9827960
[418] __rad_tcps_connect-tcps_connect(10.1.100.41) is established.
[716] __rad_rxtx-fd 10, state 4(Acct)
...
[956] fnbamd_rad_make_acct_request-
[905] __create_acct_request-Compose RADIUS request
[944] __create_acct_request-Created RADIUS Acct-Request. Len: 129.
[572] __rad_tcps_send-Sent 129/129.
[574] __rad_tcps_send-Sent all. Total 129.
```

```
[749] __rad_rxtx-Sent radius req to server 'radius-tls': fd=10, IP=10.1.100.41
(10.1.100.41:2083) code=4 id=33 len=123
[758] __rad_rxtx-Start rad conn timer.
...
```

Example 2: RADSEC over TCP

When using TCP, the default RADIUS ports remain same as with UDP: 1812 for authentication and 1813 for accounting.

To configure RADSEC over TCP:

1. Configure the RADIUS server:

```
config user radius
    edit "radius-tcp"
        set server "10.1.100.41"
        set secret ********
        set acct-interim-interval 600
        set transport-protocol tcp
        config accounting-server
            edit 1
                set status enable
                set server "10.1.100.41"
                set secret ********
            next
        end
    next
end
```

2. Enable fnbamd debug messages on the FortiGate to verify the RADIUS authentication triggered by client traffic requesting access to external networks, which requires user authentication by the firewall policy. Note the highlighted initial RADIUS authentication over TCP: 1812 and initial RADIUS accounting over TCP: 1813:

```
# diagnose debug application fnbamd -1
Debug messages will be on for 30 minutes.

# diagnose debug enable
...

[806] __fnbamd_rad_get_next_addr-Next available address of rad 'radius-tcp':
10.1.100.41:1812.
[981] __auth_ctx_start-Connection starts radius-tcp:10.1.100.41, addr 10.1.100.41:1812
proto: TCP
[449] __rad_tcps_open-vfid 0, addr 10.1.100.41, src_ip (null), ssl_opt 0
...
[1772] handle_req-r=4
[418] __rad_tcps_connect-tcps_connect(10.1.100.41) is established.
[716] __rad_rxtx-fd 10, state 1(Auth)
...
[565] fnbamd_rad_make_access_request-
[329] __create_access_request-Compose RADIUS request
[549] __create_access_request-Created RADIUS Access-Request. Len: 139.
[572] __rad_tcps_send-Sent 139/139.
[574] __rad_tcps_send-Sent all. Total 139.
[749] __rad_rxtx-Sent radius req to server 'radius-tcp': fd=10, IP=10.1.100.41
```

```
(10.1.100.41:1812) code=1 id=40 len=139
[758] __rad_rxtx-Start rad conn timer.
...
[806] __fnbamd_rad_get_next_addr-Next available address of rad 'radius-tcp':
10.1.100.41:1813.
[981] __auth_ctx_start-Connection starts radius-tcp:10.1.100.41, addr 10.1.100.41:1813
proto: TCP
[449] __rad_tcps_open-vfid 0, addr 10.1.100.41, src_ip (null), ssl_opt 0
...
[1393] create_acct_session-Acct type 6 session created, 0x982b280
[418] __rad_tcps_connect-tcps_connect(10.1.100.41) is established.
[716] __rad_rxtx-fd 10, state 4(Acct)
...
[574] __rad_tcps_send-Sent all. Total 129.
[749] __rad_rxtx-Sent radius req to server 'radius-tcp': fd=10, IP=10.1.100.41
(10.1.100.41:1813) code=4 id=41 len=123
[758] __rad_rxtx-Start rad conn timer.
...
```

Enable the FortiToken Cloud free trial directly from the FortiGate

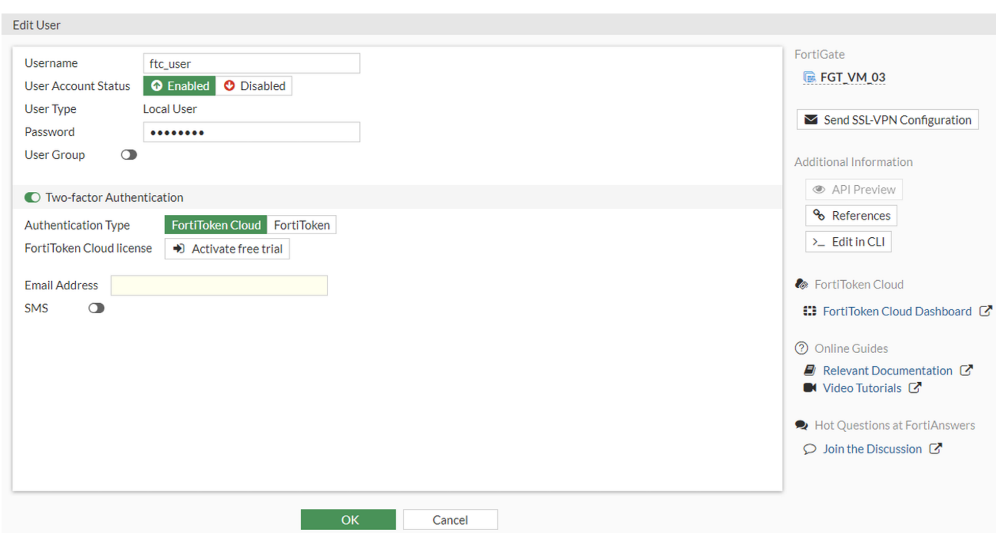
This information is also available in the FortiOS 7.4 Administration Guide:

- Enable the FortiToken Cloud free trial directly from the FortiGate

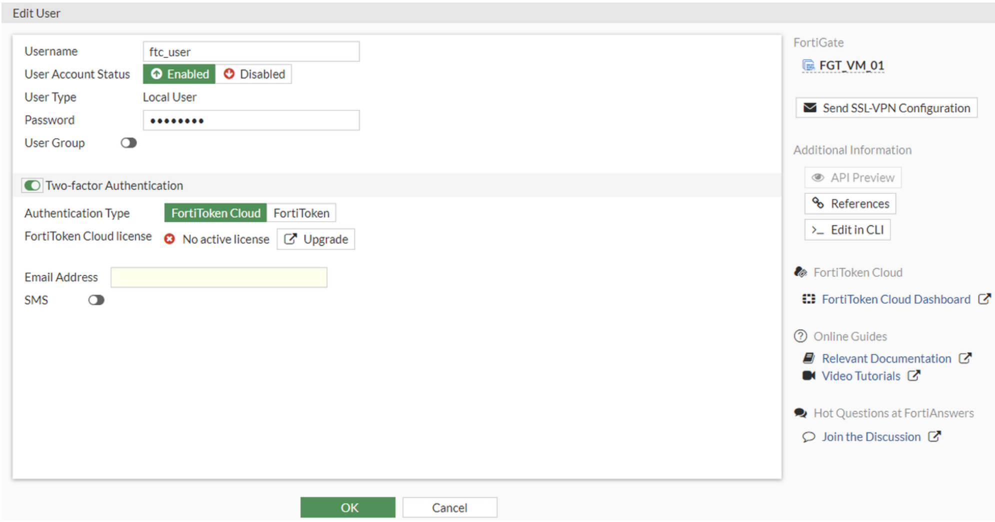
Administrators can activate a free one-month trial of FortiToken Cloud directly from the FortiGate instead of logging into the FortiCare Support Portal. This can be performed while enabling two-factor authentication within a user or administrator configuration, or from the *System > FortiGuard* page.

The FortiToken Cloud free trial can only be activated once and can only be activated if there is a registered FortiCare account. It cannot be activated if there is another FortiToken Cloud license or trial associated with the FortiGate device or the registered FortiCare accounts.

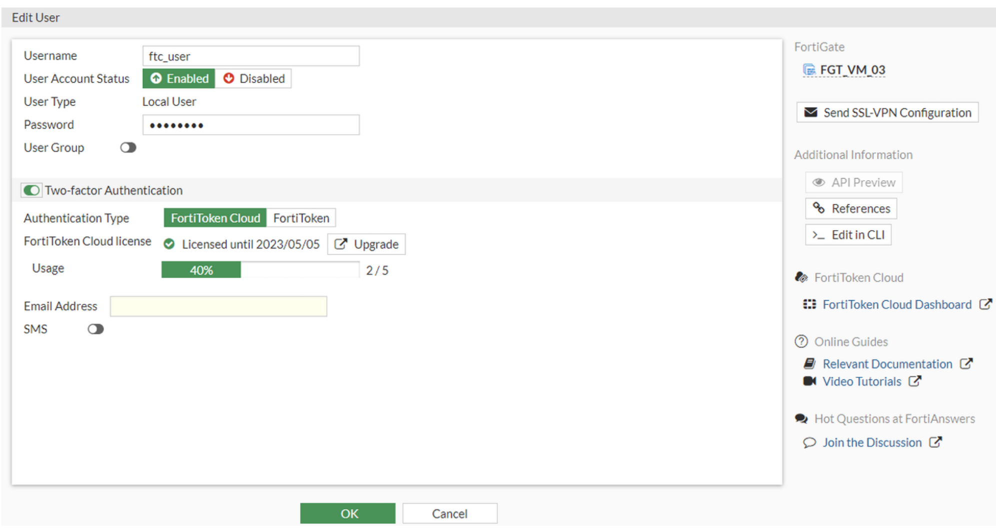
If the free trial has not been activated, the *Activate free trial* button will be available.



If the FortiToken Cloud license or free trial period is expired, the status will be displayed as *No active license*.

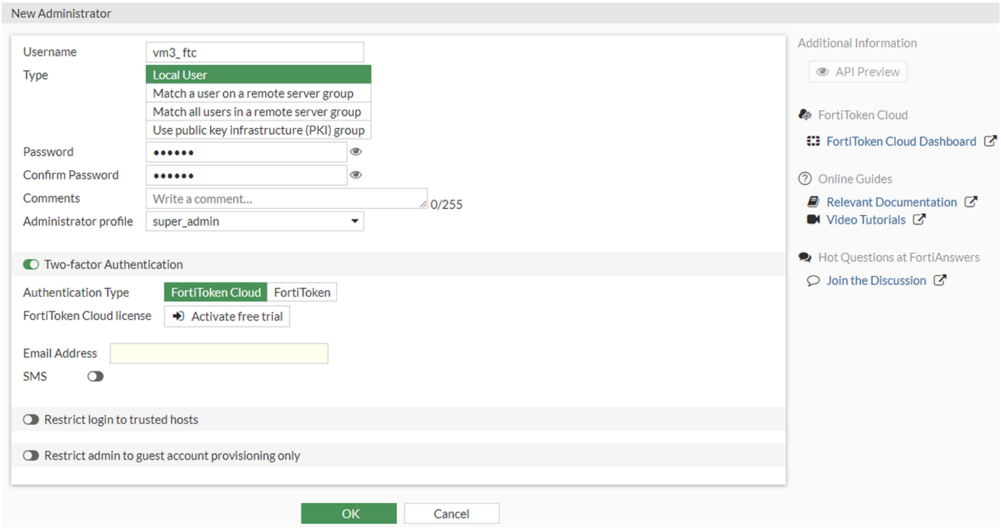


After activation, license information will be displayed and a *Usage* field will display how many of the available licenses have been assigned. Detailed usage information can be found using the CLI.

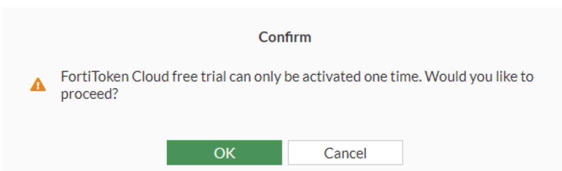


To enable the FortiToken Cloud free trial for an Administrator:

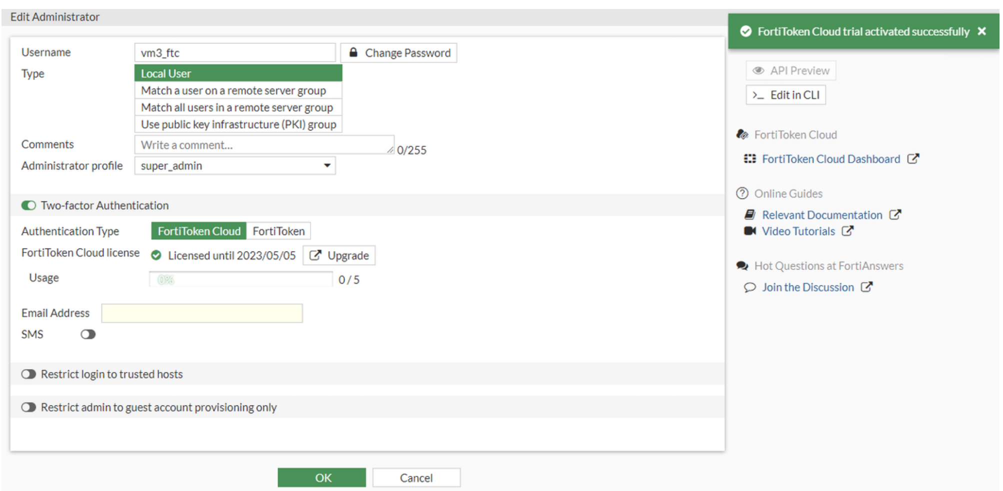
1. Go to *System > Administrators*.
2. Click *Create new > Administrator*.
3. Enable *Two-factor Authentication*.
4. Set *Authentication Type* to *FortiToken Cloud*.



5. Select *Activate free trial*. A confirmation message is displayed.



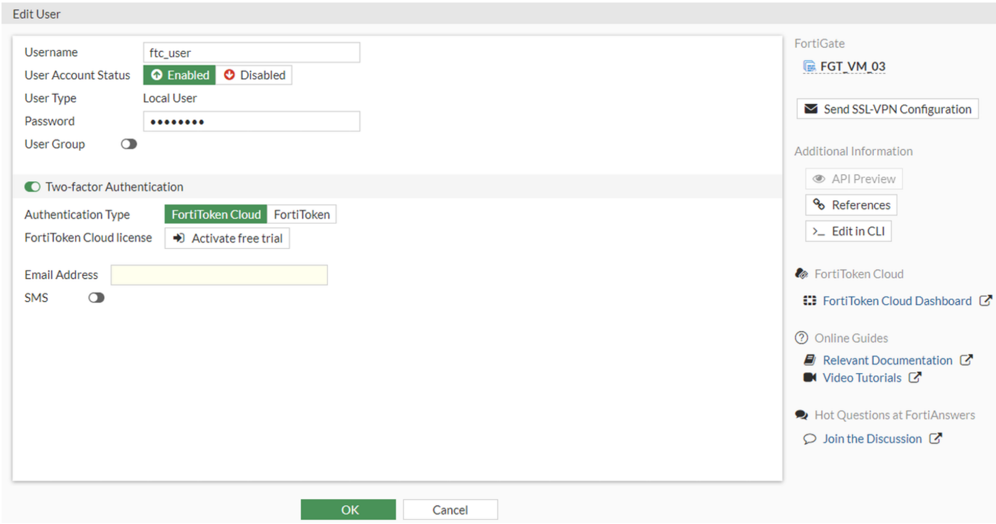
6. Click *OK*. The license information is displayed.



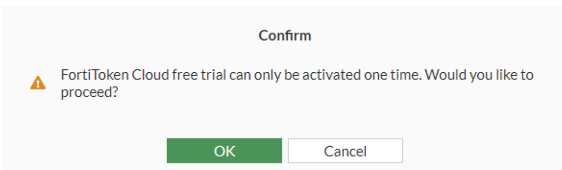
7. Click *OK*.

To enable the FortiToken Cloud free trial for a Local User:

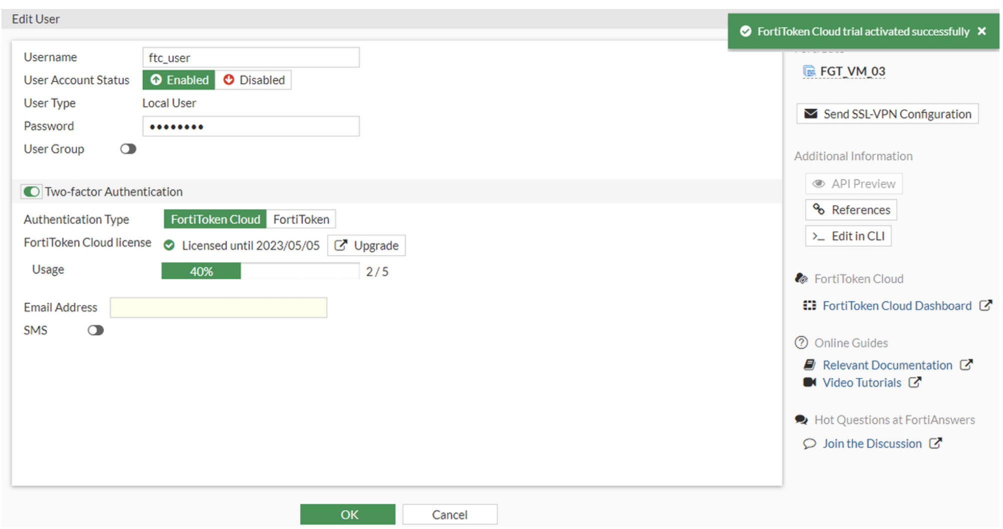
1. Go to *User & Authentication > User Definition*.
2. Click *Create new*.
3. Configure settings as needed.
4. Enable *Two-factor Authentication*.
5. Set *Authentication Type* to *FortiToken Cloud*.



6. Select *Activate free trial*. A confirmation message is displayed.



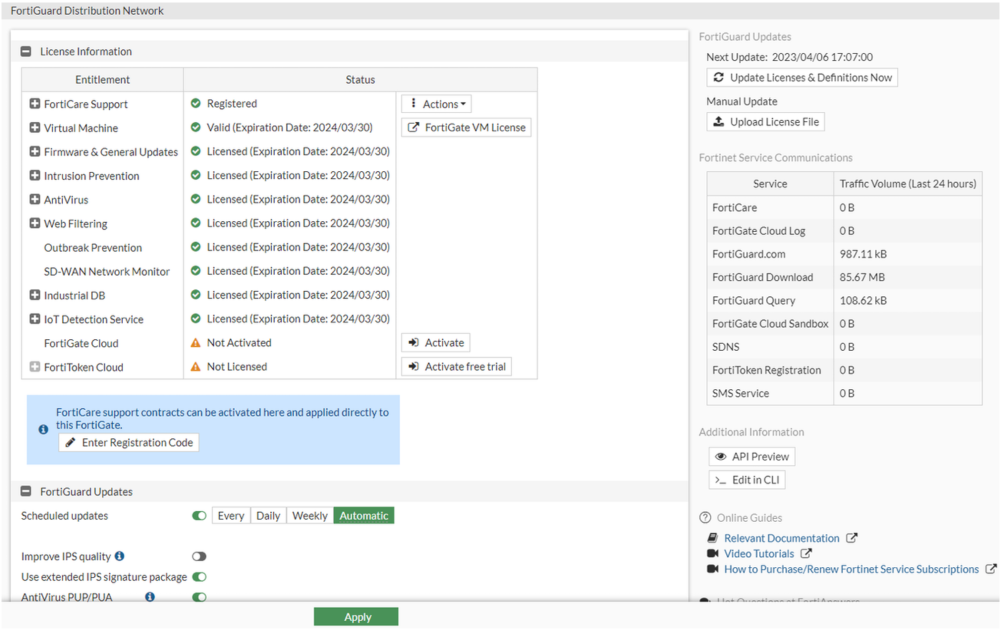
7. Click *OK*. The license information is displayed.



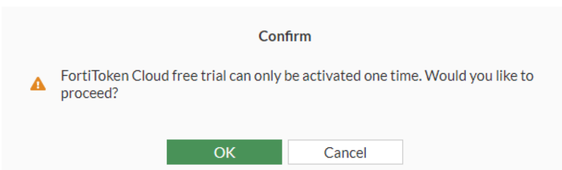
8. Click *OK*.

To enable the FortiToken Cloud free trial for FortiGuard:

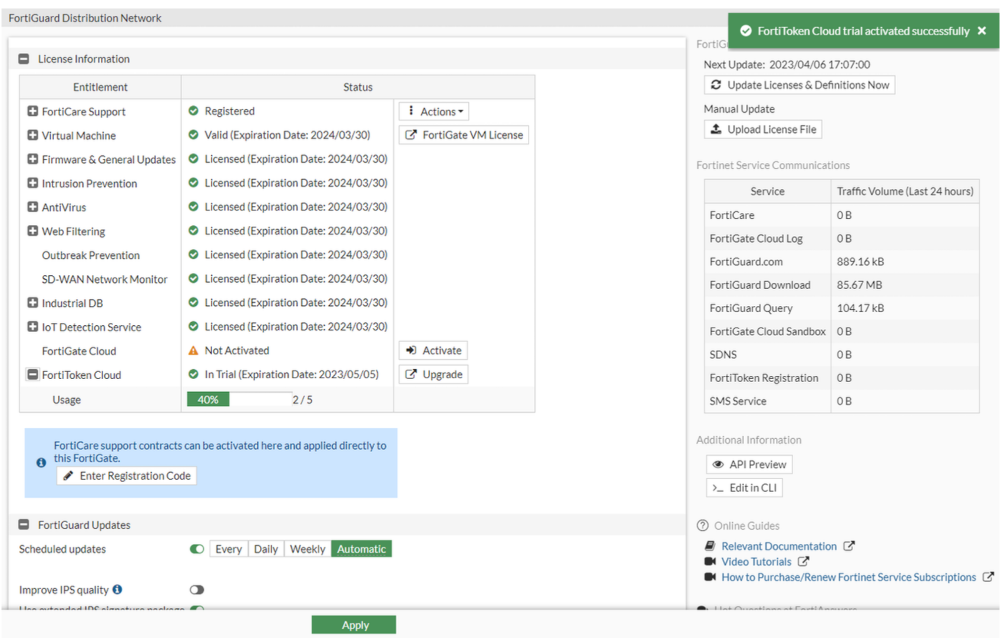
1. Go to *System > FortiGuard*.
2. Expand *License Information*.



3. Select *Activate free trial* for *FortiToken Cloud*. A confirmation message is displayed.



4. Click *OK*. The license information is displayed.



5. Click *Apply*.

To enable the FortiToken Cloud free trial in the CLI:

1. Activate the FortiToken Cloud trial:

```
# execute fortitoken-cloud trial
FortiToken Cloud free trial activated!
```

2. Review the status of the free trial:

```
# diagnose fortitoken-cloud show service
FortiToken Cloud service status: free trial.
Service balance: 0.00 users. Expiration date: 2022-07-06. Customer ID: 139XXXX.

# execute fortitoken-cloud show
FortiToken Cloud service status: free trial.
Service balance: 0.00 users. Expiration date: 2022-07-06. Customer ID: 139XXXX.
```

3. View users associated with FortiToken Cloud:

```
# diagnose fortitoken-cloud show users
Number of users in fortitoken cloud: 2
 1: username:vm3_ftc vdom:#FOS_Administrator email:fos@fortinet.com phone:
realm:FGTABCDXXXXXXXXX-#FOS_Administrator userdata:0
 2: username:ftc_user vdom:root email:fos@fortinet.com phone: realm:FGTABCDXXXXXXXXX-
root userdata:0
```

Enhance complexity options for local user password policy - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:
- Customizing complexity options for the local user password policy

The local firewall user password policy can be customized with various settings, such as minimum length, character types, and password reuse. These settings are similar to the ones available for the system administrator password policy, which offer more security and flexibility than the previous local user password policy.

```
config user password-policy
    edit <name>
        set minimum-length <integer>
        set min-lower-case-letter <integer>
        set min-upper-case-letter <integer>
        set min-non-alphanumeric <integer>
        set min-number <integer>
        set min-change-characters <integer>
        set expire-status {enable | disable}
        set reuse-password {enable | disable}
    next
end
```

`minimum-length <integer>`	Set the minimum password length (8 - 128, default = 8).
`min-lower-case-letter <integer>`	Set the minimum number of lowercase characters in the password (0 - 128, default = 0).

`min-upper-case-letter <integer>`	Set the minimum number of uppercase characters in the password (0 - 128, default = 0).
`min-non-alphanumeric <integer>`	Set the minimum number of non-alphanumeric in the password (0 - 128, default = 0).
`min-number <integer>`	Set the minimum number of numeric characters in the password (0 - 128, default = 0).
`min-change-characters <integer>`	Set the minimum number of unique characters in new password, which do not exist in the old password (0 - 128, default = 0). This attribute overrides `reuse-password` if both are enabled.
`set expire-status {enable | disable}`	Enable/disable password expiration (default = disable).
`set reuse-password {enable | disable}`	Enable/disable password reuse (default = enable. If both `reuse-password` and `min-change-characters` are enabled, `min-change-characters` overrides it.

After upgrading, users must activate the user password policy using the CLI. The previous password policy settings will remain valid, but they will not be effective unless the password policy password expiration is enabled (`expire-status`). If the password policy password expiration is not enabled, the `expire-days <integer>` option will not force users to change their password after number of specified days.

Example

The following user password policy is configured before upgrading:

```
config user password-policy
    edit "1"
        set expire-days 1
        set warn-days 1
        set expired-password-renewal enable
    next
end
```

To configure the user password policy options:

1. Check the user password policy settings after the upgrade:

```
config user password-policy
    edit 1
        get
            name                 : 1
            expire-days          : 1
            warn-days            : 1
            expired-password-renewal: enable
            minimum-length       : 8
            min-lower-case-letter: 0
            min-upper-case-letter: 0
            min-non-alphanumeric: 0
            min-number           : 0
            min-change-characters: 0
            expire-status        : disable
```

```
            reuse-password      : enable
    next
end
```

2. Edit the user password policy settings, including enabling password expiration:

```
config user password-policy
    edit "1"
        set expire-days 1
        set warn-days 1
        set expired-password-renewal enable
        set min-lower-case-letter 1
        set min-upper-case-letter 1
        set min-non-alphanumeric 3
        set min-number 3
        set min-change-characters 2
        set expire-status enable
        set reuse-password disable
    next
end
```

3. Change a password for a local user.
   a. In the CLI when the password meets the criteria:

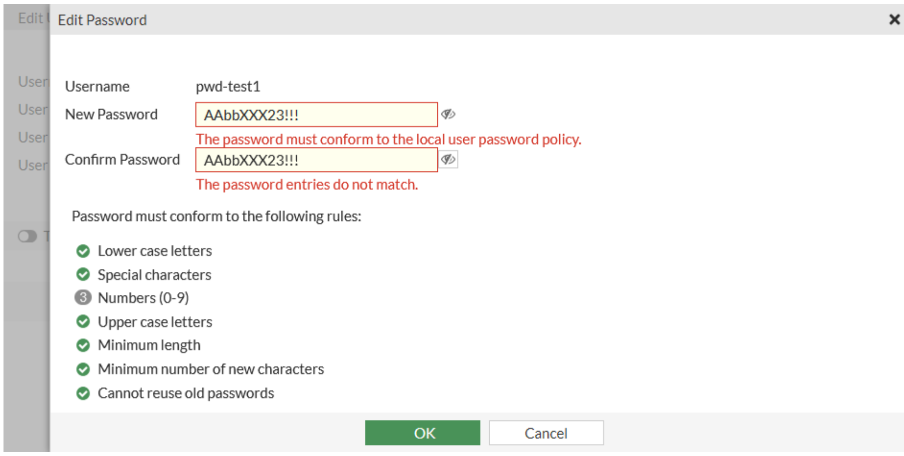
```
config user local
    edit pwd-test1
        set passwd CCbcset123!!!
    next
end
```

   b. In the CLI when the password does not meet the criteria (only two numbers, so an error message appears):

```
config user local
    edit pwd-test1
        set passwd CCbXsetp23!!!
New password must conform to the password policy enforced on this user:
Password must:
        Be a minimum length of 8
        Include at least 1 lower case letter(s) (a-z)
        Include at least 1 upper case letter(s) (A-Z)
        Include at least 3 non-alphanumeric character(s)
        Include at least 3 number(s) (0-9)
        Have at least 2 unique character(s) which don't exist in the old password
        Not be same as last two passwords

node_check_object fail! for passwd CCbXsetp23!!!

value parse error before 'CCbXsetp23!!!'
Command fail. Return code -49
```

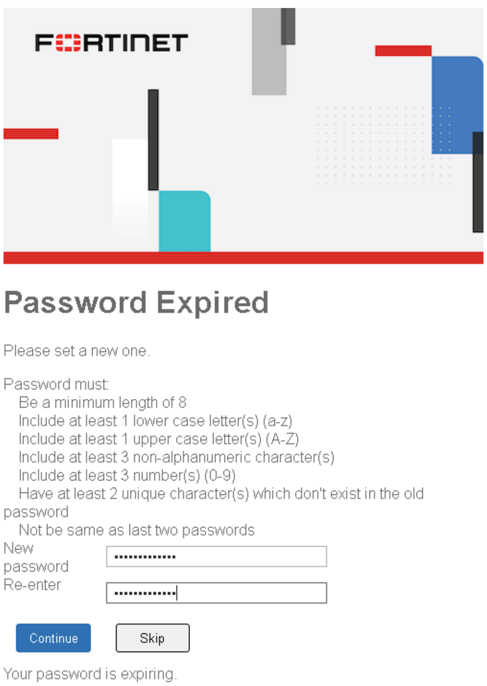
   c. In the GUI:
      i. Go to *User & Authentication* > *User Definition* and edit a local user.
      ii. Click *Change Password*.
      iii. Enter the *New Password*.
      iv. Enter the password again (*Confirm Password*). A warning will appear when the password does not match the criteria and indicates which parameters must be fixed. In this example, there are less than three

numbers used.

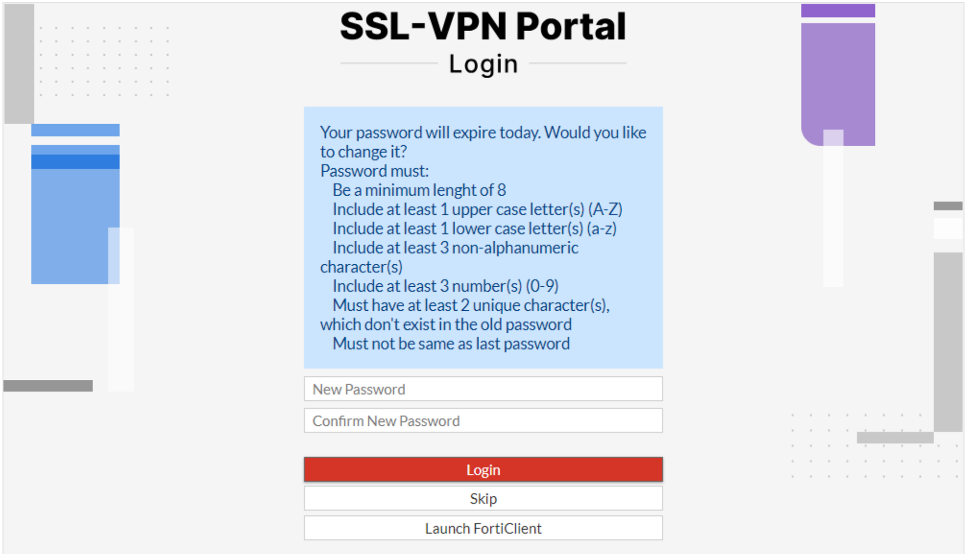


v. Click *OK*.

Sample prompt when a local user needs to update their password for firewall authentication:



Sample prompt when a local user needs to update their password for SSL VPN portal access:



RADIUS integrated certificate authentication for SSL VPN - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- RADIUS integrated certificate authentication for SSL VPN

Secure connections to SSL VPNs can be established using certificate-based authentication. Access can be granted to the user by using the content inside the Subject Alternative Name (SAN) of the user certificate to authenticate to the RADIUS server. An extra layer of security is added by ensuring that only users with valid certificates can access the VPN.

Certificate-based authentication with RADIUS supports UserPrincipalName (UPN), RFC 822 Name (corporate email address) defined in the SAN extension of the certificate, and the DNS defined in the user certificate as the unique identifier in the SAN field for peer user certificates.

```
config user radius
    edit <name>
        set account-key-processing {same | strip}
        set account-key-cert-field {othername | rfc822name | dnsname}
    next
end
```

`account-key-processing {same | strip}`	Account key processing operation. The FortiGate will keep either the whole domain or strip the domain from the subject identity. • `same`: Same as subject identity field (default). • `strip`: Strip domain string from subject identity field.

account-key-cert-field {othername | rfc822name | dnsname}	Define subject identity field in certificate for user access right checking. • othername: Match to UPN in SAN (default). • rfc822name: Match to RFC822 email address in SAN. • dnsname: Match to DNS name in SAN.

The RADIUS server configurations are applied to the user peer configuration when the PKI user is configured.

```
config user peer
    edit <name>
        set ca <string>
        set subject <string>
        set cn <string>
        set mfa-mode subject-identity
        set mfa-server <string>
    next
end
```

When a user authenticates to FortiGate over SSL VPN, the user presents a user certificate signed by a trusted CA to FortiGate. This CA should also be trusted by the FortiGate. See CA certificate for more information about importing a CA certificate to FortiGate trusted CA store. The following sequence of events occurs as the FortiGate processes the certificate for authentication:

1. The FortiGate checks whether the certificate is issued by a trusted CA. If the CA is not a public CA, FortiGate ensures that the CA certificate is uploaded and trusted by the FortiGate, and applies it to the user peer configurations (set ca <string>).
2. The FortiGate verifies that the CN field of the certificate matches the CN specified in the user peer configurations (set cn <string>).
3. If the user peer configuration has mfa-mode set to subject-identity and the mfa-server is configured, then the FortiGate uses the unique identifier in the certificate to authenticate against the RADIUS server.
   a. If account-key-cert-field is set to othername (the default setting), then the FortiGate uses the UPN in the certificate's SAN field to authenticate against RADIUS.
   b. If account-key-cert-field is set to rfc822name, then the FortiGate uses the RFC 822 Name in the certificate's SAN field to authenticate against RADIUS.
   c. If account-key-cert-field is set to dnsname, then the FortiGate uses the DNS name in the certificate to authenticate against RADIUS.

Some RADIUS servers do not require a password in an Access Request, while others need a valid password to return an ACCESS ACCEPT. If your RADIUS server requires a valid password to return an ACCESS ACCEPT, then you can configure an MFA password for each peer user using the set mfa-password command.

When you configure a user MFA password in a user peer, you must need to have a user peer configuration on the FortiGate for each user with cn=USER.

Example

In this example, a user certificate is issued to a user by a customer's CA. The certificate is used to authenticate the user to the SSL VPN web portal. The administrator uses the RFC 822 Name in the SAN field to authenticate against their corporate RADIUS. The Active Directory mail attribute is used to check against the RFC 822 Name field.

The configuration used in this example assumes the following:

- The CA certificate has already been uploaded to the FortiGate.
- SSL VPN has already been configured, pending the assignment of the PKI user group.

To configure the authentication settings:

1. Configure the RADIUS server:

```
config user radius
    edit "NPS-MFA"
        set server "172.18.60.214"
        set secret  XXXXXXXXXX
        set auth-type pap
        set password-encoding ISO-8859-1
        set account-key-processing strip
        set account-key-cert-field rfc822name
    next
end
```

2. Configure the local peer user:

```
config user peer
    edit "peer2"
        set ca "CA_Cert_1"
        set subject "L = Burnaby"
        set cn "test2"
        set mfa-mode subject-identity
        set mfa-server "NPS-MFA"
    next
end
```

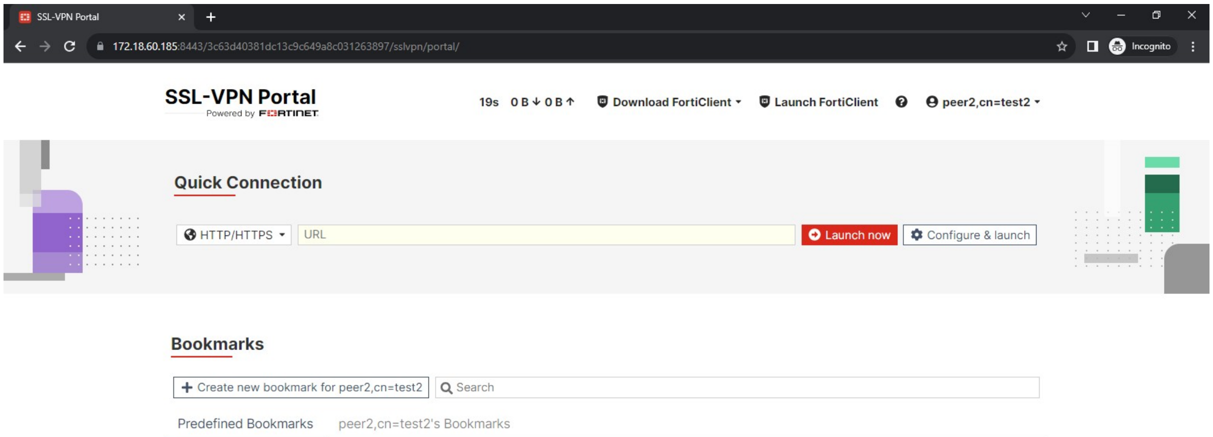
3. Configure the firewall user group for SSL VPN authentication:

```
config user group
    edit "sslvpn-mfa"
        set member  "peer2"
    next
end
```

4. Apply the user group to the SSL VPN configuration and firewall policy.

To verify the configuration:

When a user authenticates to Web mode SSL VPN using their browser, the FortiOS fnbamd daemon first validates the certificate supplied by the user. If the certificate check is successful, the information in the SAN field of the user certificate is used to find a matching user record on the RADIUS server. See SSL VPN web mode for information about configuring web mode SSL VPN.


SSL-VPN Portal
172.18.60.185:8443/3c63d40381dc13c9c649a8c031263897/sslvpn/portal/
Incognito
SSL-VPN Portal
Powered by FORTINET
19s 0 B 0 B
Download FortiClient
Launch FortiClient
peer2,cn=test2
Quick Connection
HTTP/HTTPS
URL
Launch now
Configure & launch
Bookmarks
Create new bookmark for peer2,cn=test2
Search
Predefined Bookmarks
peer2,cn=test2's Bookmarks

LAN Edge

This section includes information about LAN Edge related new features:

Wireless

This section includes information about wireless related new features:

Add profile support for UNII-4 5GHz band on FortiAP G-series models

This information is also available in the FortiWiFi and FortiAP 7.4 Configuration Guide:

- Configuring UNII-4 5GHz radio bands

FortiAP profiles support UNII-4 5GHz bands for FortiAP G-series models. FortiAP-431G and FortiAP-433G operating in Single 5G mode can make use of the UNII-4 frequency band. The 5.85 GHz-5.925 GHz channels of "169", "173", and "177" become available when configuring the 5GHz radio.

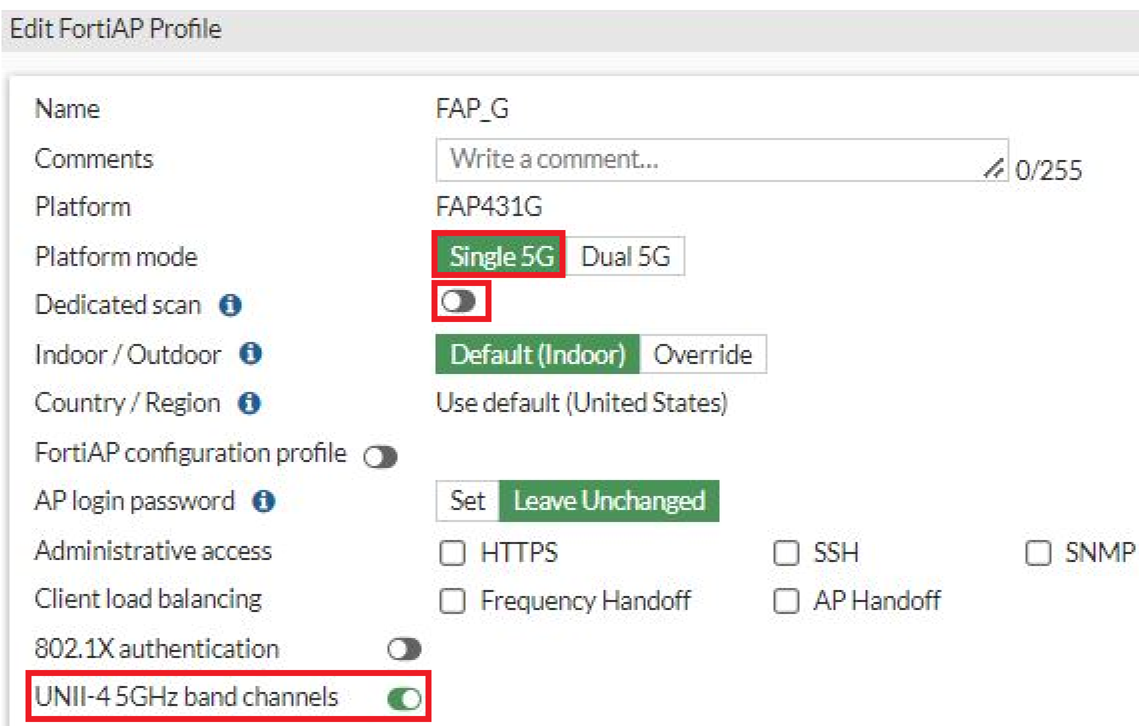
There are a few important points to note about UNII-4 band usage:

1. UNII-4 5GHz channels are not available when FAP43xG models operate in Dual 5G platform mode.
2. Not all countries allow UNII-4 band usage.
3. For APs operating in Single 5G platform mode, note the following behavior changes based on Dedicated scan:
    - When Dedicated scan is enabled, UNII-4 5 GHz channels are available by default. Radio 3 does not work in AP mode and Radio 2 can utilize all UNII-4 5GHz channels.
    - When Dedicated scan is disabled, you can choose to enable or disable UNII-4 5GHz.

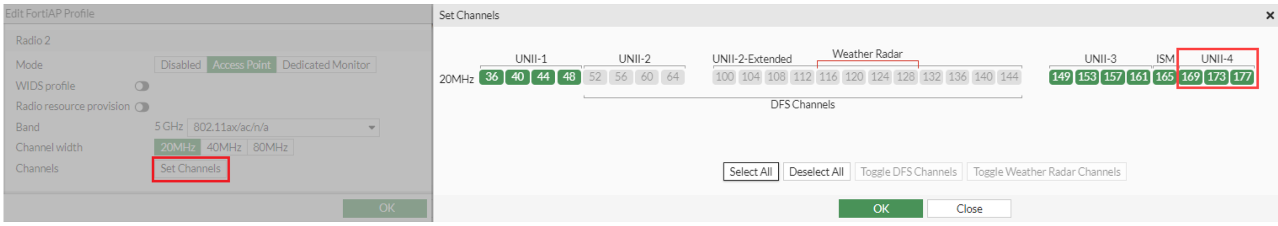
By default, FortiAP-431G and FortiAP-433G support UNII-4 5GHz channels when operating in Single 5G mode with Dedicated scan enabled; there is no need to configure anything. You can immediately select channels "169", "173", and "177" when configuring the 5GHz radio.

To configure UNII-4 5GHz band channels when the FortiAP is running in Single 5G mode with Dedicated scan disabled - GUI:

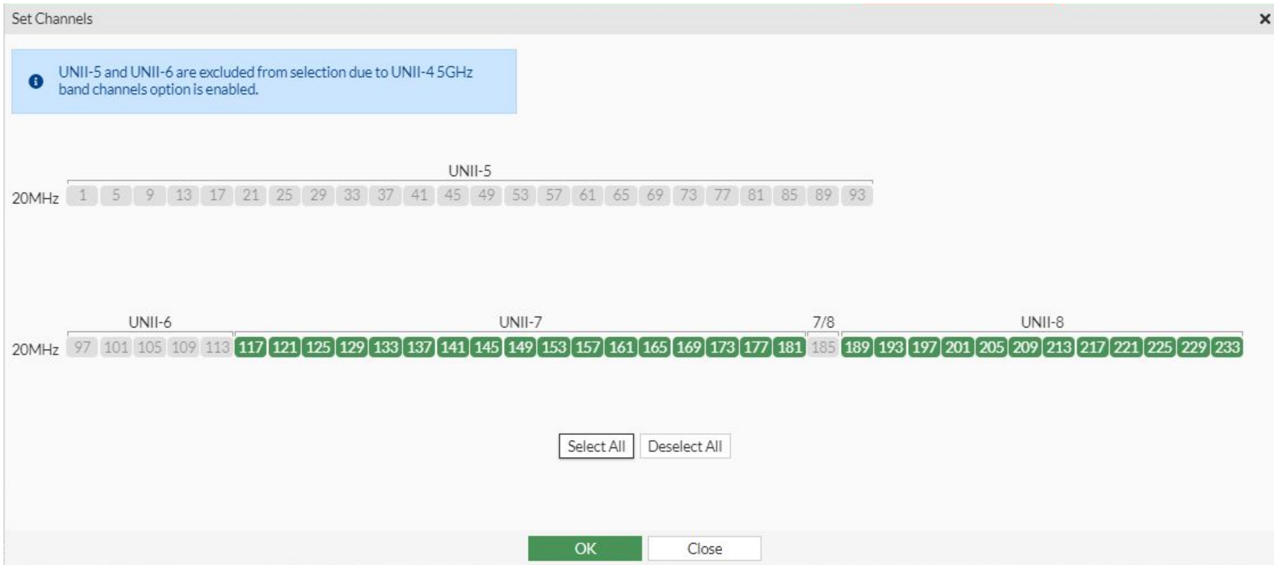
1. From the FortiGate GUI, navigate to *WiFi & Switch Controller > FortiAP Profiles*.
2. Select if you want to create a new profile or edit an existing FAP-43xG profile.
3. Set the *Platform mode* to *Single 5G.*
4. Disable *Dedicated scan*.
5. Enable *UNII-4 5GHz band channels*.



6. Under Radio 2, click *Set Channels* and select which channels you want to use.
   In the *Set Channels* window, you can see new channels "169", "173", and "177" under the UNII-4 category.



Note: Enabling UNII-4 5GHz band channels will cause the UNII-5 and UNII-6 Channels to be disabled on Radio 3.



To configure UNII-4 5GHz band channels when the FortiAP is running in Single 5G mode with Dedicated scan disabled - CLI:

1. When `DDSCAN` is disabled, you can configure the new `set unii-4-5ghz-band` command in FAP-431G or FAP-433G wtp-profiles.

```
config wireless-controller wtp-profile
  edit FAP_G
    config platform
      set  431G
    end
    set unii-4-5ghz-band ?
      enable     Enable UNII-4 5Ghz band channels.
      disable    Disable UNII-4 5Ghz band channels.
```

2. When you select enable, the following notification shows:

```
set unii-4-5ghz-band enable
  Enabling UNII-4 will reset radio-3 channel lists, UNII-5 and UNII-6 channels will be 
unavailable
  Do you want to continue? (y/n)
```

3. Enter `y` to continue. The UNII-4 5Ghz channels become available under `radio-2`.

```
config  radio-2
  set channel
  *wireless_channel    <36,40,44,48,149,153,157,161,165,169,173,177>
```

Note: Enabling UNII-4 5GHz band channels will cause the UNII-5 and UNII-6 Channels to be disabled on radio-3.

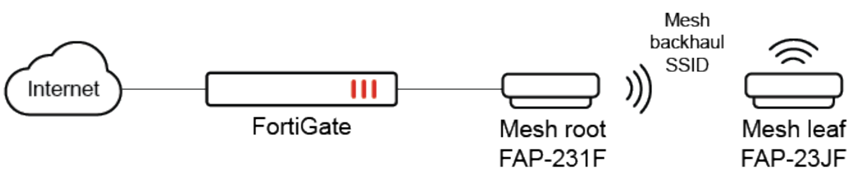
Add support for WPA3-SAE security mode on mesh backhaul SSIDs

This information is also available in the FortiWiFi and FortiAP 7.4 Configuration Guide:

- Configuring a meshed WiFi network

GUI support is available in FOS 7.4.1. For more information, see Add GUI support for configuring WPA3-SAE security mode on mesh backhaul SSIDs 7.4.1 on page 533.

This release supports configuring WPA3-SAE security mode for FortiAP wireless mesh backhaul SSIDs using the CLI. Wi-Fi 6E FortiAPs can also set up mesh connections over the 6GHz band as WPA3-SAE (with Hash-to-Element only enabled) is mandatory in Wi-Fi 6E technology.



In the topology example, FAP-231F is the mesh root that broadcasts the mesh backhaul SSID with WPA3-SAE security, and FAP-23JF is the mesh leaf that uses the mesh backhaul SSID to connect back to the FortiGate.

To configure WPA3-SAE security mode on mesh route SSIDs - CLI:

By default, `sae-h2e-only` is enabled when you set the security mode to `wpa3-sae`.

1. On the mesh root (backhaul) SSID, set the security mode to `wpa3-sae` and enable `mesh-backhaul`:

```
config wireless-controller vap
  edit "MESHWPA3"
    set mesh-backhaul enable
    set ssid "MESHWPA3"
    set security wpa3-sae
    set pmf enable
    set sae-h2e-only enable
    set schedule "always"
    set sae-password ENC *
  next
end
```

2. Add the mesh root SSID to the FortiAP profile:

```
config wireless-controller wtp-profile
  edit "FAP231F-default"
    config platform
      set type 231F
      set ddscan enable
    end
```

```
    set handoff-sta-thresh 55
    set allowaccess ssh
    config radio-1
      set band 802.11ax,n,g-only
      set vap-all manual
      set vaps "MESHWPA3"
    end
    config radio-2
      set band 802.11ax-5G
      set vap-all manual
      set vaps "MESHWPA3"
    end
    config radio-3
      set mode monitor
    end
  next
end
```

3. On the mesh leaf FortiAP, enable mesh leaf settings:

```
FortiAP-23JF# cfg -a MESH_AP_TYPE=1
FortiAP-23JF# cfg -a MESH_AP_SSID=MESHWPA3
FortiAP-23JF# cfg -a MESH_AP_PASSWD=fortinet
FortiAP-23JF# cfg -c
```

To verify FortiAP mesh configurations - CLI:

1. From the FortiGate, verify that mesh configurations have been successful applied:

- ```
  FortiGate-81E-POE (root) # diagnose wireless-controller wlac -c ws-mesh 0-
  11.11.11.3:5246
  -------------------------------WS MESH INFO    1----------------------------
  WTP session             : 0-11.11.11.3:5246  MP00   CWAS_RUN,91252 3,3
      Ctrl in_ifIdx       : 19/port11
           indev          : 19/port11
      Data in_ifIdx       : 19/port11
           indev          : 0/
      mesh uplink         : ethernet
      id                  : FP231FTF20000051
      mgmt_vlanid         : 0
      wtp_wanlan_mode     : wan-only
      refcnt              : 9
      deleted             : no
      plain_ctl           : disabled
      wtp-mode            : normal
      wtp-report-index    : 3
      data-chan-sec       : clear-text
      ctl-msg-offload     : ac=03ff/wtp_loc=03ff/wtp_rem=03ff/oper=03ff
      session_id          : 6fd0dc8e1431067779dee9796dc645ff
      ehapd cfg           : done
      message queue       : 0/128 max 14
      tId_10_sec          : 53777394
      Ekahau              : disabled
      Aeroscout           : disabled
      FortiPresence       : disabled
    Radio 1            : AP
  ```
  ```

```
        wlan cfg            : MESHWPA3
          vap-01(1)         : MESHWPA3         e0:23:ff:84:6a:b0   lsw m
    MESHWPA3         Config success State RUN(5) Age 91252
      Radio 2            : AP
        wlan cfg            : MESHWPA3
          vap-01(1)         : MESHWPA3         e0:23:ff:84:6a:b8   lsw m
    MESHWPA3         Config success State RUN(5) Age 91252
      Radio 3            : Monitor
      Radio 4            : Virtual Lan AP
        wlan cfg            :
      Radio 5            : Not Exist
    -------------------------------WS MESH INFO    2----------------------------
    WTP session             : 0-11.11.11.4:25246  MP00   CWAS_RUN,90789 7,7
        Ctrl in_ifIdx       : 19/port11
             indev          : 19/port11
        Data in_ifIdx       : 19/port11
             indev          : 0/
        mesh uplink         : mesh
             wbh sta        : 2 d4:76:a0:b1:48:ff
             wbh ap         : MESHWPA3 e0:23:ff:84:6a:b8 FP231FTF20000051
        id                  : FP23JFTF21000769
        mgmt_vlanid         : 0
        wtp_wanlan_mode     : wan-only
        refcnt              : 10
        deleted             : no
        plain_ctl           : disabled
        wtp-mode            : normal
        wtp-report-index    : 9
        data-chan-sec       : clear-text
        ctl-msg-offload     : ac=03ff/wtp_loc=03ff/wtp_rem=03ff/oper=03ff
        session_id          : 74d151af1d93fa801c5d55d6605441ba
        ehapd cfg           : ongoing
        message queue       : 0/128 max 91
        tId_10_sec          : 53777387
        Ekahau              : disabled
        Aeroscout           : disabled
        FortiPresence       : disabled
      Radio 1            : AP
    ...
```

- ```
  FortiGate-81E-POE (root) # diagnose wireless-controller wlac -d sta online
     vf=0 mpId=0 wtp=14 rId=2 wlan=MESHWPA3 vlan_id=0 ip=11.11.11.4 ip6=::
  mac=d4:76:a0:b1:48:ff vci=FortiAP-FP23JF host=FortiAP-23JF user= group= signal=-39
  noise=-95 idle=1 bw=58 use=5 chan=64 radio_type=11AX_5G security=wpa3_sae mpsk=
  encrypt=aes cp_authed=no l3r=1,0 G=0.0.0.0:0,0.0.0.0:0-0-0 -- 0.0.0.0:0 0,0
  online=yes mimo=2
  ```

2. From the FortiAP, verify the configuration on the FortiAP leaf:

```
FortiAP-23JF # cw_diag -c mesh
Sys Cfg AP addr mode: dhcp
           stp mode : 0
           dflt ip  : 192.168.1.2
           dflt mask: 255.255.255.0
           dflt gw  : 192.168.1.1
Mesh Cfg Uplink     : Mesh Uplink
         AP SSID    : MESHWPA3
```
```

```
              AP BSSID   : 00:00:00:00:00:00
              AP PASSWD  : ******
              wbh bgscan : 0
             ddscan ssid : MESHWPA3
        local eth bridge : 2(Disable)
     Mesh Oper AP Type   : Mesh Uplink
             wbh status  : running
             wbh rId     : 1
             wbh mac     : d4:76:a0:b1:48:ff
             wbh bssid   : e0:23:ff:84:6a:b8
             wbh Chan    : 144
             vap mhc     : 1
             eth type    : 0x2233
             bridge mac  : d4:76:a0:b1:48:e8
         main dhcp ip    : 11.11.11.4
         main dhcp mask  : 255.255.255.0
         main dhcp gw    : 11.11.11.11
           bh dhcp ip    : 0.0.0.0
           bh dhcp mask  : 0.0.0.0
           bh dhcp gw    : 0.0.0.0
            main ip      : 11.11.11.4
            main mask    : 255.255.255.0
            main gw      : 11.11.11.11
                bh ip    : 0.0.0.0
                bh mask  : 0.0.0.0
                bh gw    : 0.0.0.0
                bh mac   : 00:00:00:00:00:00
              eth bridge : 0(Disable)
```

Implement multi-processing for the wpad daemon for large-scale FortiAP management

This information is also available in the FortiWiFi and FortiAP 7.4 Configuration Guide:

- How to implement multi-processing for large-scale FortiAP management

This release adds the ability to configure multiple processors for the wireless daemon that handles WPA authentication requests (wpad_ac) by leveraging multi-core CPU to scale large numbers of FortiAP per FortiGate controller.

The new `wpad-process-count` allows users to configure multiple wpad_ac processes to handle WPA authentication requests. Users can set the `wpad-process-count` to a non-zero value such as 4, so the FortiGate will have four child wpad daemons where each process can handle a small group of SSIDs. The wpad daemon won't be as overloaded, and if one wpad daemon encounters an issue, it only affects that group of FortiAPs instead of all the FortiAPs managed by the FortiGate.

The `wpad-process-count` you can assign varies by FortiGate model and is based on the number of FortiAPs it is allowed to manage. The maximum value you can specify varies according to the `wireless-controller.wtp` in table size from different platforms.

wireless-controller.wtp	Maximum wpad-process-count
8192	32
4096	16
1024	8
256	4
16-64	2

To configure multiple wpad processes:

This example uses a FGT-101F that has a maximum `wpad-process-count` of 4.

1. Set the `wpad-process-count` under `wireless-controller global`:

   ```
   config wireless-controller global
     set wpad-process-count 4
   end
   ```

   Note that both `wpad_ac` and `cw_acd` processes are restarted when `wpad-process-count` is configured.

2. Verify the number of child wpad daemons created:

   ```
   # diagnose wpa wpad mp
       main process pid:       2221
       child process num:      4
               [1]:           2223
               [2]:           2225
               [3]:           2226
               [4]:           2227
   ```

3. Verify that VAPs with security modes of WPA-PSK, WPA-Enterprise, or radius-mac-auth are enabled and can be added to different wpad child daemons:

   ```
   # diagnose wpa wpad vap
   --------------------------   wpad[1]  -------------------------
   VAP number:    2
   VAP 0-10.10.24.20:35276-0-0 e0:22:ff:b2:19:30 state IDLE
       AC socket: /tmp/cwCwAcSocket_1
       Radius MAC Auth:0
       wpa version: WPA2
       preauth: 1
       ssid: FOS_101f.br1
       key_mgmt: WPA-PSK WPA-FT-PSK
       rsn_pairwise: CCMP
       rsn_group: CCMP
   VAP 0-10.10.24.20:35276-1-0 e0:22:ff:b2:19:38 state IDLE
       AC socket: /tmp/cwCwAcSocket_1
       Radius MAC Auth:0
       wpa version: WPA2
       preauth: 1
       ssid: FOS_101f.br.ent
       key_mgmt: WPA-EAP WPA-FT-EAP
       rsn_pairwise: CCMP
       rsn_group: CCMP
   ```

```
    auth: radius, server: wifi-radius
    Radius Auth NAS-IP: 0.0.0.0
    Radius Auth NAS-ID-TYPE: legacy
    Radius Auth NAS-ID: 10.10.24.20/35276-br2
VAP number: 2              Radius VAP number: 1
--------------------------   wpad[2]  -------------------------
There is no any WPA enabled VAP!
--------------------------   wpad[3]  -------------------------
VAP number:    3
VAP 0-10.6.30.254:25246-1-0 04:d5:90:b5:d7:e7 state IDLE
    AC socket: /tmp/cwCwAcSocket_3
    Radius MAC Auth:0
    wpa version: WPA2
    preauth: 1
    ssid: FOS_101f.ssid1
    key_mgmt: WPA-PSK
    rsn_pairwise: CCMP
    rsn_group: CCMP
VAP 0-10.6.30.254:5246-0-0 00:0c:e6:de:6f:31 state IDLE
    AC socket: /tmp/cwCwAcSocket_3
    Radius MAC Auth:0
    wpa version: WPA2
    preauth: 1
    ssid: FOS_101f.br1
    key_mgmt: WPA-PSK WPA-FT-PSK
    rsn_pairwise: CCMP
    rsn_group: CCMP
VAP 0-10.6.30.254:5246-1-0 00:0c:e6:de:6f:41 state IDLE
    AC socket: /tmp/cwCwAcSocket_3
    Radius MAC Auth:0
    wpa version: WPA2
    preauth: 1
    ssid: 101f.ssid.ent
    key_mgmt: WPA-EAP
    rsn_pairwise: CCMP
    rsn_group: CCMP
    auth: radius, server: wifi-radius
    Radius Auth NAS-IP: 0.0.0.0
    Radius Auth NAS-ID-TYPE: legacy
    Radius Auth NAS-ID: 10.5.30.252/5246-101f.ssid.ent
VAP number: 3              Radius VAP number: 1
--------------------------   wpad[4]  -------------------------
There is no any WPA enabled VAP!
```

4. Connect clients to the SSIDs and verify that each wpad child daemon can handle the authentication separately.

```
# diagnose wpa wpad sta
--------------------------   wpad[1]  -------------------------
VAP number:    2
    STA=48:ee:0c:23:43:d1,   state: PTKINITDONE
--------------------------   wpad[2]  -------------------------
There is no any WPA enabled VAP!
--------------------------   wpad[3]  -------------------------
VAP number:    3
    STA=f8:e4:e3:d8:5e:af,   state: PTKINITDONE
--------------------------   wpad[4]  -------------------------
There is no any WPA enabled VAP!
```

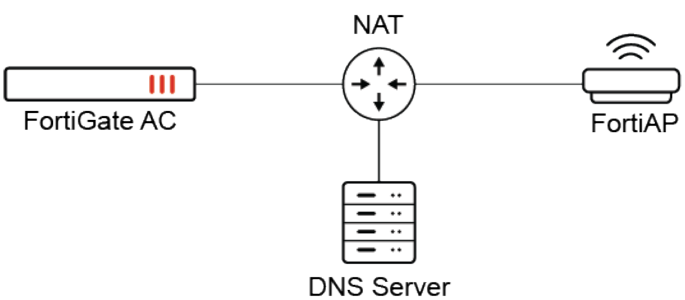
Add support for an IPsec VPN tunnel that carries the FortiAP SN

This information is also available in the FortiWiFi and FortiAP 7.4 Configuration Guide:

- Data channel security: clear-text, DTLS, and IPsec VPN

This release adds support for a new DTLS policy encryption policy, `ipsec-vpn-sn`. The `ipsec-vpn-sn` policy automatically establishes an IPsec VPN tunnel between the FortiGate and FortiAP that carries CAPWAP data packets and includes the FortiAP serial number within this tunnel.

```
config wireless-controller wtp-profile
    edit < profile_name >
        set dtls-policy ipsec-vpn-sn
    next
end
```



To encrypt the data channel with IPsec VPN that exposes the FortiAP SN - CLI:

1. From the FortiAP, configure the following:
   - AC_DISCOVERY_TYPE to `DNS`.
   - AC_HOSTNAME_1 with the provided FQDN.
   - AP_DATA_CHAN_SEC to `ipsec-sn`.

   ```
   FortiAP-231F # cfg -s
   AC_DISCOVERY_TYPE:=3
   AC_HOSTNAME_1:=portal1.fortigate.test
   AP_DATA_CHAN_SEC:=ipsec-sn
   ```

2. From the FortiGate, configure a wtp profile and enable `ipsec-sn-vpn` in the DTLS-policy setting.

   ```
   config wireless-controller wtp-profile
     edit FAP231F
       set dtls-policy ipsec-sn-vpn
     next
   end
   ```

3. Assign the wtp profile to a FortiAP.

   ```
   config wireless-controller wtp
     edit "FP231FTF20035672"
       set admin enable
       set wtp-profile "FAP231F"
     next
   end
   ```

4. The FortiAP starts to send ISAKMP packets to the FortiGate on port number 4500. Packets captured on the FortiGate shows the FortiAP serial number has been attached in clear-text format.

```
78.778713 10.1.99.254.64916 -> 10.1.99.103.4500: udp 399
0x0000   704c a599 40f9 00ff 0e0a 13fa 0800 4500        pL..@.........E.
0x0010   01ab 942e 4000 3f11 caac 0a01 63fe 0a01        ....@.?.....c...
0x0020   6367 fd94 1194 0197 1b85 0000 0000 ee35        cg.............5
0x0030   7924 2a8e e1a7 0000 0000 0000 0000 0110        y$*.............
0x0040   0400 0000 0000 0000 018b 0400 003c 0000        .............<..
0x0050   0001 0000 0001 0000 0030 0101 0001 0000        .........0......
0x0060   0028 0101 0000 800b 0001 000c 0004 0001        .(..............
0x0070   5180 8001 0007 800e 0100 8003 0001 8002        Q...............
0x0080   0004 8004 0014 0a00 0064 8aef 01b2 260a        .........d....&.
0x0090   0bd6 4a63 7364 af7b ebaa bdad 22d1 09dc        ..Jcsd.{...."...
0x00a0   43fc 92dd 9c31 4750 9897 ff2f d2fb b592        C....1GP.../....
0x00b0   5685 1eef 41d8 9417 c447 f080 e5f4 57e3        V...A....G....W.
0x00c0   f0eb 9a43 dd9d 6f76 8a36 cf3f f5b3 250b        ...C..ov.6.?..%.
0x00d0   7ddd d1bb 6e30 1217 bfe7 6c21 624b 9b10        }...n0....l!bK..
0x00e0   ac9e 71e5 d087 28f2 6a48 0500 0014 a6e9        ..q...(.jH......
0x00f0   4c31 c101 48e6 09a1 be35 58b1 3112 0d00        L1..H....5X.1...
0x0100   0013 0200 0000 776c 632d 3030 3032 2e30        ......wlc-0002.0
0x0110   300d 0000 144a 131c 8107 0358 455c 5728        0....J.....XE\W(
0x0120   f20e 9545 2f0d 0000 14cd 6046 4335 df21        ...E/.....`FC5.!
0x0130   f87c fdb2 fc68 b6a4 480d 0000 1490 cb80        .|...h..H.......
0x0140   913e bb69 6e08 6381 b5ec 427b 1f0d 0000        .>.in.c...B{....
0x0150   1444 8515 2d18 b6bb cd0b e8a8 4695 79dd        .D..-.......F.y.
0x0160   cc0d 0000 0c09 0026 89df d6b7 120d 0000        .......&........
0x0170   1412 f5f2 8c45 7168 a970 2d9f e274 cc01        .....Eqh.p-..t..
0x0180   000b 0000 14af cad7 1368 a1f1 c96b 8696        .........h...k..
0x0190   fc77 5701 0000 0000 2400 0000 0101 08f1        .wW.....$.......
0x01a0   02ee 3579 242a 8ee1 a746 5032 3331 4654        ..5y$*...FP231FT
0x01b0   4632 3030 3236 3437 32                         F20035672
```

5. From the FortiAP, verify that the connection is established with `ipsec-sn` data channel security.

```
FortiAP-231F # wcfg
WTP Configuration
< other output omitted >
    name                 : FortiAP-231F
    fsm-state            : RUN 69
    wtp-ip-addr          : 10.1.1.111:5246 - 169.254.0.2:57840
    ac-ip-addr           : 10.1.99.103:5246 - 169.254.0.1:5247        DNS
    data-chan-sec-cfg    : ipsec-sn
    data-chan-sec-oper   : ipsec-sn
```

6. From the FortiGate, verify the connection is established with `ipsec-sn-vpn` data channel security.

```
FortiGate-81E-POE # di wir wlac -c ws
-------------------------------WTP SESSION    4---------------------------
WTP session             : 0-10.1.99.254:5246-169.254.0.2:57840  MP00  CWAS_RUN,752 6,6
    Ctrl in_ifIdx       : 6/wan2
         indev          : 6/wan2
    Data in_ifIdx       : 42/wlc-0002.00
         indev          : 0/
    mesh uplink         : ethernet
    id                  : FP231FTF20035672
    mgmt_vlanid         : 0
    wtp_wanlan_mode     : wan-only
```

```
refcnt              : 9
deleted             : no
plain_ctl           : disabled
wtp-mode            : normal
wtp-report-index    : 4
data-chan-sec       : ipsec-sn-vpn
```

Support for WPA3 security modes on FortiWiFi units operating in Client Mode

GUI support is available in FOS 7.4.2. For more information, see GUI support for WPA3 security mode on Client mode FortiWiFi units 7.4.2 on page 540.

This release supports WPA3-SAE and OWE security modes on FortiWiFi units operating in wireless client mode. When the local radio of a FortiWiFi 8xF/6xF/40F model is operating in client mode, it can connect with third-party SSIDs with a WPA3-SAE or OWE security mode.

CLI changes

```
config wifi-networks
  edit < ID >
    set wifi-security [open | wpa-personal | wpa3-sae | owe]
  next
end
```

To configure WPA3 security mode SSID on a FortiWiFi running in client mode - CLI:

1. Change the wireless mode to client.

```
config system global
  set wireless-mode client
end
```

   Note: You must remove any AP WiFi configurations such as SSIDs, DHCP servers, policies, and software switch members before you can change the mode to Wireless Client. Once you select Wireless Client, the FortiWiFi unit will reboot.

2. Create a wireless network by connect to a third-party SSID and setting the security mode. In this example, the SSID is FOS_101F_WAP3_SAE and the security mode is WPA3 SAE.

```
config system interface
  edit "wifi"
    config wifi-networks
      edit 1
        set wifi-ssid "FOS_101F_WAP3_SAE"
        set wifi-security wpa3-sae
        set wifi-passphrase *
      next
    end
  next
end
```

To verify the connection status:

1. Verify the connection between the local radio and the third-party SSID with `diagnose wireless-controller wlsta cfg`.

```
diagnose wireless-controller wlsta cfg
STA intf        name: wlan17
              status: up
                  ip: 3.1.1.2
                 mac: d4:76:a0:18:e0:8f
        auto connect: yes
           auto save: no
             ap band: any
    wifi network cnt: 1
                   1: FOS_101F_WPA3_SAE, 19, 1
           connected: FOS_101F_WPA3_SAE
```

Support Dynamic VLAN assignment with multiple VLAN IDs per Name Tag - 7.4.1

This information is also available in the FortiWiFi and FortiAP 7.4 Configuration Guide:
- VLAN assignment by Name Tag

This enhancement expands the Support Dynamic VLAN assignment by Name Tag feature in FOS 7.0.4 where users could be dynamically assigned to VLANs based on a VLAN Name Table supported by a single VLAN ID. This update allows for multiple VLAN IDs to be configured per name tag, up to a maximum of 8 VLAN IDs. Once wireless clients connect to the SSID, the FortiGate wireless controller can assign the VLAN ID by a Round-robin method from the pool to ensure optimal utilization of VLAN resources.

To configure assigning multiple VLAN IDs per VLAN name tag:

1. Set up an SSID with `dynamic-vlan` enabled, and configure `vlan-name` with multiple entries under `vlan-id`:

```
config wireless-controller vap
  edit "wifi.fap.02"
    set ssid "Example_SSID"
    set security wpa2-only-enterprise
    set voice-enterprise disable
    set auth radius
    set radius-server "peap"
    set schedule "always"
    set dynamic-vlan enable
    config vlan-name
      edit "data"
        set vlan-id 100 200 300
      next
      edit "voip"
        set vlan-id 100
      next
    end
  next
end
```

VLAN Name	VLAN ID Pool
data	100, 200, 300 The new VLAN assignment method with multiple IDs per name.
voip	100 The previous VLAN assignment method with only one ID per name. Used as a comparison example.

2. Create user accounts in the Radius server with the `Tunnel-Private-Group-Id` matching the previously configured `vlan-name`.

```
data      Cleartext-Password := "123456"
          Tunnel-Type = "VLAN",
          Tunnel-Medium-Type = "IEEE-802",
          Tunnel-Private-Group-Id = data

voip      Cleartext-Password := "123456"
          Tunnel-Type = "VLAN",
          Tunnel-Medium-Type = "IEEE-802",
          Tunnel-Private-Group-Id = voip
```

To verify the clients connect and are assigned to the correct VLAN ID:

1. Connect four WiFi clients with `user=data` to verify that they can be assigned to the VLAN IDs from the VLAN Pool 100, 200, and 300 using a Round-robin method:
   a. Connect the first client and verify that it is assigned VLAN ID 100.

```
vf=2 mpId=6 wtp=2 rId=2 wlan=wifi.fap.02 vlan_id=100 ip=100.1.10.2 ip6=::
mac=00:0e:c9:9f:77:04 vci= host= user=data group= signal=-40 noise=-95 idle=25 bw=0
use=5 chan=48 radio_type=11N_5G security=wpa2_only_enterprise mpsk= encrypt=aes cp_
authed=no l3r=1,0 G=0.0.0.0:0,0.0.0.0:0-0-0 -- 0.0.0.0:0 0,0 online=yes mimo=2
```

   b. Connect the second client and verify that it is assigned VLAN ID 200.

```
vf=2 mpId=6 wtp=2 rId=2 wlan=wifi.fap.02 vlan_id=200 ip=100.2.10.2 ip6=::
mac=00:0e:ce:2d:e0:dd vci= host= user=data group= signal=-40 noise=-95 idle=0 bw=0
use=5 chan=48 radio_type=11N_5G security=wpa2_only_enterprise mpsk= encrypt=aes cp_
authed=no l3r=1,0 G=0.0.0.0:0,0.0.0.0:0-0-0 -- 0.0.0.0:0 0,0 online=yes mimo=2
```

   c. Connect the third client and verify that it is assigned VLAN ID 300.

```
vf=2 mpId=6 wtp=2 rId=2 wlan=wifi.fap.02 vlan_id=300 ip=100.3.10.2
ip6=fe80::20e:95ff:fef3:f124 mac=00:0e:95:f3:f1:24 vci= host= user=data group=peap
signal=-41 noise=-95 idle=0 bw=0 use=5 chan=48 radio_type=11N_5G security=wpa2_only_
enterprise mpsk= encrypt=aes cp_authed=no l3r=1,0 G=0.0.0.0:0,1.149.24.1:39198-0-0 --
0.0.0.0:0 0,0 online=yes mimo=2
```

   d. Connect the fourth client and verify that it is assigned VLAN ID 100 again.

```
vf=2 mpId=6 wtp=2 rId=2 wlan=wifi.fap.02 vlan_id=100 ip=100.1.10.3 ip6=::
mac=00:0e:44:9e:71:e5 vci= host= user=data group= signal=-40 noise=-95 idle=29 bw=0
use=5 chan=48 radio_type=11N_5G security=wpa2_only_enterprise mpsk= encrypt=aes cp_
authed=no l3r=1,0 G=0.0.0.0:0,0.0.0.0:0-0-0 -- 0.0.0.0:0 0,0 online=yes mimo=2
```

2. As a comparison, connect two WiFi clients stations with `user=voip`. They are assigned VLAN ID 100 as it matches the VLAN name "voip".

```
vf=2 mpId=6 wtp=2 rId=2 wlan=wifi.fap.02 vlan_id=100 ip=100.1.10.5
ip6=fe80::20e:5cff:fe03:e411 mac=00:0e:5c:03:e4:11 vci= host= user=voip group=peap
signal=-43 noise=-95 idle=14 bw=0 use=5 chan=48 radio_type=11N_5G security=wpa2_only_
enterprise mpsk= encrypt=aes cp_authed=no l3r=1,0 G=0.0.0.0:0,0.0.0.0:0-0-0 -- 0.0.0.0:0
0,0 online=yes mimo=2

vf=2 mpId=6 wtp=2 rId=2 wlan=wifi.fap.02 vlan_id=100 ip=100.1.10.4 ip6=::
mac=f8:e4:e3:d8:5e:af vci= host=WiFi-Client-2 user=voip group=peap signal=-39 noise=-95
idle=4 bw=0 use=5 chan=48 radio_type=11AX_5G security=wpa2_only_enterprise mpsk=
encrypt=aes cp_authed=no l3r=1,0 G=0.0.0.0:0,2.3.81.76:29193-0-0 -- 0.0.0.0:0 0,0
online=yes mimo=2
```

3. Check the VLAN assignment count using the following diagnostic command: `Diagnose wpa wpd vlan-name <SSID_NAME>`.

```
# diagnose wpa wpad vlan-name Example_SSID
No SSID is configured in hostapd.
No SSID is configured in hostapd.
SSID config: SSID(Example_SSID) VAP(wifi.fap.02) refcnt(2)
        Vlan info (1): v100.wifi => 100
        Vlan info (2): v200.wifi => 200
        Vlan info (3): v300.wifi => 300
        Vlan info (4): wqtn.50.wifi.fa => 4093
        Vlan info (5): data => 100(2) 200(1) 300(1)
        Vlan info (6): voip => 100(2)
```

Support for EAP/TLS on FortiWiFi models operating in Client Mode - 7.4.1

This information is also available in the FortiWiFi and FortiAP 7.4 Configuration Guide:

- Enabling EAP/TLS authentication on a FortiWiFi unit in client mode

EAP/TLS authentication is supported on FortiWiFi 80F/60F/40F series models operating in wireless client mode. This allows the FortiWiFi local radio to connect with a WPA2/WPA3-Enterprise SSID and support PEAP and EAP-TLS authentication methods.

This enhancements adds a new `wpa-enterprise` CLI option for the `wifi-security` setting under `wifi-network` configuration.

New CLI:

```
config wifi-networks
  edit < ID >
    set wifi-security wpa-enterprise
    set wifi-eap-type [both | tls | peap]
    set wifi-username < username >
    set wifi-client-certificate < client_cert_name >
    set wifi-private-key < client_cert_name >
  next
end
```

When `wifi-security` is set to `wpa-enterprise`, the local radio can recognize the security mode of third-party SSIDs and automatically adapt when connecting. These security modes include WPA2-Only-Enterprise, WPA3-Only-Enterprise, WPA3-Enterprise with 192-bit encryption, and etc.

When connecting to a WPA2/WPA3-Enterprise SSID via EAP-TLS, users must also configure the WiFi username, client certificate, private key settings, and etc as applicable.

To configure FortiWiFi to run in client mode and support EAP/TLS:

1. Change the wireless mode to client.

```
config system global
  set wireless-mode client
end
```

   Note: You must remove any AP WiFi configurations such as SSIDs, DHCP servers, policies, and software switch members before you can change the mode to Wireless Client. Once you select Wireless Client, the FortiWiFi unit will reboot.

2. Set the `wifi-security` mode to `wpa-enterprise`.

```
config system interface
  edit "wifi"
    config wifi-networks
      edit 1
        set wifi-ssid "FOS_101F_WPA2_ENT_PEAP"
        set wifi-security wpa-enterprise
        ...
```

3. After setting `wpa-enterprise`, configure the following as needed:

`wifi-eap-type`	Select a WPA2/WPA3-ENTERPRISE EAP method.
	• `PEAP` - `wifi-username` and wifi-passphrase should be set as the user account's name and password.
	• `TLS` - The client certificate should be specified by following settings:
	  ◦ `wifi-client-certificate`
	  ◦ `wifi-private-key`
	  ◦ `wifi-private-key-password:`

`wifi-username`	Username for WPA2/WPA3-ENTERPRISE.
`wifi-client-certificate`	Client certificate for WPA2/WPA3-ENTERPRISE.
`wifi-private-key`	Private key for WPA2/WPA3-ENTERPRISE.
`wifi-private-key-password`	Password for private key file for WPA2/WPA3-ENTERPRISE.
`wifi-ca-certificate`	CA certificate for WPA2/WPA3-ENTERPRISE.

Example Use Case - WPA2-Only-Enterprise SSID using the EAP-PEAP

The following example configures the local radio to connect to a WPA2-Only-Enterprise SSID using the EAP-PEAP authentication method.

1. Upload the CA certificate to verify the server certificate from the 3rd-party SSID.

   The CA certificate verification is an optional setting, users can decide whether to verify the server certificate by changing `wifi-ca-certificate` setting. To upload the CA certificate to FortiGate, log into the GUI and go to *System* > *Certificates*. Click *Create/Import* > *CA Certificate*, and follow the onscreen instructions to import the CA certificate.

2. Configure the `wifi-network` entry:

```
config system interface
  edit "wifi"
    config wifi-networks
      edit 1
        set wifi-ssid "FOS_101F_WPA2_ENT_PEAP"
        set wifi-security wpa-enterprise
        set wifi-eap-type peap
        set wifi-username "tester"
        set wifi-passphrase *
        set wifi-ca-certificate "CA_Cert_1"     <---This is an optional setting. "CA_
Cert_1" is the imported CA certificate
      next
    end
  next
end
```

3. Check the connection status:

```
FortiWiFi-81F-2R-POE # diagnose wireless-controller wlsta cfg
STA intf        name: wlan17
              status: up
                  ip: 10.4.1.2
                 mac: d4:76:a0:18:e0:8f
        auto connect: yes
           auto save: no
             ap band: any
    wifi network cnt: 1
                   1: FOS_101F_WPA2_ENT_PEAP, 16, 1
           connected: FOS_101F_WPA2_ENT_PEAP
```

Example Use Case - WPA3-Only-Enterprise SSID using EAP-TLS

The following example configures the local radio to connect to a WPA3-Only-Enterprise SSID using EAP-TLS authentication method.

1. Upload the CA certificate to verify the server certificate from the 3rd-party SSID.

   The CA certificate verification is an optional setting, users can decide whether to verify the server certificate by changing `wifi-ca-certificate` setting. To upload the CA certificate to FortiGate, log into the GUI and go to *System* > *Certificates*. Click *Create/Import* > *CA Certificate*, and follow the onscreen instructions to import the CA certificate.

2. Upload the client certificate (with private key file), which will be sent to the 3rd-party SSID side for verification and authentication.

a. To upload the client certificate with private key file to FortiGate, log into the GUI and go to *System > Certificates*.
b. Click *Create/Import > Certificate*
c. Click *Import Certificate*, select *PKCS #12 Certificate* or *Certificate,* and then follow the onscreen instructions to import the client certificate with private key file.

3. Configure the `wifi-network` entry:

```
config system interface
  edit "wifi"
    config wifi-networks
      edit 2
        set wifi-ssid "FOS_101F_WPA3_ENT_TLS"
        set wifi-security wpa-enterprise
        set wifi-eap-type tls
        set wifi-username "81F-client"
        set wifi-client-certificate "client-cert"     <----"client-cert" is the name of 
imported client certificate
        set wifi-private-key "client-cert"          <---It uses the same name of 
imported client certificate
        set wifi-private-key-password *
        set wifi-ca-certificate "CA_Cert_1"     <---This is an optional setting. "CA_
Cert_1" is the imported CA certificate
      next
    end
  next
end
```

- `wifi-username` is the "identity" of the client-mode local radio during EAP-TLS authentication.
- `wifi-private-key-password` is the password created when importing the client certificate on the FortiWiFi.

4. Check the connection status:

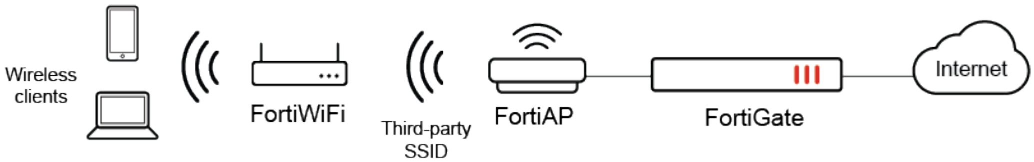
```
FortiWiFi-81F-2R-POE # diagnose wireless-controller wlsta cfg
STA intf         name: wlan07
               status: up
                   ip: 10.30.80.2
                  mac: d4:76:a0:18:e0:87
         auto connect: yes
            auto save: no
              ap band: any
     wifi network cnt: 1
                    1: FOS_101F_WPA3_ENT_TLS, 16, 1
            connected: FOS_101F_WPA3_ENT_TLS
```

Enable AP and Client mode on FortiWiFi 80F series models - 7.4.1

This information is also available in the FortiWiFi and FortiAP 7.4 Configuration Guide:

- Configuring a FortiWiFi unit to run in concurrent AP and wireless client mode

This enhancement supports concurrent AP and Client mode on FortiWiFi 80F/81F-2R-XX models. When the FortiWiFi is configured to run in wireless client mode and the FortiWiFi local radio connects to a third-party SSID, the local radio can concurrently operate in AP mode to provide service to wireless clients. Since the FortiWiFi can have VAP and SSID interfaces configured on the local radio profile, connected clients can then access wired and wireless resources through the FortiWiFi firewall policies.



To configure a FortiWiFi 80F series model to run in AP and Client Mode - CLI:

1. Configure the FortiWiFi unit to operate in client mode.

```
config system global
  set wireless-mode client
end
```

2. Connect to a third-party SSID, in this example FOS_101F_psk.

```
config system interface
  edit "wifi"
      config wifi-networks
        edit 1
          set wifi-ssid "FOS_101F_psk"
          set wifi-passphrase *
        next
      end
  next
end
```

Optionally, you can configure the wireless client to use a static IP or DHCP by modifying the addressing mode of the WiFi interface:

```
config system interface
  edit "wifi"
    set vdom "root"
    set mode static # For static IP. Use “set mode dhcp” for DHCP
    set ip 10.20.80.3 255.255.255.0 # For static IP only
    set allowaccess ping fabric
    set type wireless
    config wifi-networks
      edit 1
        set wifi-ssid "FOS_101F_psk"
        set wifi-passphrase *
      next
    end
  next
end
```

3. Verify the connection between the local radio and the third-party SSID with `diag wireless-controller wlsta cfg`.

```
FortiWiFi-81F-2R-POE # diagnose wireless-controller wlsta cfg
STA intf        name: wlan17
              status: up
                  ip: 192.168.81.2
                 mac: d4:76:a0:18:e0:8f
        auto connect: yes
           auto save: no
             ap band: any
    wifi network cnt: 1
                   1: FOS_101F_psk, 8, 1
           connected: FOS_101F_ psk
```

4. Verify the local radio status when working in AP mode with `diag wireless-controller wlac -c wtp`.

```
FortiWiFi-81F-2R-POE # diagnose wireless-controller wlac -c wtp FW81FP-WIFI0 | grep
connection
connection state : Connected
```

To configure VAP and SSID interfaces on the FortiWiFi local radio profile - CLI:

By default, the FortiWiFi local radio has a FWF-default profile; no other profiles can be applied to the local radio. You can modify the band, channel, and SSID selections in the FWF-default profile to apply to the local radio. Wireless clients that connect to the local radio are subject to the FortiWiFi firewall policies.

1. Create a new VAP interface and select it in the FWF-default profile.

```
config wireless-controller vap
  edit "wifi1"
    set ssid "FOS_lab_psk"
    set passphrase *
  next
end
config wireless-controller wtp-profile
  edit "FWF-default"
    config radio-1
      set vap-all manual
      set vaps "wifi1"
    end
    config radio-2
      set vap-all manual
      set vaps "wifi1"
    end
  next
end
```

The local radio applies the profile setting when broadcasting SSIDs.

2. Verify that these settings are applied with `diag wireless-controller wlac -c wtp`.

```
FortiWiFi-81F-2R-POE # diagnose wireless-controller wlac -c wtp
-------------------------------WTP    1---------------------------
WTP vd               : root, 0-FW81FP-WIFI0    MP00
    uuid             : 4b7c0b96-1ce9-51ee-7547-14eeab836b46
    mgmt_vlanid      : 0
    region code      : A  valid
```

```
    refcnt            : 3 own(1) wtpprof(1) ws(1)   deleted(no)
    apcfg status      : N/A,N/A cfg_ac=0.0.0.0:0 val_ac=0.0.0.0:0 cmds T 0 P 0 U 0 I 0 M
0
    apcfg cmd details:
    plain_ctl         : disabled
    image-dl(wtp,rst): yes,no
    admin             : enable
    wtp-profile       : cfg(FWF-default) override(disabled) oper(FWF-default)
    wtp-mode          : normal
    wtp-wanlan-mode   : aggregate
...
    split-tunneling-acl-path         : local
    split-tunneling-local-ap-subnet  : disabled
    active sw ver     : FP231F-v7.2-build5354
    local IPv4 addr   : 192.168.80.2
    board mac         : d4:76:a0:18:e0:78
    join_time         : Fri Jul  7 10:32:21 2023
    mesh-uplink       : ethernet
    mesh hop count    : 0
    parent wtp id     :
    connection state : Connected
....
  Radio 1             : AP
    80211d enable:    : enabled
    country name      : US
    country code      : 841
    drma_manual_mode : ncf
    radio_type        : 11AX
    channel list      : 1 6 11
    darrp             : disabled
    airtime fairness : disabled
    bss color mode    : Auto
    bss color(actual): 36
    txpower           : 100% (calc 27 oper 27 max 27 dBm)
    beacon_intv       : 100
    rts_threshold     : 2346
    frag_threshold    : 2346
    ap scan           : disable
    ap scan passive   : disabled
    sensor mode       : disabled
    ARRP profile      : ---
    WIDS profile      : ---
      wlan  0         : wifi1
    max vaps          : 8
    base bssid        : d4:76:a0:18:e0:80
    oper chan         : 11
    noise_floor       : -95
    chutil            : enabled
    oper chutil time : Fri Jul  7 14:26:28 2023  (age=7)
    oper chutil data : 88,89,90,88,86, 89,89,87,90,88, 86,88,88,88,85 ->newer
    station info      : 0/0
  Radio 2             : AP
    80211d enable:    : enabled
    country name      : US
    country code      : 841
    drma_manual_mode : ncf
```

```
    radio_type       : 11AX_5G
    channel list     : 36 40 44 48 52 56 60 64 100 104 108 112 116 120 124 128 132 ...
    darrp            : disabled
    airtime fairness : disabled
    bss color mode   : Auto
    bss color(actual): 61
    txpower          : 100% (calc 27 oper 25 max 27 dBm)
    beacon_intv      : 100
    rts_threshold    : 2346
    frag_threshold   : 2346
    ap scan          : disable
    ap scan passive  : disabled
    sensor mode      : disabled
    ARRP profile     : ---
    WIDS profile     : ---
      wlan  0        : wifi1
    max vaps         : 8
    base bssid       : d4:76:a0:18:e0:88
    oper chan        : 108
    noise_floor      : -95
    chutil           : enabled
    oper chutil time : Fri Jul  7 14:26:28 2023  (age=7)
    oper chutil data : 7,7,7,10,5, 9,10,10,14,13, 15,12,10,18,13 ->newer
    station info     : 0/0
  Radio 3            : Monitor
…..
```

3. Create a firewall policy from "wifi1" to the "aplink" interface to allow wireless clients to pass traffic from the unit.

```
config firewall policy
  edit 1
    set name "wifi1"
    set uuid e0140546-1d0d-51ee-da6c-53fb724051ac
    set srcintf "wifi1"
    set dstintf "aplink"
    set action accept
    set srcaddr "all"
    set dstaddr "all"
    set schedule "always"
    set service "ALL"
    set nat enable
  next
end
```

4. Connect a wireless client through the local radio of the FortiWiFi and verify that it has the correct IP and can pass traffic to the Internet.

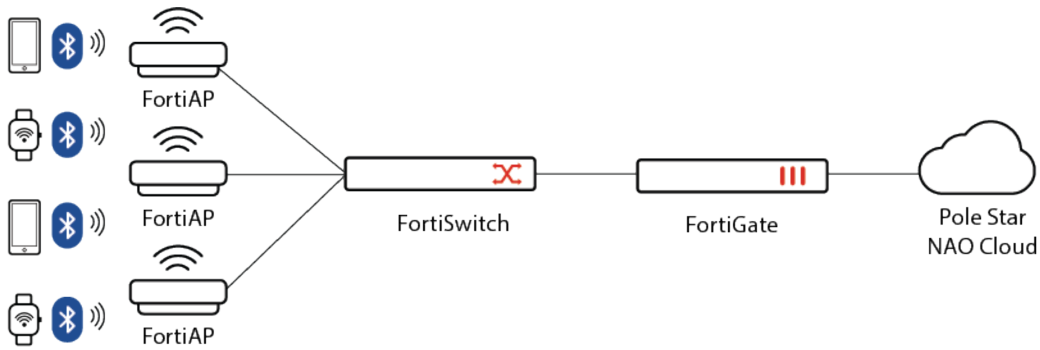
```
FortiWiFi-81F-2R-POE # diagnose wireless-controller wlac -d sta online
   vf=0 mpId=0 wtp=1 rId=2 wlan=wifi1 vlan_id=0 ip=10.10.80.2 ip6=::
mac=f8:e4:e3:d8:5e:af vci= host=WiFi-Client-2 user= group= signal=-45 noise=-95 idle=0
bw=0 use=5 chan=108 radio_type=11AX_5G security=wpa2_only_personal mpsk= encrypt=aes cp_
authed=no l3r=1,0 G=0.0.0.0:0,0.0.0.0:0-0-0 -- 0.0.0.0:0 0,0 online=yes mimo=2
```

Integration with Pole Star's NAO Cloud service for BLE asset tag tracking - 7.4.1

This information is also available in the FortiWiFi and FortiAP 7.4 Configuration Guide:

- Pole Star NAO Cloud service integration

This release adds integration with Pole Star's NAO Cloud service by supporting Pole Star's BLE asset tags and forwarding their data to the cloud service. Managed FortiAP units can be configured to scan Pole Star BLE asset tags and send the scanned data to the Pole Star’s NAO Cloud. This enables wearable devices with BLE asset tags to communicate with FortiAPs via their built-in Bluetooth radios. The data forwarded to the cloud service is processed by Pole Star and analytics are generated to map the location of each asset.



There are two primary enhancements within this feature: enhancements to FortiAP Bluetooth capabilities and improvements in FortiAP location-based services.

Bluetooth Low Energy Profile CLI changes

FortiAP Bluetooth integrates the device discovery process for Pole Star BLE tags, leading to more efficient BLE device discovery scanning. The following new CLI settings are available under the `ble-profile`:

`scan-type`	There are two types of scanning; active and passive. • Active BLE scanning: Send a scan request for additional information from the advertiser. • Passive BLE scanning: Only receive data from the advertising device. Scan Type (default = active).
`scan-threshold`	Enter a minimum signal level/threshold in dBm required for the AP to report detected BLE device (-95 to -20, default = -90).
`scan-period`	The scan period is the total time for each round. Enter an integer value from <1000> to <10000> (default = <4000>).
`scan-time`	The scan time is the duration in which the device stays in the scanning state. Enter an integer value from <1000> to <10000> (default = <1000>).
`scan-interval`	The scan interval is the interval between the start of two consecutive scan windows. Enter an integer value from <10> to <1000> (default = <50>).
`scan-window`	The scan window is the duration the Link layer scans on one channel. Enter an integer value from <10> to <1000> (default = <50>)

To configure a Pole Star BLE profile - CLI:

```
config wireless-controller ble-profile
  edit "testpolestar"
    set ble-scanning enable
    set scan-type passive
    set scan-period 1000
    set scan-interval 30
    set scan-window 30
  next
end
```

Improvements to FortiAP location-based services

Pole Star server settings can be configured under location-based services (LBS) in the wtp-profile. The following new settings are available under `config lbs`:

`polestar`	Enable/disable Pole Star BLE NAO Track Real Time Location Service (RTLS) support (default = disable).
`polestar-protocol`	Select the protocol to report Measurements, Advertising Data, or Location Data to NAO Cloud (default = WSS).
`polestar-server-fqdn`	FQDN of Pole Star NAO Track Server (default = ws.nao-cloud.com).
`polestar-server-path`	Path of Pole Star NAO Track Server (default = /v1/token/<access_token>/pst-v2).
`polestar-server-token`	Access Token of Pole Star NAO Track Server.
`polestar-server-port`	Port of Pole Star NAO Track Server (default = 443).
`polestar-accumulation-interval`	Time that measurements should be accumulated in seconds (default = 2).
`polestar-reporting-interval`	Time between reporting accumulated measurements in seconds (default = 2).
`polestar-asset-uuid-list1`	Tags and asset UUID list 1 to be reported (string in the format of 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX').
`polestar-asset-uuid-list2`	Tags and asset UUID list 2 to be reported (string in the format of 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX').
`polestar-asset-uuid-list3`	Tags and asset UUID list 3 to be reported (string in the format of 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX').
`polestar-asset-uuid-list4`	Tags and asset UUID list 4 to be reported (string in the format of 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX').

`polestar-asset-addrgrp-list`	Tags and asset `addrgrp` list to be reported.
	The `polestar-asset-addrgrp-list` setting uses a FortiOS firewall address group to include MAC addresses of Pole Star BLE tags. Either individual MAC address or MAC address range can be supported. For example:
	`config firewall addrgrp`
	  `edit "pole-grp"`
	    `set member "addr-01" "addr-05"`
	  `next`
	`end`
	`config firewall address`
	  `edit "addr-01"`
	    `set type mac`
	    `set macaddr "ee:0f:4d:00:11:22"`
	  `next`
	  `edit "addr-05"`
	    `set type mac`
	    `set macaddr "ee:0f:4d:00:00:00-ff:ff:ff:00:00:00"`
	  `next`

To configure Pole Star location-based services - CLI:

The following example shows how to configure Pole Star location-based services and apply the previously configured Pole Star BLE profile to a FortiAP profile.

```
config wireless-controller wtp-profile
  edit "FAP431G-default"
    config platform
      set type 431G
    end
    set ble-profile "testpolestar"
    set handoff-sta-thresh 55
    config radio-1
      set band 802.11ax,n,g-only
    end
    config radio-2
      set band 802.11ax-5G
      set channel-bonding 40MHz
    end
    config radio-3
      set band 802.11ax-6G
      set channel-bonding 160MHz
    end
    config lbs
      set polestar enable
      set polestar-server-fqdn "ws-staging.nao-cloud.com"
      set polestar-server-token "nrhxj1qlmagx7dqakfihhw"
      set polestar-asset-uuid-list1 "1234*-*-*-*-12345*12"
      set polestar-asset-uuid-list2 "1234*-1234-1234-1234-123456789012"
      set polestar-asset-uuid-list3 "*-12*-*12-*-1234*55"
      set polestar-asset-uuid-list4 "12345678-1234-1234-1234-123456789012"
      set polestar-asset-addrgrp-list "pole-grp"
    end
```

```
  next
end
```

To verify the configurations:

From the FortiAP CLI, enter the following diagnostic commands:

```
FortiAP-431G # cw_diag -c ble-config
WTP Bluetooth Low Energy Configuration:
        ble scan report interval   : 30
        advertising                :
        ibeacon_uuid               : 00000000-0000-0000-0000-000000000000
        major ID                   : 0
        minor ID                   : 0
        eddystone namespace ID     :
        eddystone instance ID      :
        eddystone URL              :
        txpower                    : 0
        beacon interval            : 100
        ble scanning               : enabled (mode=passive,thresh=-
90,period=1000,time=1000,intv=30,wind=30)

BLE address: c4:39:8f:ef:5b:67
BLE oper pid: 17473
BLE conf pid: 17473


FortiAP-431G # cw_diag -c ble-polestar
BLE PoleStar Config:
   ps_enable = enabled
   ps_proto = WSS
   ps_server_fqdn = ws-staging.nao-cloud.com
   ps_server_path = /v1/token/<access_token>/pst-v2
   ps_server_token = nrhxj1qlmagx7dqakfihhw
   ps_server_port = 443
   ps_acc_intv = 2
   ps_rpt_intv = 2
   ps_addrgrp_uuid_policy = allow
       B001 12340000-0000-0000-0000-123450000012 - ffff0000-0000-0000-0000-fffff00000ff
       B002 12340000-1234-1234-1234-123456789012 - ffff0000-ffff-ffff-ffff-ffffffffffff
       B003 00000000-1200-0012-0000-123400000055 - 00000000-ff00-00ff-0000-ffff000000ff
       B004 12345678-1234-1234-1234-123456789012 - ffffffff-ffff-ffff-ffff-ffffffffffff
   ps_addrgrp_policy = allow
       S005 ee:0f:4d:00:11:22
       B006 ee:0f:4d:00:00:00 - ff:ff:ff:00:00:00
   ps_ble_dev_max_rpt = 128
   ps_ble_dev_max_batch = 64
```

Wireless Foreground Scan improvements - 7.4.1

This information is also available in the FortiWiFi and FortiAP 7.4 Configuration Guide:

- Enabling AP scan channel lists to optimize foreground scanning

This release optimizes wireless foreground scanning by limiting the number of radio channels scanned. When DAARP, location-based services (LBS) for FortiPresence, or rogue AP monitoring are configured, you can select which channels to run a wireless foreground scan on based on frequency bands. With fewer channels to scan, the overall dwell cycle time is reduced while the frequency of the reporting interval is increased.

Under the Wireless Intrusion Detection System (WIDS) profile, the following CLI commands have been added to configure the selected channels:

```
config wireless-controller wids-profile
  edit < WIDS_profile_name >
    set ap-scan enable
    set ap-scan-channel-list-2G-5G < channel-1 > < channel-2 > ... < channel-x >
    set ap-scan-channel-list-6G < channel-1 > < channel-2 > ... < channel-y >
  next
end
```

`ap-scan-channel-list-2G-5G`	Add the 2.4G and 5G band AP channels you want to scan.
`ap-scan-channel-list-6G`	Add the 6G band AP channels you want to scan.

To create a WIDS profile to scan for specific radio channels:

1. Create a WIDS profile and add the selected channels to the appropriate AP scan channel list:

```
config wireless-controller wids-profile
  edit "wids.test"
    set sensor-mode both
    set ap-scan enable
    set ap-scan-channel-list-2G-5G "1" "6" "149" "161"
    set ap-scan-channel-list-6G "109" "201" "217"
  next
end
```

To scan specified 2.4G and 5G channels:

1. From the FortiAP profile, enable dedicated scanning and set Radio 3 to monitor mode with the WIDS profile applied.

```
config wireless-controller wtp-profile
  edit "FAP431G.ddscan"
    config platform
      set type 431G
      set ddscan enable
    end
    set handoff-sta-thresh 55
    config radio-1
      set band 802.11ax,n,g-only
    end
    config radio-2
      set band 802.11ax-5G
    end
    config radio-3
      set mode monitor
      set wids-profile "wids.test"
    end
```

```
  next
end
```

Radio 3 will scan the 2.4G and 5G channels specified in `ap-scan-channel-list-2G-5G`.

2. Verify that the scan is only run on the specified 2.4G and 5G channels.

```
FortiGate-40F # diag wireless-controller wlac -c ap-rogue
CMWP AP: vf                  bssid ssid             ch  rate  sec
 signal noise  age      sta mac                 wtp cnt     ici    bw sgi band
        freq(MHz)
UNNN AP: 0       04:d5:90:4a:19:b1 FOS_test_001_... 161 260   WPA3 OWE              -
55 -95    562      00:00:00:00:00:00       1   /1     none    20 0  11ACVHT20 (wave2)
   5805

 N                FP431GTY22003576 FOS_test_001_... 161 260   WPA3 OWE              -
55 -95    562      172.20.1.29:5246 -2  11

UNNN AP: 0       06:18:d6:67:29:42                  6   144   WPA2 Personal         -
85 -95    958      00:00:00:00:00:00       1   /1     none    20 1  11NGHT20
   2437

 N                FP431GTY22003576                  6   144   WPA2 Personal         -
85 -95    958      172.20.1.29:5246 -2  11

UNNN AP: 0       06:93:7c:65:49:f8                  1   1181  WPA2 Personal         -
87 -95    688      00:00:00:00:00:00       1   /1     none    20 1  11AXGHE20
   2412

 N                FP431GTY22003576                  1   1181  WPA2 Personal         -
87 -95    688      172.20.1.29:5246 -2  11

UNNN AP: 0       90:6c:ac:45:5b:8a Example_001_test 149 130   WPA2 Personal         -
69 -95    51438    00:00:00:00:00:00       1   /1     none    20 0  11NAHT20 (wave2)
   5745

 N                FP431GTY22003576 Example_001_test 149 130   WPA2 Personal         -
69 -95    51438    172.20.1.29:5246 -2  11
```

To scan specified 6G channels:

1. From the FortiAP profile, **do not** enable dedicated scanning. Set Radio 3 to monitor mode with the WIDS profile applied.

```
config wireless-controller wtp-profile
  edit "FAP431G.noddscan"
    config platform
      set type 431G
    end
    set handoff-sta-thresh 55
    config radio-1
      set band 802.11ax,n,g-only
    end
    config radio-2
      set band 802.11ax-5G
    end
    config radio-3
```

```
        set mode monitor
        set wids-profile "wids.test"
      end
    next
end
```

Radio 3 will scan the 6G channels specified in `ap-scan-channel-list-6G`.

2. Verify that the scan is only run on the specified 6G channels.

```
FortiGate-40F # diag wireless-controller  wlac -c ap-rogue

CMWP AP: vf                bssid ssid             ch   rate   sec
signal noise  age      sta mac                 wtp cnt      ici     bw sgi band
              freq(MHz)

UNNN AP: 0        84:39:8f:1f:0e:c8 test01-GUI-SS... 109 1147  WPA3 SAE              -
80 -95    6         00:00:00:00:00:00       1    /1     none     20 0  11AX6HE20-6G
 6495

 N                 FP431GTY22003576 test01-GUI-SS... 109 1147  WPA3 SAE              -
80 -95    6         172.20.1.29:5246 -2   17
```

Support for MIMO mode configuration - 7.4.1

This information is also available in the FortiWiFi and FortiAP 7.4 Configuration Guide:

- Configure FortiAP MIMO values

This release enables Multiple-Input Multiple-Output (MIMO) values to be configured on select FortiAP and FortiAP-U models, offering users more flexibility when using a third-party antenna connected to a limited number of AP ports.

MIMO mode configuration is supported on the following:

Family	Series
FortiAP	F and G series models
FortiAP-U	EV and F series models

MIMO values can be set under radio configuration when creating or editing a FortiAP profile. The value range available is confined within each AP platform and radio's MIMO specifications (default, 1x1, 2x2, 3x3, 4x4, 8x8).

```
config wireless-controller wtp-profile
  edit < profile_name >
    config radio-< number >
      set mimo-mode [ actual modes supported depend on AP platform ]
    end
  next
end
```

For example, FAP-231G radios support a maximum of 2x2 MIMO, so you can select between 1x1 or 2x2. Meanwhile FAP-831F radios support a maximum of 8x8 MIMO, so you can select between 1x1, 2x2, 3x3, 4x4 or 8x8.

To configure MIMO mode values:

```
config wireless-controller wtp-profile
  edit FAP431G-default
    config radio-1
      set mimo-mode 3x3
    end
    config radio-2
      set mimo-mode 3x3
    end
    config radio-3
      set mimo-mode 2x2
    end
  end
```

To verify that the MIMO mode settings have been applied:

```
FortiAP-431G # rcfg | grep mimo
   mimo,chainmask : 3, 0x7 (mimo) 0xf (power) 0x7/0x7 (oper)
   mimo,chainmask : 3, 0x70 (mimo) 0xf0 (power) 0x70/0x70 (oper)
   mimo,chainmask : 2, 0x3 (mimo) 0xf (power) 0x3/0x3 (oper)
```

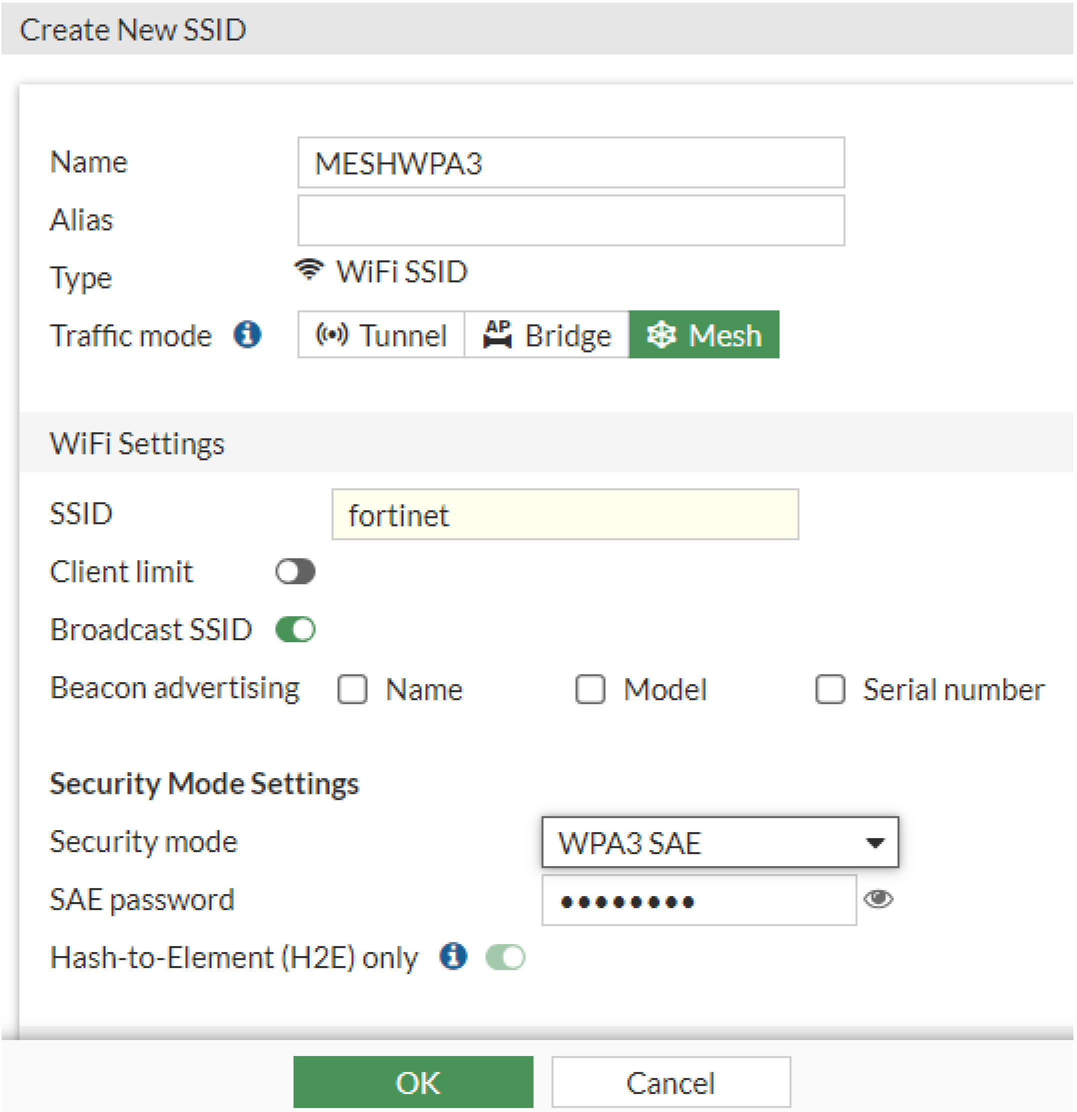
Add GUI support for configuring WPA3-SAE security mode on mesh backhaul SSIDs - 7.4.1

Support for configuring WPA3-SAE security mode on FortiAP mesh backhaul SSIDs using the CLI was added in FortiOS 7.4.0. FortiOS 7.4.1 adds GUI support. For more information, see Add support for WPA3-SAE security mode on mesh backhaul SSIDs on page 507.

This release adds GUI supports for configuring WPA3-SAE security mode for FortiAP wireless mesh backhaul SSIDs. Wi-Fi 6E FortiAPs can also set up mesh connections over the 6GHz band as WPA3-SAE (with Hash-to-Element only enabled) is mandatory in Wi-Fi 6E technology.

To configure WPA3-SAE security mode on mesh route SSIDs - GUI:

1. From the FortiGate GUI, navigate to *WiFi & Switch Controller* > *SSIDs* and select *Create New* > *SSID*.
2. In *Traffic mode*, select *Mesh*.
3. Under *Security Mode Settings*, set the *Security mode* to *WPA3 SAE*.
4. In *SAE password*, enter an *SAE password*.



5. By default, *Hash-to-Element (H2E) only* is enabled and cannot be disabled as it is mandatory for WiFi 6E technology.
6. When you are finished, click *OK*.

Add support for SAE-PK generation - 7.4.2

This release adds support for generating an SAE-PK private key and password in FortiOS, a crucial component for SAE-PK authentication and WPA3 Security configuration.

The following CLI command has been added:

```
execute wireless-controller create-sae-pk [SSID] [curve:prime256v1|secp384r1|secp521r1]
```

You can use the CLI command to create a SAE-PK private key and password directly in FortiOS. Once the private key and password are generated, you can then apply them to an SSID with the security mode set to a WPA3-SAE option and SAE-PK authentication enabled.

To generate a SAE private key and password - CLI:

1. Use the SAE-PK generation command to create a SAE-PK Private Key and password. In this example, the SSID is "*Example_wpa3_sae_pk*" with the curve set to *prime256v1*.

   ```
   execute wireless-controller create-sae-pk Example_wpa3_sae_pk prime256v1
   ```

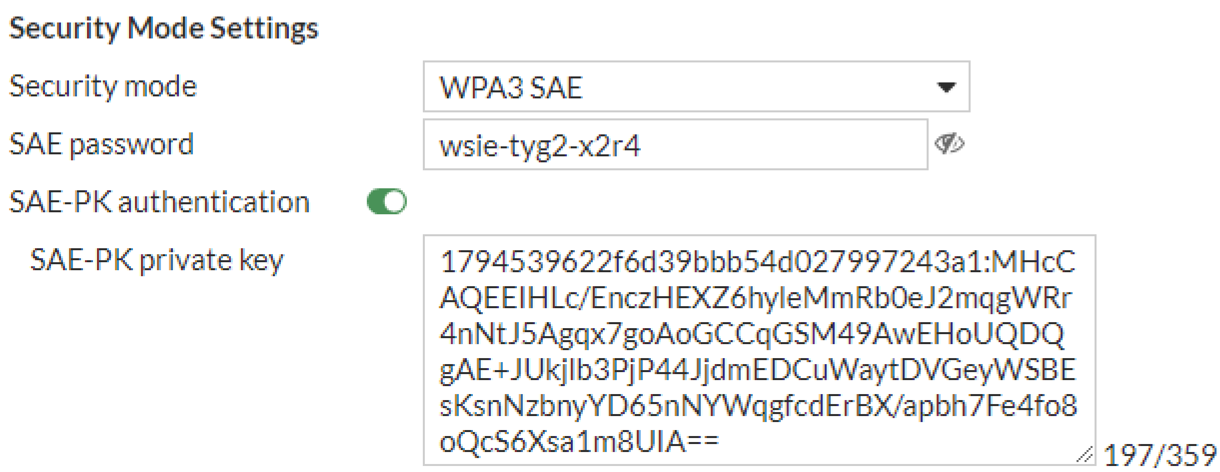
2. The command runs and displays the following:

   ```
   sae_pk_gen ssid Example_wpa3_sae_pk sec 3 curve prime256v1:

   Searching for a suitable Modifier M value
   12.98%Found a valid hash in 2178339 iterations:
   0000006920878369f515848ab8d3047dc106a231c7ddd19e86ea1f2435d31f26
   PasswordBase binary data for base32:
   b49049e0dabea2b848abc69829f7048d4469c7dde8cf49ba87e486bd31# SAE-PK password/M/private
   key for Sec=3.
   sae_password=wsie-tyg2-x2r4
   pk
   =1794539622f6d39bbb54d027997243a1:MHcCAQEEIHLc/EnczHEXZ6hyleMmRb0eJ2mqgWRr4nNtJ5Agqx7goA
   oGCCqGSM49AwEHoUQDQgAE+JUkjlb3PjP44JjdmEDCuWaytDVGeyWSBEsKsnNzbnyYD65nNYWqgfcdErBX/apbh7
   Fe4fo8oQcS6Xsa1m8UIA==
   # Longer passwords can be used for improved security at the cost of usability:
   # wsie-tyg2-x2rl-qsfs
   # wsie-tyg2-x2rl-qsfl-y2mr
   # wsie-tyg2-x2rl-qsfl-y2mc-t5yi
   # wsie-tyg2-x2rl-qsfl-y2mc-t5ye-rvc6
   # wsie-tyg2-x2rl-qsfl-y2mc-t5ye-rvcg-tr6e
   # wsie-tyg2-x2rl-qsfl-y2mc-t5ye-rvcg-tr65-5dhj
   # wsie-tyg2-x2rl-qsfl-y2mc-t5ye-rvcg-tr65-5dhu-touh
   # wsie-tyg2-x2rl-qsfl-y2mc-t5ye-rvcg-tr65-5dhu-touh-4sdz
   # wsie-tyg2-x2rl-qsfl-y2mc-t5ye-rvcg-tr65-5dhu-touh-4sdl-2mpz
   ```

3. Copy the *sae-password* and *pk* values.
   - *sae-password* is the SAE Password. You can also copy one of the longer passwords instead for improved security.
   - *pk* is the SAE Private Key.

To apply the generated SAE private key and password to an SSID - GUI:

1. Go to *WiFi Controller > SSID* and select the SSID you want to apply the SAE-PK to.
2. In the *WiFi Settings* section, set the *Security Mode* to a WPA3 option.
3. In *SAE password*, paste the *sae_password* value you previously generated.
4. Enable *SAE-PK authentication*.
5. In *SAE-PK private key*, paste the *pk* value you previously generated.



6. When you are finished, click *OK*.

To apply the generated SAE private key and password to an SSID - CLI:

1. From the FortiOS CLI, go to the SSID you want to configure and enter the SAE-PK Private Key and Password values you copied:

```
config wireless-controller vap
  edit "wpa3-test"
    set ssid "Example_wpa3_sae_pk"
    set security wpa3-sae
    set sae-pk enable
    set sae-private-key
"1794539622f6d39bbb54d027997243a1:MHcCAQEEIHLc/EnczHEXZ6hyleMmRb0eJ2mqgWRr4nNtJ5Agqx7goA
oGCCqGSM49AwEHoUQDQgAE+JUkjlb3PjP44JjdmEDCuWaytDVGeyWSBEsKsnNzbnyYD65nNYWqgfcdErBX/apbh7
Fe4fo8oQcS6Xsa1m8UIA=="
    set sae-password wsie-tyg2-x2r4
  next
end
```

2. After applying the SSID to a FortiAP, confirm the WiFi station can connect.

```
diagnose wireless-controller wlac -d sta online
   vf=0 mpId=0 wtp=3 rId=2 wlan=wpa3-test vlan_id=0 ip=0.0.0.0 ip6=::
mac=f8:e4:e3:d8:5e:af vci= host= user= group= signal=-9 noise=-89 idle=1 bw=0 use=3
chan=161 radio_type=11AC(wave2) security=wpa3_sae mpsk= encrypt=aes cp_authed=no l3r=1,0
G=0.0.0.0:0,0.0.0.0:0-0-0 -- 0.0.0.0:0 0,0 online=yes mimo=2
```

Support RADIUS accounting interim update on roaming for WPA Enterprise security - 7.4.2

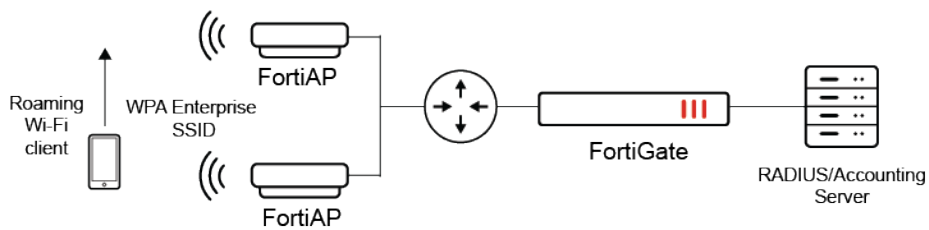
This enhancement adds a CLI option to support accounting interim updates on SSIDs using RADIUS authentication with a WPA Enterprise security mode. This accounting message resolves compatibility issues with Cisco's Identity Services Engine (ISE) session stitching feature. When a Wi-Fi station roams between FortiAPs, the FortiGate creates an "Interim-Update" accounting message with the same "Acct-Session-Id" value to avoid interrupting the ISE session.

CLI Changes:

```
config wireless-controller vap
    edit <name>
      set security wpa2-only-enterprise
      set roaming-acct-interim-update enable
    next
end
```

Note that `roaming-acct-interim-update` can only be enabled when the security mode is set to a WPA Enterprise type.

Example Topology:



To enable roaming account interim updates - CLI:

1. Create a RADIUS server with an accounting server:

```
config user radius
  edit "peap"
    set server "172.18.56.104"
      set secret ENC
      set nas-ip 192.168.1.10
      set nas-id-type custom
      set nas-id "FWF-61F-AUTH"
      set acct-interim-interval 300
      set radius-coa enable
      set password-renewal disable
      config accounting-server
        edit 1
          set status enable
          set server "172.18.56.104"
          set secret ENC
      next
    end
  next
end
```

2. Create a WPA2-Enterprise SSID with the authentication method set to `radius` and the radius server set to the example you previously configured (`peap`).

```
config wireless-controller vap
  edit "wifi4"
    set ssid "FOS_61F_ENT"
    set security wpa2-only-enterprise
    set auth radius
    set radius-server "peap"
```

```
    set schedule "always"
  next
end
```

3. Enable `roaming-acct-interim-update`.

```
config wireless-controller vap
  edit "wifi4"
    set ssid "FOS_61F_ENT"
    set security wpa2-only-enterprise
    set auth radius
    set radius-server "peap"
    set schedule "always"
    set roaming-acct-interim-update enable
  next
end
```

4. Apply this SSID to the FortiAPs you want to roam between.

To verify that roaming account interim updates are successful:

1. Connect a Wi-Fi client to one FortiAP (FAP23JF) and check the `Acct-Status-Type` is `Start`. Take note of the `Acct-Session-Id` value (`653FE2DC00000003`).

```
Mon Oct 30 10:17:45 2023
Acct-Status-Type = Start
Acct-Authentic = RADIUS
User-Name = "tester"
NAS-IP-Address = 192.168.1.10
NAS-Identifier = "FWF-61F-AUTH"
Called-Station-Id = "E0-23-FF-B2-15-48:FOS_61F_ENT"
NAS-Port-Type = Wireless-802.11
Service-Type = Framed-User
NAS-Port = 1
Fortinet-SSID = "FOS_61F_ENT"
Fortinet-AP-Name = "FP23JFTF20000015"
Calling-Station-Id = "5C-1B-F4-89-F4-36"
Connect-Info = "CONNECT 5/5Mbps(Tx/Rx) 11AX_5G"
Acct-Session-Id = "653FE2DC00000003"
```

2. Let the Wi-Fi client roam to a different FortiAP (FAP223E) and verify that the `Acct-Status-Type` is `Interim-Update` and that the `Acct-Session-Id` value remains the same as before ((`653FE2DC00000003`).

```
Mon Oct 30 10:36:37 2023
Acct-Status-Type = Interim-Update
Acct-Authentic = RADIUS
User-Name = "tester"
NAS-IP-Address = 192.168.1.10
NAS-Identifier = "FWF-61F-AUTH"
Called-Station-Id = "E8-1C-BA-9E-5D-98:FOS_61F_ENT"
NAS-Port-Type = Wireless-802.11
Service-Type = Framed-User
NAS-Port = 1
Fortinet-SSID = "FOS_61F_ENT"
Fortinet-AP-Name = "DESK-223E"
Calling-Station-Id = "5C-1B-F4-89-F4-36"
Connect-Info = "CONNECT 0/0Mbps(Tx/Rx) 11AC"
Acct-Session-Id = "653FE2DC00000003"
```

Improve Bonjour profile provisioning and redundancy - 7.4.2

This release improves FortiAP Bonjour profiles with the following features:

- **Simplified Bonjour Profile Provisioning**: You can now set the Bonjour profile at the FortiAP profile level so that it can apply to multiple FortiAP devices. If a Bonjour profile is applied at both the device and profile level, the configuration made at the device level takes precedence.
- **Failover Mechanism**: To ensure uninterrupted service, a new election procedure provides a failover mechanism or redundancy in case the Bonjour Gateway goes down. After a Bonjour profile is applied to multiple APs, the APs execute an algorithm to determine the Bonjour Default Gateway. The AP with the highest base MAC address is selected as the primary default gateway while the other APs are designated as backup default gateways in case the primary default gateway becomes unavailable.

To apply a Bonjour profile at the FortiAP profile level - CLI:

1. Create a Bonjour profile:

```
config wireless-controller bonjour-profile
  edit "Bonjour-Profile"
    config policy-list
      edit 1
        set description "All"
        set from-vlan "10"
        set to-vlan "20"
      next
      edit 2
        set from-vlan "101"
        set to-vlan "202"
      next
    end
  next
end
```

2. Apply the Bonjour profile to a FortiAP profile:

```
config wireless-controller wtp-profile
  edit FAP234F-default
    set bonjour-profile "Bonjour-Profile"
  next
end
```

3. When you select the Bonjour profile, a notice loads asking you to ensure that the FortiAPs you apply the Bonjour profile to have auto-election enabled. Enter `y` to confirm.

To verify that the Bonjour profile is successfully applied to a FortiAP:

1. From the FortiAP CLI, enter `cw_diag -c bonjour`:

```
cw_diag -c bonjour
Bonjour Gateway: Controlled by AC
Configured Bonjour Vlans:
    10    ==> 20    services 00000001  all
    101   ==> 202   services 00000001  all
```

```
Total 2 Bonjour Vlans
Bonjour Gateway Election Info:
1/2 e8:ed:d6:a5:2e:e8 state=cap,8825 live=16605 age=1
2/2 e8:ed:d6:a5:31:08 state=oper,8807 live=8825 age=1
---- e0:23:ff:b2:18:68 state=cap,16609
```

The diagnoses output also provides details of the last election process under "Bonjour Gateway Election Info". The AP with the MAC address of `8:ed:d6:a5:31:08` is in the `oper` state, meaning it serves as the default gateway. Another two APs are in the `cap` state, meaning they act as back-up gateways in case the primary gateway becomes unavailable. If there are any more APs in the same setup, they will go into a `hold` state.

GUI support for WPA3 security mode on Client mode FortiWiFi units - 7.4.2

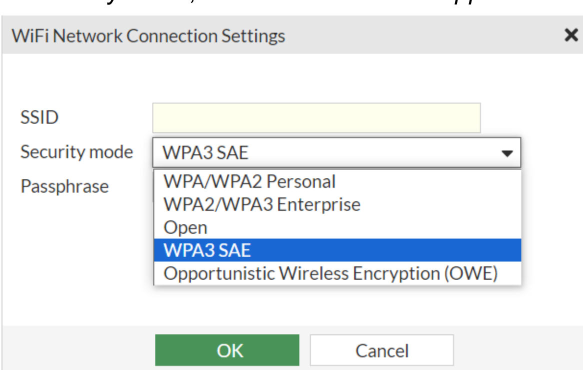
Support for configuring WPA3-SAE security mode on client mode FortiWiFi using the CLI was added in FortiOS 7.4.0. FortiOS 7.4.2 adds GUI support. For more information, see Support for WPA3 security modes on FortiWiFi units operating in Client Mode on page 515.

This release adds GUI support when selecting WPA3-SAE and OWE security modes on FortiWiFi units operating in wireless client mode. When the local radio of a FortiWiFi 8xF/6xF/40F model is operating in client mode, it can connect with third-party SSIDs with a WPA3-SAE or OWE security mode.

To configure WPA3 security mode SSID on a FortiWiFi running in client mode - GUI:

1. Go to *WiFi and Switch Controller > Local WiFi Radio* and change the *Mode* to *Wireless Client*.

   Note: You must remove any AP WiFi configurations such as SSIDs, DHCP servers, policies, and software switch members before you can change the mode to Wireless Client. Once you select Wireless Client, the FortiWiFi unit will reboot.
2. Click *Add Network*.
3. In *SSID*, enter the name of the SSID you want to use.
4. In *Security mode*, select *WPA3 SAE* or *Opportunistic Wireless Encryption (OWE)*.



5. If you selected WAP3 SAE, enter a *Passphrase*.
6. When you are finished, click *OK*.

Support WPA3 options when the FortiAP radio mode is set to SAM - 7.4.2

This release supports WPA3 (Wi-Fi Protected Access 3) options when the radio mode is set to Fortinet's SAM (Service Assurance Manager). This includes WPA3-SAE (Simultaneous Authentication of Equals) and WPA3 OWE (Opportunistic Wireless Encryption). It also includes support for WPA2/WPA3-Enterprise with certificate authentication, encompassing both PEAP and EAP-TLS.

CLI changes:

```
config wireless-controller wtp-profile
  edit < name >
    config radio-1
      set mode sam
      set sam-ssid < string >
      set sam-security-type { wpa-enterprise |wpa3-sae | owe }
    end
  next
end
```

Example use case

In this example, a FortiGate manages two FortiAPs. One FortiAP (FAP_1) broadcasts a test SSID using WPA3 security, while the second FortiAP (FAP_2) is configured as a SAM test client with the same WPA3 security method so it can connect with the SSID on FAP_1 and perform a SAM ping or Iperf test.

The following example shows how to configure a FortiAP profile with WPA3 Enterprise using EAP-TLS, WPA3-SAE, and OWE authentication.

To configure a FortiAP profile to run in SAM mode - CLI:

1. (Optional) Upload the CA certificate to verify the server certificate.
   a. Go to System > Certificates > Create/Import > CA Certificate and complete the fields to upload the certificate.
2. (Optional) Upload the client certificate with private key file.
   a. Go to *System* > *Certificates* > *Create/Import* > *Certificate* and click *Import Certificate*.
   b. Select *Certificate* or *PKCS #12 Certificate*, then follow the onscreen instructions to import the client certificate with private key file, and set the private-key-password.
3. Create an SSID and select an authentication method:

WPA3 Enterprise authentication using EAP-TLS	WPA3-SAE authentication	OWE authentication

`config wireless-controller vap`
`  edit "sam-test-ent3"`
`    set ssid "sam-test-ent3"`
`    `**`set security wpa3-only-enterprise`**
`    set pmf enable`
	`config wireless-controller vap`
	`  edit "sam-test-sae"`
	`    set ssid "sam-test-sae"`
	`    `**`set security wpa3-sae`**
	`    set pmf enable`
	`    set schedule "always"`
	`    set sae-password ENC`
		`config wireless-controller vap`
		`  edit "sam-test-owe"`
		`    set ssid "sam-test-owe"`
		`    `**`set security owe`**
		`    set pmf enable`
		`    set schedule "always"`
		`  next`

WPA3 Enterprise authentication using EAP-TLS	WPA3-SAE authentication	OWE authentication
set auth radius **set radius-server "eap_tls"** set schedule "always" next end	next end	end

4. Broadcast the SSID on FAP_1:

WPA3 Enterprise authentication using EAP-TLS	WPA3-SAE authentication	OWE authentication
config wireless-controller wtp-profile edit "FAP433F-sam-test" config platform set type 433F set ddscan enable end config radio-2 set band 802.11ax-5G set vap-all manual **set vaps "sam-test-ent3"** end next end	config wireless-controller wtp-profile edit "FAP433F-sam-test " config platform set type 433F set ddscan enable end config radio-2 set band 802.11ax-5G set vap-all manual **set vaps "sam-test-sae"** end next end	config wireless-controller wtp-profile edit "FAP433F-sam-test" config platform set type 433F set ddscan enable end config radio-2 set band 802.11ax-5G set vap-all manual **set vaps "sam-test-owe"** end next end

5. Configure the AP profile for FAP_2 to run in SAM mode and select a SAM security type. Then enable a SAM ping or Iperf test:

SAM ping test with WPA3 Enterprise authentication using EAP-TLS	SAM Iperf test with WPA3-SAE authentication	SAM ping test with OWE authentication
If the SAM security type is set to `wpa-enterprise`, you can configure SAM EAP methods and SAM certificate settings: config wireless-controller wtp-profile edit "FAP431F-sam-ent3" config radio-2 set mode sam set sam-ssid "sam-	config wireless-controller wtp-profile edit "FAP431F-sam-sae" config radio-2 set mode sam set sam-ssid "sam-test-sae" **set sam-security-type wpa3-sae** set sam-password ENC **set sam-test iperf** set sam-server-ip	config wireless-controller wtp-profile edit "FAP431F-sam-owe" config radio-2 set mode sam set sam-ssid "sam-test-owe" **set sam-security-type owe** set sam-server-ip 8.8.8.8 **set sam-test ping**

SAM ping test with WPA3 Enterprise authentication using EAP-TLS	SAM Iperf test with WPA3-SAE authentication	SAM ping test with OWE authentication
test-ent3" **set sam-security-type wpa-enterprise** set sam-eap-method tls set sam-client-certificate "client2.cert" set sam-private-key "client2.cert" set sam-private-key-password ENC set sam-ca-certificate "CA_Cert_1" set sam-username "tester" set sam-password ENC **set sam-test ping** set sam-server-ip 8.8.8.8 set sam-report-intv 60 end next end	"172.18.56.99" set iperf-server-port 5201 set iperf-protocol tcp set sam-report-intv 60 end next end	set sam-report-intv 60 end next end

When the "sam-eap-method" is "tls" or "both", the "sam-client-certificate", "sam-private-key", and "sam-private-key-password" settings are required.

- `sam-client-certificate`: The name of imported client certificate.
- `sam-private-key`: Uses the same name of imported client certificate.
- `sam-private-key-password`: Created when importing the client certificate.
- `sam-ca-certificate`: The name of the imported CA certificate.

6. Log in to the FAP_2 CLI to verify the configurations:

SAM ping test with WPA3 Enterprise authentication using EAP-TLS	SAM Iperf test with WPA3-SAE authentication	SAM ping test with OWE authentication
FortiAP-431F # rcfg < other output omitted > sam ssid : sam-test-ent3 sam bssid : 00:00:00:00:00:00	FortiAP-431F # rcfg sam ssid : sam-test-sae sam bssid : 00:00:00:00:00:00 sam security type : SAE	FortiAP-431F # rcfg < other output omitted > sam ssid : sam-test-owe sam bssid : 00:00:00:00:00:00

SAM ping test with WPA3 Enterprise authentication using EAP-TLS	SAM Iperf test with WPA3-SAE authentication	SAM ping test with OWE authentication

`sam security type : Enterprise`
`sam captive portal : disabled`
`sam test : `**`Ping`**
`sam server : 8.8.8.8`
`sam report interval: 60`
`sam eap method : EAP TLS`
`sam client cert : 1`
`sam ca cert : 1`
`< other output omitted >`
	`sam captive portal : disabled`
	`sam test : `**`Iperf`**
	`sam server : 172.18.56.99`
	`sam report interval: 60`
	`sam iperf port : 5201`
	`sam iperf protocol : TCP`
	`< other output omitted >`
		`sam security type : OWE`
		`sam captive portal : disabled`
		`sam test : `**`Ping`**
		`sam server : 8.8.8.8`
		`sam report interval: 60`
		`< other output omitted >`

7. The FortiOS WiFi event log shows the corresponding event:

WPA3 Enterprise authentication using EAP-TLS	SAM Iperf test with WPA3-SAE authentication	SAM ping test with OWE authentication
`1: date=2023-11-10 time=12:02:16 eventtime=1699646536236321385 tz="-0800" logid="0104043711" type="event" subtype="wireless" level="notice" vd="root" logdesc="SAM ping test result" sn="FP431FTF23031585" ap="FP431FTF23031585" vap="sam-test-ent3" ssid="sam-test-ent3" stamac="80:80:2c:0c:01:9f" radioid=2 channel=161 security="WPA3 Enterprise Only" encryption="AES" action="sam-ping-result" msg="Connected to AP FP433FTF20001556, 0.0% packet loss" remotewtptime="3012.616987"`	`1: date=2023-11-10 time=12:20:31 eventtime=1699647630989156870 tz="-0800" logid="0104043710" type="event" subtype="wireless" level="notice" vd="root" logdesc="SAM iperf test result" sn="FP431FTF23031585" ap="FP431FTF23031585" vap="sam-test-sae" ssid="sam-test-sae" stamac="80:80:2c:0c:01:9f" radioid=2 channel=161 security="WPA3 SAE" encryption="AES" action="sam-iperf-result" msg="Connected to AP FP433FTF20001556, TCP, max rate 0.6 MB/s" remotewtptime="11.468787"`	`1: date=2023-11-10 time=12:28:11 eventtime=1699648091131525936 tz="-0800" logid="0104043711" type="event" subtype="wireless" level="notice" vd="root" logdesc="SAM ping test result" sn="FP431FTF23031585" ap="FP431FTF23031585" vap="sam-test-owe" ssid="sam-test-owe" stamac="80:80:2c:0c:01:9f" radioid=2 channel=161 security="OWE" encryption="AES" action="sam-ping-result" msg="Connected to AP FP433FTF20001556, 0.0% packet loss" remotewtptime="469.609833"`

Add automated reboot functionality for FortiAPs - 7.4.2

This release enables FortiAPs to automatically reboot when they are stuck in an AP Controller (AC) discovery dead loop, eliminating the need to manually reboot or power cycle those FortiAP units to recover. FortiAPs have a configurable timeout period during AC discovery and can automatically reboot if they do not detect an active AC within the set time interval. Once the FortiAPs reboot, they can detect any changes made to the LAN/WAN and discover the AC.

The following CLI commands have been added to configure automatic AP reboot:

```
config wireless-controller timers
  set ap-reboot-wait-interval < integer >
  set ap-reboot-wait-time < hh:mm >
  set ap-reboot-wait-interval2  < integer >
end
```

`ap-reboot-wait-interval1`	Time in minutes to wait before the AP reboots when there is no controller detected (5 - 65535, default = 0, 0 for no reboot). Applies only to FortiAP units that have no local-standalone SSID assigned.
`ap-reboot-wait-time`	Time to reboot the AP when there is no controller detected and standalone SSIDs are pushed to the AP in the previous session, format hh:mm. This command apples to FortiAPs with at least one local-standalone SSID and ones with no local-standalone SSIDs. If both "`ap-reboot-wait-interval1`" and "`ap-reboot-wait-time`" are set, FortiAPs *with* standalone SSIDs will reboot at the configured "`ap-reboot-wait-time`" every day. However, FortiAPs *without* standalone SSIDs will reboot after waiting for "`ap-reboot-wait-interval1`" or "`ap-reboot-wait-time`", whichever come first.
`ap-reboot-wait-interval2`	Time in minutes to wait before the AP reboots when there is no controller detected and standalone SSIDs are pushed to the AP in the previous session (5 - 65535, default = 0, 0 for no reboot). Applies only to FortiAP units that have at least one local-standalone SSID assigned.

For automatic reboot to be enabled, the FortiAPs need to be managed by a FortiGate once and have an interval and wait-time set from the FortiGate side. Only then will the APs auto-reboot if they cannot detect an active AC.

To configure FortiAP automatic reboot intervals - CLI:

1. Configure the FortiAP reboot interval:

   ```
   config wireless-controller timers
     set ap-reboot-wait-interval1 5
     set ap-reboot-wait-interval2 10
   end
   ```

2. Assign a non-standalone SSID to FAP1:

   ```
   config wireless-controller vap
     edit "test_bridge"
       set ssid "test_bridge"
   ```

```
    set passphrase ENC
    set local-bridging enable
    set schedule "always"
  next
end
```

3. Assign a standalone SSID to FAP2:

```
config wireless-controller vap
  edit "test_standalone"
    set ssid "test_standalone"
    set passphrase ENC
    set local-standalone enable
    set local-bridging enable
    set schedule "always"
  next
end
```

4. Verify that FAP1 is managed by FortiGate and has an SSID assigned with local-standalone disabled:

```
FortiAP-432FR # wcfg | grep fsm
    fsm-state            : RUN 1801
FortiAP-432FR #
FortiAP-432FR # vcfg
-------------------------------VAP Configuration    1---------------------------
Radio Id  1 WLAN Id  0 test_bridge ADMIN_UP(INTF_UP) init_done 0.0.0.0/0.0.0.0 unknown
(-1)
           vlanid=0, intf=wlan10, vap=0x18d0002c, bssid=74:78:a6:e3:63:48
           11ax high-efficiency=enabled target-wake-time=enabled
           bss-color-partial=enabled
           mesh backhaul=disabled
           local_auth=disabled standalone=disabled nat_mode=disabled

FortiAP-432FR # cw_diag -c acs | grep "last seen"
AC last seen time:  0 SSID cnt 0,0 ap reboot wait time 300,600 00:00
```

5. Verify that FAP2 is managed by FortiGate and has an SSID assigned with local-standalone enabled.:

```
FortiAP-831F # wcfg | grep fsm
    fsm-state            : RUN 589
FortiAP-831F #
FortiAP-831F # vcfg
-------------------------------VAP Configuration    1---------------------------
Radio Id  1 WLAN Id  0 test_standalone ADMIN_UP(INTF_UP) init_done 0.0.0.0/0.0.0.0
unknown (-1)
           vlanid=0, intf=wlan10, vap=0x3b61f02c, bssid=e8:ed:d6:b8:02:f8
           11ax high-efficiency=enabled target-wake-time=enabled
           bss-color-partial=enabled
           mesh backhaul=disabled
           local_auth=enabled standalone=enabled nat_mode=disabled

FortiAP-831F # cw_diag -c acs | grep "last seen"
AC last seen time:  0 SSID cnt 0,0 ap reboot wait time 300,600 00:00
```

6. When the FortiAPs are disconnected from the FortiGate, they will reboot at the configured time.
   - The FortiAP with no standalone SSID (FAP1) reboots at the time interval configured in interval1 (5 minutes or 300 seconds).

```
FortiAP-432FR # 03901.181 *****cwFwctlReboot:*****
03901.181     SSID_CNT 1,0. No AC is found in 309 sec (> 300) Rebooting...
[ 4134.665936] reboot: Restarting system
```

- The FortiAP with standalone SSID (FAP2) reboots at the time interval configured in interval2 (10 minutes or 600 seconds).

```
FortiAP-831F login: 01548.738 *****cwFwctlReboot:*****
01548.738     SSID_CNT 1,1. No AC is found in 625 sec (> 600) Rebooting...
[ 1603.673046] reboot: Restarting system
```

To configure FortiAP automatic reboot intervals and wait time - CLI:

When `ap-reboot-wait-interval1` and `ap-reboot-wait-time` is configured, FortiAPs *without* standalone SSIDs wait for `ap-reboot-wait-interval1` or `ap-reboot-wait-time` (whichever comes first). Meanwhile FortiAPs *with* standalone SSIDs wait for the set time in `ap-reboot-wait-time` before automatically rebooting.

1. Configure the FortiAP reboot interval and wait time:

```
config wireless-controller timers
  set ap-reboot-wait-interval1 5
  set ap-reboot-wait-time "15:50"
end
```

2. Verify that FAP1 is managed by FortiGate and has an SSID assigned with local-standalone disabled:

```
FortiAP-432FR # wcfg | grep fsm
fsm-state            : RUN 463
FortiAP-432FR # vcfg
-------------------------------VAP Configuration    1----------------------------
Radio Id  1 WLAN Id  0 test_bridge ADMIN_UP(INTF_UP) init_done 0.0.0.0/0.0.0.0 unknown
(-1)
           vlanid=0, intf=wlan10, vap=0x3915502c, bssid=74:78:a6:e3:63:48
           11ax high-efficiency=enabled target-wake-time=enabled
           bss-color-partial=enabled
           mesh backhaul=disabled
           local_auth=disabled standalone=disabled nat_mode=disabled

FortiAP-432FR # cw_diag -c acs | grep "last seen"
AC last seen time:  0 SSID cnt 0,0 ap reboot wait time 300,0 16:51
```

The `cw_diag -c acs` command output shows the AP reboot wait time as hh+1:mm+1. The 00:00 value is used to indicate that the reboot time is not configured, not that the reboot time is set to 00:00.

3. Verify that FAP2 is managed by FortiGate and has an SSID assigned with local-standalone enabled:

```
FortiAP-831F # wcfg | grep fsm
    fsm-state            : RUN 693
FortiAP-831F # vcfg
-------------------------------VAP Configuration    1----------------------------
Radio Id  1 WLAN Id  0 test_standalone ADMIN_UP(INTF_UP) init_done 0.0.0.0/0.0.0.0
unknown (-1)
           vlanid=0, intf=wlan10, vap=0x321c902c, bssid=e8:ed:d6:b8:02:f8
           11ax high-efficiency=enabled target-wake-time=enabled
           bss-color-partial=enabled
           mesh backhaul=disabled
```

```
            local_auth=enabled standalone=enabled nat_mode=disabled
FortiAP-831F # cw_diag -c acs | grep "last seen"
AC last seen time:  0 SSID cnt 1,1 ap reboot wait time 300,0 16:51
```

4. When the FortiAPs are disconnected from the FortiGate, they will reboot at the configured time.
   - The FortiAP with no standalone SSID (FAP1) reboots at the time interval configured in interval1 (5 minutes or 300 seconds).

     ```
     FortiAP-432FR # 03901.181 *****cwFwctlReboot:*****
     03901.181     SSID_CNT 1,0. No AC is found in 309 sec (> 300) Rebooting...
     [ 4134.665936] reboot: Restarting system
     ```

   - The FortiAP with standalone SSID (FAP2) reboots at the time configured in wait-time (15:50).

     ```
     FortiAP-831F # date
     Fri Nov 17 15:50:10 GMT 2023
     FortiAP-831F # 01140.026 *****cwFwctlReboot:*****
     01140.026     SSID_CNT 1,1. No AC is found in 177 sec (15:50) Rebooting...
     [ 1195.218481] reboot: Restarting system
     ```

Support individual control of 802.11k and 802.11v protocols - 7.4.2

When upgrading from FOS 7.4.1 to 7.4.2, the configurations made under `set voice-enterprise` will be kept the same. If `voice-enterprise` was enabled, then `set 80211k` and `set 80211v` will both be enabled.

In earlier FOS versions, 802.11k and 802.11v protocol were jointly controlled via the 'voice-enterprise' option. This release allows 802.11k and 802.11v protocols to be individually enabled and disabled. Network administrators can enable 802.11k if they want clients to connect to APs with the best signal, or enable 802.11v to let clients connect to APs with less traffic.

The following CLI commands have been added to manage the 802.11k and 802.11v protocol:

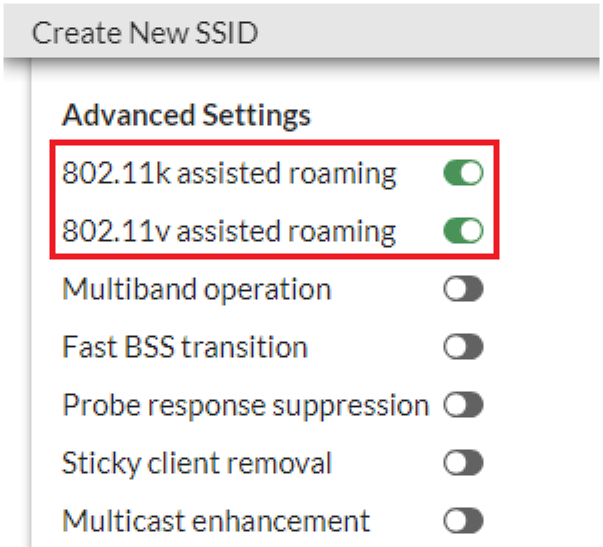
```
config wireless-controller vap
  edit <name>
    set 80211k {Enable | disable}
    set 80211v {Enable | disable}
  next
end
```

`80211k`	Enable/disable 802.11k assisted roaming (default = enable). When 802.11k is enabled, APs provide clients with a list of other neighboring APs and a site report, passively assisting roaming clients in deciding which APs to connect to.
`80211v`	Enable/disable 802.11v assisted roaming (default = enable). When 802.11v is enabled, APs help clients choose the least congested AP by actively sending deauthentication frames to clients that try to connect to congested APs when other APs have better RSSI.

To configure 802.11k and 802.11v protocols - GUI

1. From the FortiOS GUI, go to *System* > *Feature Visibility*.
2. Under the *Additional Features* column, locate and enable *Advanced Wireless Features*.

3. Click *Apply*.
4. Go to *WiFi & Switch Controller > SSIDs* and select the SSID you want to configure.
5. Under *Advanced Settings*, enable the *802.11k* and *802.11v* protocols.



6. When you are finished, click *OK*.

To configure 802.11k and 802.11v protocols - CLI

1. Enable 801.11k and 802.11v protocols on an SSID:

```
config wireless-controller vap
  edit "test.11kv"
    set ssid "11kv.enable"
    set 80211k enable
    set 80211v enable
  next
end
```

2. On the AP, verify the configuration settings:

```
# vcfg
Radio Id  1 WLAN Id  0 11kv.enable ADMIN_UP(INTF_UP) init_done 0.0.0.0/0.0.0.0 unknown
(-1)
           vlanid=0, intf=wlan10, vap=0x7243258, bssid=e0:23:ff:d8:0b:50
           80211k=enabled 80211v=enabled mbo=disabled
```

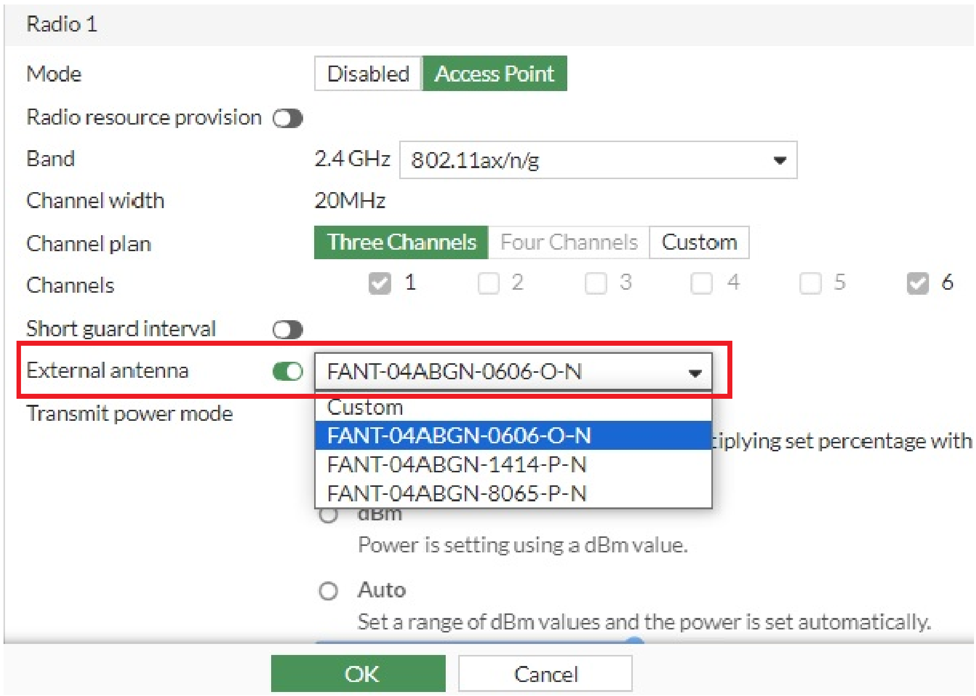
Support external antennas in select FortiAP models - 7.4.2

The release supports installing external antennas on FAP-432F, FAP-433F, FAP-U432F, and FAP-U433F models. Fortinet external antennas can help optimize coverage and overall wireless performance in various installation settings. On supported FortiAP models, you can configure a new FortiAP profile setting and choose from a list of supported Fortinet external antenna models. This setting allows antenna gains specific to the Fortinet external antenna model and the Wi-Fi band (2.4 GHz or 5 GHz) to be taken into consideration by the FortiGate Wireless controller when setting transmit power for a managed FortiAP device.

To see which external antenna and predefined types correspond to which SKU, refer to the Fortinet Antenna Portfolio Data Sheet.

To configure supported external antenna - GUI

1. Go to *WiFi and Switch Controller > FortiAP Profiles* and select *Create New*.
2. From *Platform*, select a FortiAP model that supports external antennas.
3. Under the *Radio* section, enable *External antenna* and select the antenna that you want to install.



4. When you are finished, click *OK*.

To configure supported external antenna - CLI

1. Create a FortiAP profile and select a platform that supports external antennas. In `set optional-antenna`, enter the antenna model.

```
config wireless-controller wtp-profile
  edit "FP432F"
    config platform
      set type 432F
    end
    config radio-2
      set optional-antenna FANT-04ABGN-1414-P-N
    end
  next
end
```

2. Verify the settings have been applied:

```
# diagnose wireless-controller wlac -c wtpprof FP432F | grep antenna
    ..
    opt antenna       : FANT_04ABGN_1414_P_N
```

3. From the FortiAP CLI, check that antenna configurations have been applied:

```
FortiAP-432F # rcfg
… …
Radio 1: AP
   country         : cfg=US oper=US
   countryID       : cfg=841 oper=841
   802.11d enable   : enabled
   sta info        : 0/0
   radio type      : 11AX_5G
   mimo,chainmask : 4, 0xf0 (mimo) 0xf0 (power) 0xf0/0xf0 (oper)
   airtime fairness    : disabled
   ps optimize     : 0
   tx optimize     : f
   11g prot mode   : 0
   HT20/40 coext   : 1
   beacon intv     : 100
   opt antenna     : FANT_04ABGN_1414_P_N
   txpwr mode      : set by percentage (100%)
… …
```

Support Hitless Rolling AP upgrade - 7.4.2

This release introduces Hitless Rolling upgrades for FortiAPs. When upgrading FortiAPs, an algorithm considers the reach of neighboring APs and their locations. The APs are then upgraded in staggered process with some APs being immediately upgraded while others continue to provide Wi-Fi service to clients and are placed in a standby queue. Once the SSIDs on the initial upgraded APs are able to serve clients, the APs in the standby queue begin upgrading.

CLI changes

The following CLI commands for configuring Hitless Rolling AP upgrades have been added to both global settings and per-VDOM settings:

Enabling Hitless Rolling Upgrade at the global level

```
config wireless-controller global
  set rolling-wtp-upgrade {Enable | disable}
  set rolling-wtp-upgrade-threshold <integer>
end
```

`rolling-wtp-upgrade`	Enable/disable rolling WTP upgrade (default = disable). **Note:** Enabling this at the global-level will enforce all managed FortiAPs in all VDOMs to implement the rolling upgrade, regardless of the VDOM-level settings.
`rolling-wtp-upgrade-threshold`	Minimum signal level/threshold in dBm required for the managed WTP to be included in rolling WTP upgrade (-95 to -20, default = -80).

Enabling Hitless Rolling Upgrade at the per-VDOM level

```
config wireless-controller setting
  set rolling-wtp-upgrade {Enable | disable}
```

```
end
```

`rolling-wtp-upgrade`	Enable/disable rolling WTP upgrade (default = disable). **Note:** Enabling this at the VDOM-level will let managed FortiAPs in the current VDOM to implement the rolling upgrade, regardless of the global-level setting.

Executing Hitless Rolling Upgrade

```
exec wireless-controller rolling-wtp-upgrade <all>|<SN>|<wtp-group>
```

`rolling-wtp-upgrade`	Select which APs you want to upgrade with the Hitless Rolling upgrade. You can select all APs, by their WTP serial number, or WTP group.

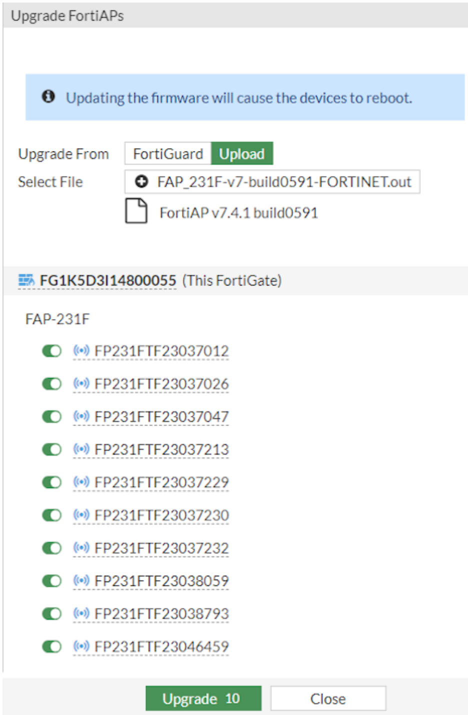
To configure Hitless Rolling AP upgrade - GUI

1. Before you can run Hitless Rolling AP upgrade from the GUI, you must first enable `rolling-wtp-upgrade` and configure the `rolling-wtp-upgrade-threshold` level in the CLI.

   ```
   config wireless-controller global
     set rolling-wtp-upgrade enable
     set rolling-wtp-upgrade-threshold -70
   end
   ```

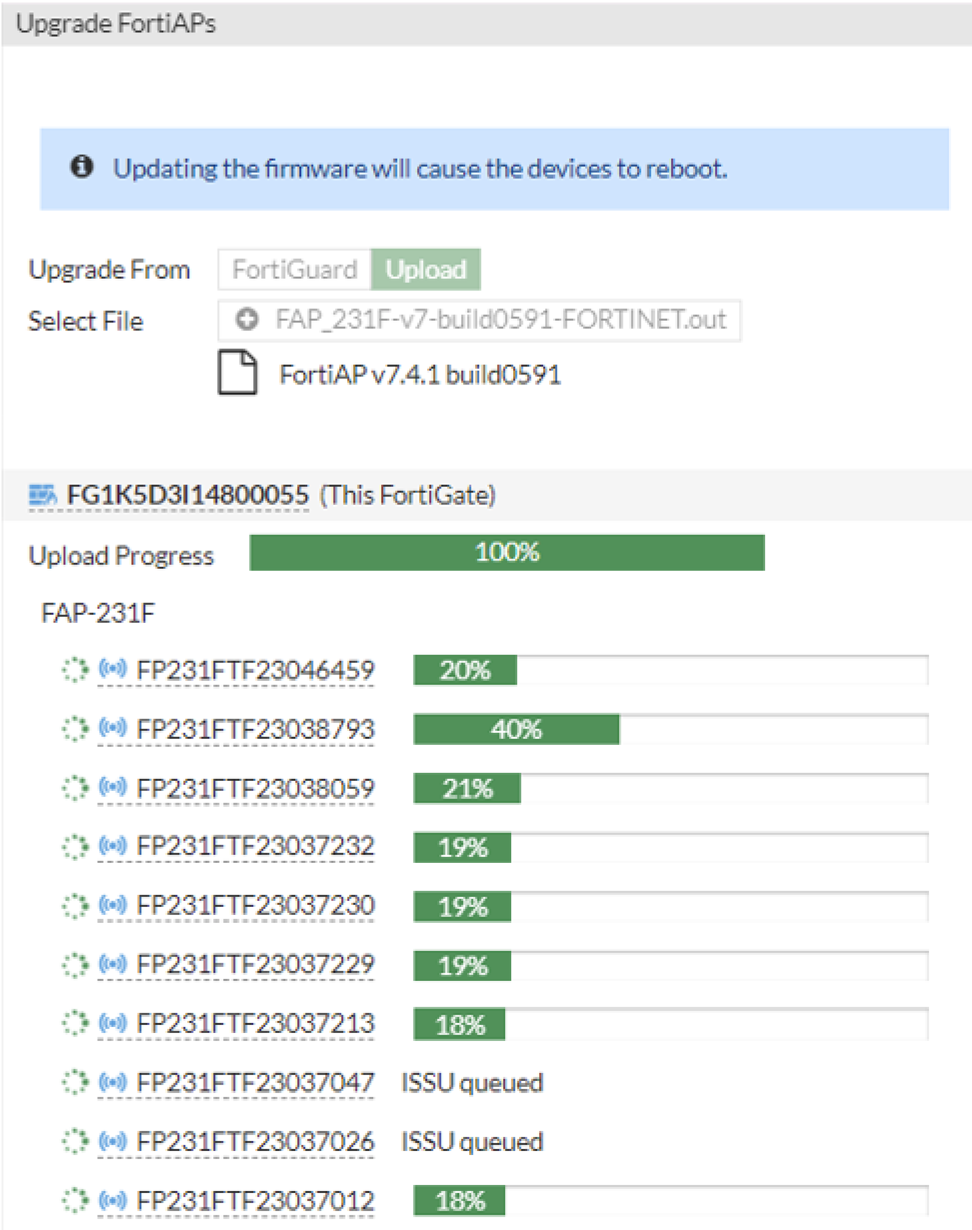
   ```
   config wireless-controller setting
     set rolling-wtp-upgrade enable
   end
   ```

2. From the FortiGate GUI, go to *WiFi & Switch Controller > Managed FortiAPs*.
3. Select multiple FortiAPs of the same model, and then right-click and select *Upgrade*.
   The *Upgrade FortiAPs* window loads.
4. Upload the FortiAP image file and click *Upgrade*.



The FortiAPs are automatically upgraded using the Hitless Rolling upgrade method.

5. Some FortiAPs immediately begin upgrading while others are marked with "ISSU queued". In-Service Software Upgrade (ISSU) indicates that these are the standby APs that continue to provide Wi-Fi service to clients and are queued to be upgraded later.



6. Once the first batch of FortiAPs are upgraded and can provide service, the ISSU queued FortiAPs will begin upgrading.

To configure Hitless Rolling AP upgrade - CLI

1. Enable `rolling-wtp-upgrade` at either the global or VDOM level and configure the `rolling-wtp-upgrade-threshold` level.

```
config wireless-controller global
  set rolling-wtp-upgrade enable
  set rolling-wtp-upgrade-threshold -70
end
```

```
config wireless-controller setting
  set rolling-wtp-upgrade enable
end
```

2. Upload FortiAP images to FortiGate and check the image list. In this example, FAP231F is uploaded:

```
execute wireless-controller upload-wtp-image tftp /FortiAP/v7.00/images/build0626/FAP_
231F-v7-build0626-FORTINET.out 172.18.52.254
```

3. Verify the uploaded FortiAP images:

```
execute wireless-controller list-wtp-image
WTP Images on AC:
ImageName                              ImageSize(B)   ImageInfo             ImageMTime
…
FP231F-v7.4.2-build0626-IMG.wtp        37605058       FP231F-v7.4-build0626  Mon Nov 27
10:39:53 2023
```

4. Run the Rolling WTP Upgrade and prepare to check the FortiAP upgrade status.

```
exec wireless-controller rolling-wtp-upgrade all
```

5. Promptly check the FortiAP upgrade status to verify that the APs are upgrading:

```
diagnose wireless-controller wlac -c ap-upd

1,50,66 0-FP231FTF23037012 FP231F-v7.4-build0591 ==> FP231F-v7.4-build0626 ws (0-
10.233.10.7:5246) upd-download,3 5%           <- The image download has started (may
still be blocked by concurrent AP image downloading limit)
2,50,66 0-FP231FTF23037026 FP231F-v7.4-build0591 ==> FP231F-v7.4-build0626 ws (0-
10.233.10.3:5246) upd-download,3 6%
3,50,66 0-FP231FTF23037047 FP231F-v7.4-build0591 ==> FP231F-v7.4-build0626 ws (0-
10.233.10.24:5246) upd-download,3 6%
…
15,50,66 0-FP431FTF23000559 FP431F-v7.4-build0591 ==> FP431F-v7.4-build0626 ws (0-
10.233.30.40:5246) upd-enqueue-issu,4 0%      <- In queue for rolling AP upgrade to
avoid Wi-Fi service drop
16,50,66 0-FP431FTF23021146 FP431F-v7.4-build0591 ==> FP431F-v7.4-build0626 ws (0-
10.233.30.42:5246) upd-enqueue-issu,4 0%
…
19,50,66 0-FP433FTF21001215 FP433F-v7.4-build0591 ==> FP433F-v7.4-build0626 ws (0-
10.233.30.41:5246) upd-enqueue-issu,4 0%
…
```

6. After a few minutes, check the FortiAP upgrade status again to see any changes:

```
diagnose wireless-controller wlac -c ap-upd

1,44,66 0-FP231FTF23037012 FP231F-v7.4-build0626 ws (0-10.233.10.7:5246) upd-ap-up,58
  <- The AP has reconnected after image upgrade
…
7,44,66 0-FP231FTF23037232 FP231F-v7.4-build0626 ws (0-10.233.10.36:5246) upd-ssid-up,5
  <- The AP's SSIDs are UP after image upgrade
…
15,44,66 0-FP431FTF23000559 FP431F-v7.4-build0591 ==> FP431F-v7.4-build0626 ws (0-
10.233.30.40:5246) upd-enqueue-issu,404 0%      <- Still in queue for rolling AP upgrade
to avoid Wi-Fi service drop
16,44,66 0-FP431FTF23021146 FP431F-v7.4-build0591 ==> FP431F-v7.4-build0626 ws (0-
10.233.30.42:5246) upd-enqueue-issu,404 0%
```

```
…
19,44,66 0-FP433FTF21001215 FP433F-v7.4-build0591 ==> FP433F-v7.4-build0626 ws (0-
10.233.30.41:5246) upd-enqueue-issu,404 0%
…
```

7. After a few more minutes, check the FortiAP upgrade status again to see APs in the queue begin upgrading:

```
diagnose wireless-controller wlac -c ap-upd

1,48,66 0-FP231FTF23037012 FP231F-v7.4-build0626 ws (0-10.233.10.7:5246) upd-ssid-up,6
…
15,48,66 0-FP431FTF23000559 FP431F-v7.4-build0591 ==> FP431F-v7.4-build0626 ws (0-
10.233.30.40:5246) upd-download,12 48%      <- Previously queued APs have begun the
upgrade process since enough SSIDs from other APs are up to provide service
16,48,66 0-FP431FTF23021146 FP431F-v7.4-build0591 ==> FP431F-v7.4-build0626 ws (0-
10.233.30.42:5246) upd-download,12 49%
…
19,48,66 0-FP433FTF21001215 FP433F-v7.4-build0591 ==> FP433F-v7.4-build0626 ws (0-
10.233.30.41:5246) upd-download,12 47%
…
```

Support third-party antennas in select FortiAP models - 7.4.2

The release supports installing third-party antennas on select FortiAP models and customizing their antenna gain. On FortiAP models that support third-party antennas, you can enable the FortiAP profile external antenna setting and customize the antenna gain in dB. Third-party antennas can help optimize coverage and overall wireless performance in various installation settings.

The following table shows which FortiAP models support third-party antennas:

FortiAP F models	FAP-432F
	FAP-432FR
	FAP-433F
FortiAP G models	FAP-233G
	FAP-432G
	FAP-433G
FortiAP-U F models	FAP-U432F
	FAP-U433F

The following CLI commands have been added to configure third-party antenna parameters:

```
config wireless-controller wtp-profile
  edit <name>
    config platform
      set type [432F|432FR|...]
    end
    config radio-2
      set optional-antenna [none|custom|FANT-04ABGN-0606-O-R|...]
      set optional-antenna-gain {integer}
    end
```

```
  next
end
```

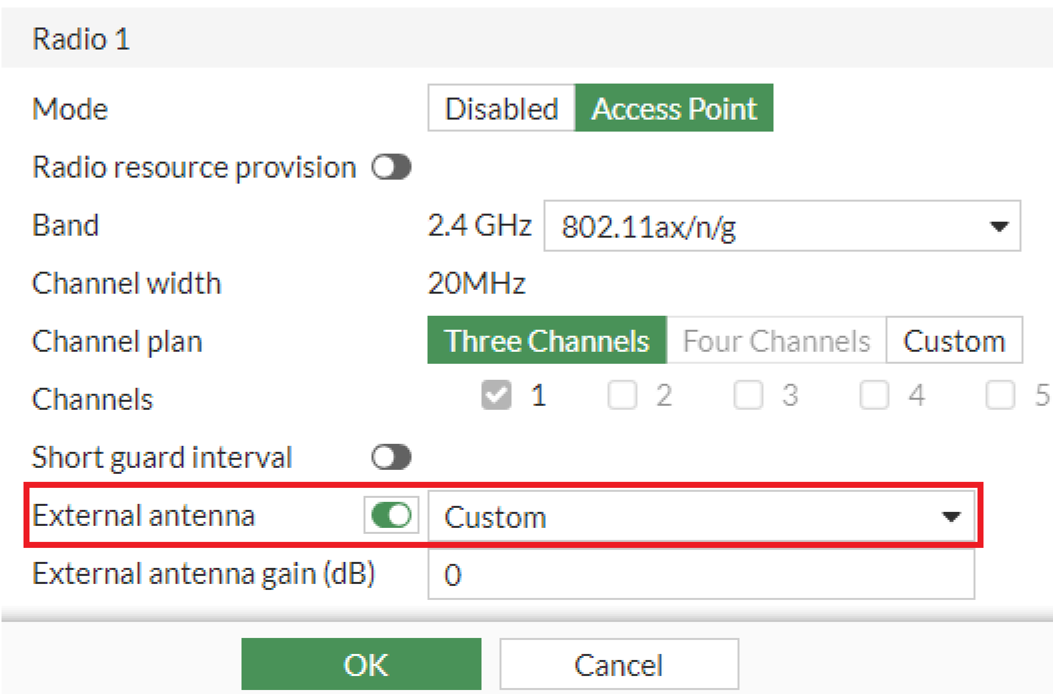
`set optional-antenna`	Set which optional antenna you want to use on the FAP (default = none).
`set optional-antenna-gain`	Optional antenna gain in dBi (0 to 20, default = 0).

Antenna gain values in dBi configurable for your antenna should remain within regulatory EIRP limits.

Please consult your external antenna documentation and regulatory authority standards for details.

To configure FortiAP to use third-party antennas - GUI

1. Go to *WiFi and Switch Controller > FortiAP Profiles* and select *Create New*.
2. In *Platform*, select a FortiAP model that supports third-party antennas.
3. Under the *Radio* section, enable *External antenna* and select *Custom*.



4. In *External antenna gain (dB)*, configure a value between 0 to 20.
5. When you are finished, click *OK*.

To configure FortiAP to use third-party antennas - CLI

1. Create a FortiAP profile and select a platform that supports third-party antennas.
   Set `optional-antenna` to `custom` and configure an `optional-antenna-gain` value between 0 to 20.

```
config wireless-controller wtp-profile
  edit "FP433G"
    config platform
```

```
        set type 433G
      end
      config radio-2
        set optional-antenna custom
        set optional-antenna-gain "10"
      end
    next
end
```

2. Verify the settings have been applied:

```
# diagnose wireless-controller wlac -c wtpprof FP433G | grep antenna
    opt antenna      : Custom
    opt antenna gain : 10
```

3. From the FortiAP CLI, check that antenna configurations have been applied:

```
FortiAP-433G # rcfg
… …
Radio 1: AP
   country         : cfg=US oper=US
   countryID       : cfg=841 oper=841
   802.11d enable   : enabled
   sta info        : 0/0
   radio type      : 11AX_5G
   mimo,chainmask : 4, 0xf0 (mimo) 0xf0 (power) 0xf0/0xf0 (oper)
   airtime fairness   : disabled
   ps optimize     : 0
   tx optimize     : f
   11g prot mode   : 0
   HT20/40 coext   : 1
   beacon intv     : 100
   opt antenna     : Custom
   opt ant gain    : 10
… …
```

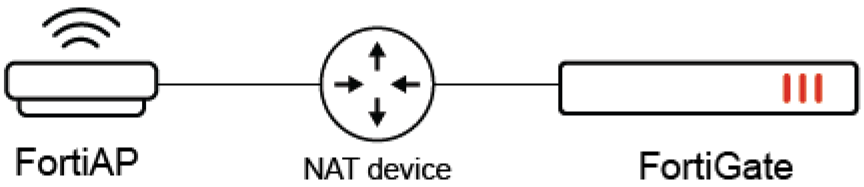
Improve CAPWAP stability over NAT - 7.4.2

This release improves CAPWAP stability for FortiAPs that are managed by a FortiGate behind a Network Address Translation (NAT) device. This enhancement enables users to customize the interval at which keep-alive messages are sent from FortiAPs to their managing FortiGate, keeping the NAT session alive and ensuring a consistent connection. Once the keep-alive message is sent, FortiAPs will not disconnect from the FortiGate even if there is a session timeout configured on the NAT device. This enhances network reliability and minimizes downtime caused by unstable NAT device networks.

The following CLI command has been added:

```
config wireless-controller timers
  set nat-session-keep-alive <integer>
end
```

`set nat-session-keep-alive`	Maximal time in seconds between control requests sent by the managed WTP, AP, or FortiAP (0 - 255 seconds, default = 0).



To configure NAT session keep-alive message - CLI

1. Configure the interval at which NAT session keep-alive messages are sent in seconds.

```
config wireless-controller timers
  set nat-session-keep-alive 10
end
```

2. Verify the configurations on the FortiAP.

```
FortiAP-231F # cw_diag -c acs
WTP Configuration
    name                 : FortiAP-231F
    loc                  : N/A
    ap mode              : thin AP
    led state            : enable
    PWR LED state        : GREEN     REASON: ACS 0 changed in DATA_CHECK state.
    poe mode cal         : full
    poe mode oper        : full
    allowaccess          :
    lldp enable          : enable
    extension info enable: enable
    radio cnt            : 3
    sta info             : 0/0
    echo-interval        : 30
    nat-sess-keep-alive  : 10
    keep-alive-interval  : 30
...
```

From the cwWtpd deamon output, you can see that a `FTNT_WTP_NOTIF` message is sent every 10 seconds to keep the connection alive if there is no `ECHO_REQ` sent. The timer of `FTNT_WTP_NOTIF` is 10 seconds while the timer of `ECHO_REQ` is 30 seconds.

```
[12/5/2023 7:17:46 PM] 15290.608 AC0    msgType     : 3163149 FTNT_WTP_NOTIF     0
10.40.49.58:5247
[12/5/2023 7:17:56 PM] 15300.609 AC0    msgType     : 3163149 FTNT_WTP_NOTIF     0
10.40.49.58:5247
[12/5/2023 7:18:02 PM] 15306.680 AC0    msgType     : 13 ECHO_REQ                163
10.40.49.58:5247
[12/5/2023 7:18:12 PM] 15316.608 AC0    msgType     : 3163149 FTNT_WTP_NOTIF     0
10.40.49.58:5247
[12/5/2023 7:18:22 PM] 15326.609 AC0    msgType     : 3163149 FTNT_WTP_NOTIF     0
10.40.49.58:5247
[12/5/2023 7:18:32 PM] 15336.608 AC0    msgType     : 3163149 FTNT_WTP_NOTIF     0
10.40.49.58:5247
[12/5/2023 7:18:32 PM] 15336.677 AC0    msgType     : 13 ECHO_REQ                164
10.40.49.58:5247
[12/5/2023 7:18:46 PM] 15350.609 AC0    msgType     : 3163149 FTNT_WTP_NOTIF     0
10.40.49.58:5247
```

Switch controller

This section includes information about switch-controller-related new features:

Specify FortiSwitch names to use in switch-controller CLI commands

You can now use names for managed FortiSwitch units in switch-controller CLI commands. The user-defined name is also used in the FortiOS GUI and logs. The FortiSwitch unit's serial number is saved in a new read-only field.

Follow these rules for defining a managed FortiSwitch name:

- The name can be a maximum of 16 characters in length.
- Use numbers (0-9), letters (a-z and A-Z), dashes, and underscores for the managed FortiSwitch name.

When you upgrade from FortiOS 7.4.0, the FortiSwitch unit's serial number is used as the managed FortiSwitch name if a managed FortiSwitch name has not been defined. If you downgrade from FortiOS 7.4.0 to FortiOS 6.4.x, the managed FortiSwitch name is changed to the FortiSwitch unit's serial number.

Using the GUI

1. Go to *WiFi & Switch Controller > Managed FortiSwitches*.
2. Select an unauthorized FortiSwitch unit and then click *Edit*.
3. In the *Name* field, enter a name for the managed FortiSwitch unit.
4. Click *OK* to save the new name.

Using the CLI

```
config switch-controller managed-switch
   rename <FortiSwitch_serial_number> to <managed_FortiSwitch_name>
end
```

For example:

```
config switch-controller managed-switch
```

```
   rename S524DN4K16000116 to Distribution
end
```

Other CLI changes

When you pre-configure a managed switch, you must use the new `set sn` command under `config switch-controller managed-switch` to store the FortiSwitch serial number. For example:

```
config switch-controller managed-switch
   edit switch1
      set sn S524DNTV21000212
      set fsw-wan1-peer fortilink
      set fsw-wan1-admin enable
   next
end
```

The `execute switch-controller get-sync-status switch-id <managed_FortiSwitch_name>` command uses the user-defined switch name, and the `execute switch-controller get-sync-status serial <FortiSwitch_serial_number>` command uses the FortiSwitch serial number. For example:

- `execute switch-controller get-sync-status serial S524DN4K16000116`
- `execute switch-controller get-sync-status switch-id Racktray-127`

There is a new `set isl-peer-device-sn` command under `config switch-controller managed-switch` to store the serial number of the ISL peer device. For example:

```
config switch-controller managed-switch
   edit Distribution
      config ports
         edit port2
            set isl-local-trunk-name isltrunk1
            set isl-peer-port-name port23
            set isl-peer-device-name islpeerswitch
            set isl-peer-device-sn S124EN5918003682
         next
      end
   next
end
```

The following switch-controller CLI commands now use the user-defined FortiSwitch name:

- `diagnose switch-controller trigger config-sync <managed_FortiSwitch_name>`
- `execute switch-controller get-conn-status`
- `execute switch-controller get-physical-conn standard <port_name>`
- `execute switch-controller get-sync-status all`
- `execute switch-controller get-upgrade-status`

Support user-configurable ACL

You can now use an access control list (ACL) to configure a policy for the ingress stage of the pipeline for incoming traffic. After creating an ACL group for the ingress policy, you apply the ACL group to a managed switch port.

A user-configurable ACL might conflict with or be overridden by an ACL implemented by other managed FortiSwitch features. If a user-configurable ACL and an internal ACL do not conflict, the resulting behavior depends on the FortiSwitch model. Fortinet recommends validating user-configurable ACLs to make certain that they operate correctly with other enabled features.

To use an ACL:

1. Create an ACL ingress policy.
2. Create an ACL group and add the ingress policy to it.
3. Apply the ACL group to a managed switch port.
4. View the counters on page 564.

Create an ACL ingress policy

The ACL ingress policy includes the following key attributes:

- *Interface*—The port on which traffic arrives at the switch. The policy applies to ingress traffic only (not egress traffic).
- *Classifier*—The classifier identifies the packets that the policy will act on. Each packet can be classified based on one or more criteria. The supported criteria are source and destination MAC address, VLAN identifier, and source and destination IP address.
- *Actions*—If a packet matches the classifier criteria for a given ACL, the following types of action can be applied to the packet:
  - Allow or block the packet
  - Count the number of ingress packets

The switch uses specialized TCAM memory to perform ACL matching.

The order of the classifiers provided during group creation (or during an ACL update in a group when new classifiers are added) matter. Hardware resources are allocated as best fit at the time of creation, which can cause some fragmentation and segmentation of hardware resources because not all classifiers are available at all times. Because the availability of classifiers is order dependent, some allocations succeed or fail at different times.

To create an ACL ingress policy in the CLI:

```
config switch-controller acl ingress
   edit <policy_identifier>
      config action
         set count {enable | disable}
         set drop {enable | disable}
      end
      config classifier
         set dst-ip-prefix <IPv4_address> <netmask>
         set dst-mac <destination_MAC_address>
         set src-ip-prefix <IPv4_address> <netmask>
         set src-mac <source_MAC_address>
         set vlan <1-4094>
      end
   next
```

```
end
```

Create an ACL group

An ACL group contains one or more ACLs.

The ACL ingress policies are assigned to ACL group 3 in the managed FortiSwitch unit. If the managed FortiSwitch unit does not support ACL group 3, the user-configurable ACL is not supported.

To create an ACL group in the CLI:

```
config switch-controller acl group
   edit "<ACL_group_name>"
      set ingress <policy_identifier1> <policy_identifier2> ...
   next
end
```

For example:

```
config switch-controller acl group
   edit "ACLgroup1"
      set ingress 2 3 4
   next
end
```

Apply the ACL group to a managed switch port

You can apply one or more ACL groups to a managed switch port.

To apply an ACL group to a managed switch port in the CLI:

```
config switch-controller managed-switch
   edit <FortiSwitch_serial_number>
      config ports
         edit <managed_switch_port_name>
            set acl-group "<ACL_group_name1> <ACL_group_name2> ..."
         next
      end
   next
end
```

For example:

```
config switch-controller managed-switch
   edit FS1D243Z14000016
      config ports
         edit port10
            set acl-group "ACLgroup1 ACLgroup2 ACLgroup3"
         next
      end
   next
end
```

View the counters

On the 4xxE, 1xxE, and 1xxF platforms, the ACL byte counters are not available (they will always show as 0 on the CLI). The packet counters are available.

You can use the CLI to view the counters associated with the ingress policies.

To view the counters in the CLI:

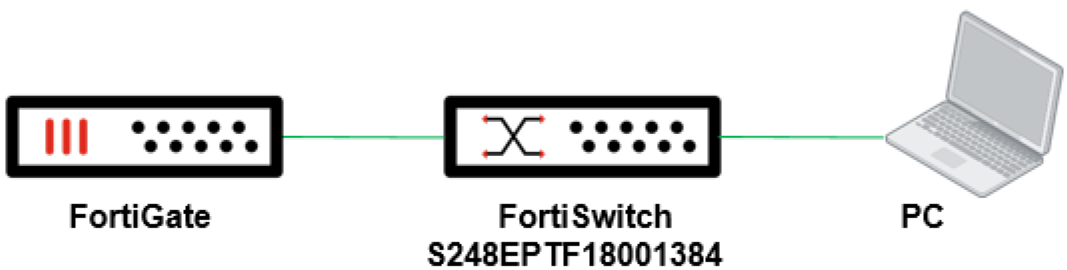
```
diagnose switch-controller switch-info acl-counters <FortiSwitch_serial_number>
```

For example:

```
diagnose switch-controller switch-info acl-counters FS1D243Z14000016
```

Configuration example

In the following example, the ingress ACL policy prevents a PC connected to S248EPTF18001384 (which is managed by a FortiGate device) from accessing `8.8.8.8 255.255.255.255`.




```
config switch-controller acl ingress
   edit 1
      config action
         set drop enable
      end
      config classifier
         set dst-ip-prefix 8.8.8.8 255.255.255.255
         set src-mac 00:0c:29:d4:4f:3c
      end
   next
end

config switch-controller acl group
   edit "group1"
      set ingress 1
   next
end

config switch-controller managed-switch
   edit "S248EPTF18001384"
      config ports
         edit "port6"
            set acl-group "group1"
```

```
        next
      end
    next
end
```

Support configuring DHCP-snooping option-82 settings

This feature requires FortiSwitchOS 7.2.2 or later.

You can now include option-82 data in the DHCP request for DHCP snooping. DHCP option-82 data provides additional security by enabling a controller to act as a DHCP relay agent to prevent DHCP client requests from untrusted sources. You can select a fixed format (`set dhcp-option82-format legacy`) for the Circuit ID and Remote ID fields or select which values appear in the Circuit ID and Remote ID fields (`set dhcp-option82-format ascii`).

The following is the fixed format for the option-82 Circuit ID field:

```
hostname-[<vlan:16><mod:8><port:8>].32bit
```

The following is the fixed format for the option-82 Remote ID field:

```
[mac(0..6)].48bit
```

If you want to select which values appear in the Circuit ID and Remote ID fields:

- For the Circuit ID field, you can include the interface name, VLAN name, host name, mode, and description.
- For the Remote ID field, you can include the MAC address, host name, and IP address.

You can specify whether the DHCP-snooping client only broadcasts packets on trusted ports in the VLAN (`set dhcp-snoop-client-req drop-untrusted`) or broadcasts packets on all ports in the VLAN (`set dhcp-snoop-client-req forward-untrusted`).

You can set a limit for how many entries are in the DHCP-snooping binding database for each port with the `set dhcp-snoop-db-per-port-learn-limit` command. By default, the number of entries is 64. The range of values depends on the switch model.

Before configuring the learning limit, check the range for your switch model by typing `set dhcp-snoop-db-per-port-learn-limit ?`.

You can also specify how long entries are kept in the DHCP-snooping server database with the `set dhcp-snoop-client-db-exp` command. By default, the entries are kept for 86,400 seconds. The range of values is 300-259,200 seconds.

If you have included option-82 data in the DHCP request, it applies globally. You can override the global option-82 setting to specify plain text strings for the Circuit ID field and the Remote ID field for a specific VLAN on a port. If `dhcp-snoop-option82-override` is not configured for the incoming VLAN and switch interface, the settings for the Circuit ID and Remote ID fields are taken from the global option-82 configuration.

NOTE: The values for the Circuit ID and Remote ID field are either both taken from the global option-82 configuration or both taken from the `dhcp-snoop-option82-override` settings. The system cannot take one value at the global level and the other value from the override settings.

Each plain text string can be a maximum of 256 characters long. Together, the combined length of both plain text strings can be a maximum of 256 characters long.

NOTE: You can override the option-82 settings for DHCP snooping but not for DHCP relay.

To configure the option-82 data on a global level:

```
config switch-controller global
   set dhcp-option82-format {ascii | legacy}
   set dhcp-option82-circuit-id {intfname <interface_name> | vlan <VLAN_name> | hostname
         <host_name> | mode <mode> | description <string>}
   set dhcp-option82-remote-id {mac <MAC_address> | hostname <host_name> | ip <IP_address>}
   set dhcp-snoop-client-req {drop-untrusted | forward-untrusted}
   set dhcp-snoop-client-db-exp <300-259200>
   set dhcp-snoop-db-per-port-learn-limit <integer>
end
```

To override the option-82 global settings for a specific VLAN on a port:

```
config switch-controller managed-switch
   edit "<FortiSwitch_serial_number>"
      config ports
         edit "<port_name>"
            config dhcp-snoop-option82-override
               edit <VLAN_name>
                  set remode-id <string>
                  set circuit-id <string>
               next
            end
         next
      end
   next
end
```

For example:

```
config switch-controller managed-switch
   edit "S524DF4K15000024"
      config ports
         edit "port10"
            config dhcp-snoop-option82-override
               edit vlan15
                  set remode-id "remote-id test"
                  set circuit-id "circuit-id test"
               next
            end
         next
      end
   next
end
```

Display DHCP-snooping option-82 data

This feature requires FortiSwitchOS 7.2.2 or later. The managed FortiSwitch units must be configured with DHCP-snooping option -82 settings.

You can use the `diagnose switch-controller switch-info option82-mapping snooping` command to display option-82 Circuit ID and Remote ID values in ASCII or hexadecimal format. This command requires the serial number of the managed switch unit and VLAN identifier. Specifying the port name is optional.

To display option-82 Circuit ID and Remote ID values in ASCII format:

```
diagnose switch-controller  switch-info option82-mapping snooping ascii <FortiSwitch_serial_
     number> <VLAN_ID> <port_name>
```

For example:

```
diagnose switch-controller  switch-info option82-mapping snooping ascii S524DN4K16000116
     vlan11 port3
```

To display option-82 Circuit ID and Remote ID values in hexadecimal format:

```
diagnose switch-controller  switch-info option82-mapping snooping hex <FortiSwitch_serial_
     number> <VLAN_ID> <port_name>
```

For example:

```
diagnose switch-controller  switch-info option82-mapping snooping hex S524DN4K16000116
     vlan11 port5
```

Support automatically allowing and blocking intra-VLAN traffic based on FortiLink connectivity - 7.4.1

You can now allow or block intra-VLAN traffic on the managed FortiSwitch units when the connection to the FortiGate device is lost.

To allow or block intra-VLAN traffic when the connection to the FortiGate device is lost:

```
config switch-controller fortilink-settings
   edit "<FortiLink_interface>"
      set access-vlan-mode { legacy | fail-open | fail-close}
   next
end
```

Option	Description
legacy	This is the default. When the connection to the FortiGate device is lost, intra-VLAN traffic on the managed FortiSwitch units is blocked.
fail-open	When the connection to the FortiGate device is lost, intra-VLAN traffic on the managed FortiSwitch units is allowed.

Option	Description
fail-close	When the connection to the FortiGate device is lost, intra-VLAN traffic on the managed FortiSwitch units is blocked.

Support the FortiOS one-arm sniffer on a mirrored VLAN interface - 7.4.1

You can now use the FortiOS one-arm sniffer to configure a VLAN interface on a managed FortiSwitch unit as an intrusion detection system (IDS). Traffic sent to the interface is examined for matches to the configured security profile. The matches are logged, and the unmatched sniffed traffic is not forwarded to the FortiGate device. Sniffing only reports on attacks; it does not deny or influence traffic.

Traffic scanned on the FortiOS one-arm sniffer interface is processed by the CPU. The FortiOS one-arm sniffer might cause higher CPU usage and perform at a lower level than traditional inline scanning.

The absence of high CPU usage does not indicate the absence of packet loss. Packet loss might occur due to the capacity of the TAP devices hitting maximum traffic volume during mirroring or, on the FortiGate device, when the kernel buffer size is exceeded and it is unable to handle bursts of traffic.

To configure the FortiOS one-arm sniffer in the CLI:

1. Specify the managed switch port to use to mirror traffic in RSPAN or ERSPAN mode on page 568.

2. Enable the FortiOS one-arm sniffer on the VLAN interface that will mirror traffic on page 569.

3. Configure the FortiOS one-arm sniffer in a firewall policy on page 569.

4. Generate traffic on the client.

5. Review the logs for the sniffer policy on page 570.

1. Specify the managed switch port to use to mirror traffic in RSPAN or ERSPAN mode

You can mirror traffic in RSPAN or ERSPAN mode on a layer-2 VLAN. Specify which ingress port you want to use for a mirroring source.

```
config switch-controller traffic-sniffer
   set mode {rspan | erspan-auto}
   config target-port
      edit <FortiSwitch_serial_number>
         set in-ports <port_name>
      next
   end
end
```

For example:

```
config switch-controller traffic-sniffer
   set mode rspan
   config target-port
      edit S524DF4K15000024
         set in-ports port6
      next
   end
end
```

2. Enable the FortiOS one-arm sniffer on the VLAN interface that will mirror traffic

After you enable `ips-sniffer-mode`, `switch-controller-access-vlan` and `switch-controller-rspan-mode` are enabled by default, and `switch-controller-traffic-policy` is set to `sniffer` by default.

```
config system interface
   edit <interface_name>
      set ips-sniffer-mode enable
      set switch-controller-access-vlan enable
      set switch-controller-traffic-policy sniffer
      set switch-controller-rspan-mode enable
   next
end
```

For example:

```
config system interface
   edit rspan
      set ips-sniffer-mode enable
      set switch-controller-access-vlan enable
      set switch-controller-traffic-policy sniffer
      set switch-controller-rspan-mode enable
   next
end
```

3. Configure the FortiOS one-arm sniffer in a firewall policy

Specify the same interface that you used in step 2. Enable the security profiles that you want to use and specify the `sniffer-profile` profile for each security profile. By default, all security profiles are disabled.

```
config firewall sniffer
   edit <sniffer_ID>
      set logtraffic {all | utm}
      set interface <interface_name>
      set av-profile-status {enable | disable}
      set av-profile "sniffer-profile"
      set webfilter-profile-status {enable | disable}
      set webfilter-profile "sniffer-profile"
      set application-list-status {enable | disable}
      set application-list "sniffer-profile"
      set ips-sensor-status {enable | disable}
      set ips-sensor "sniffer-profile"
      set file-filter-profile-status {enable | disable}
      set file-filter-profile "sniffer-profile"
   next
end
```

For example:

```
config firewall sniffer
   edit 50
      set logtraffic all
      set interface rspan
      set av-profile-status enable
      set av-profile sniffer-profile
      set webfilter-profile-status enable
      set webfilter-profile sniffer-profile
      set application-list-status enable
```

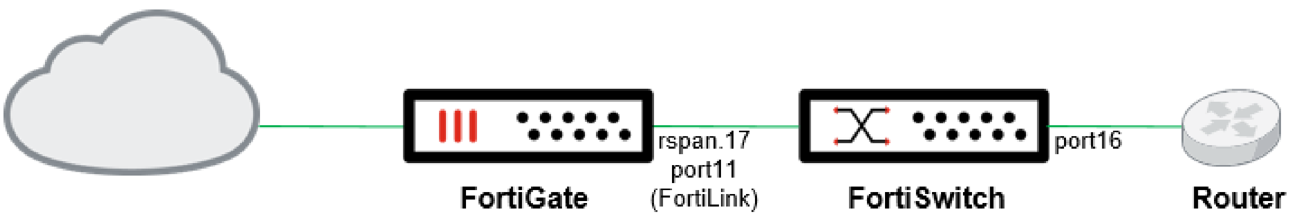
```
        set application-list sniffer-profile
        set ips-sensor-status enable
        set ips-sensor sniffer-profile
        set file-filter-profile-status enable
        set file-filter-profile sniffer-profile
    next
end
```

5. Review the logs for the sniffer policy

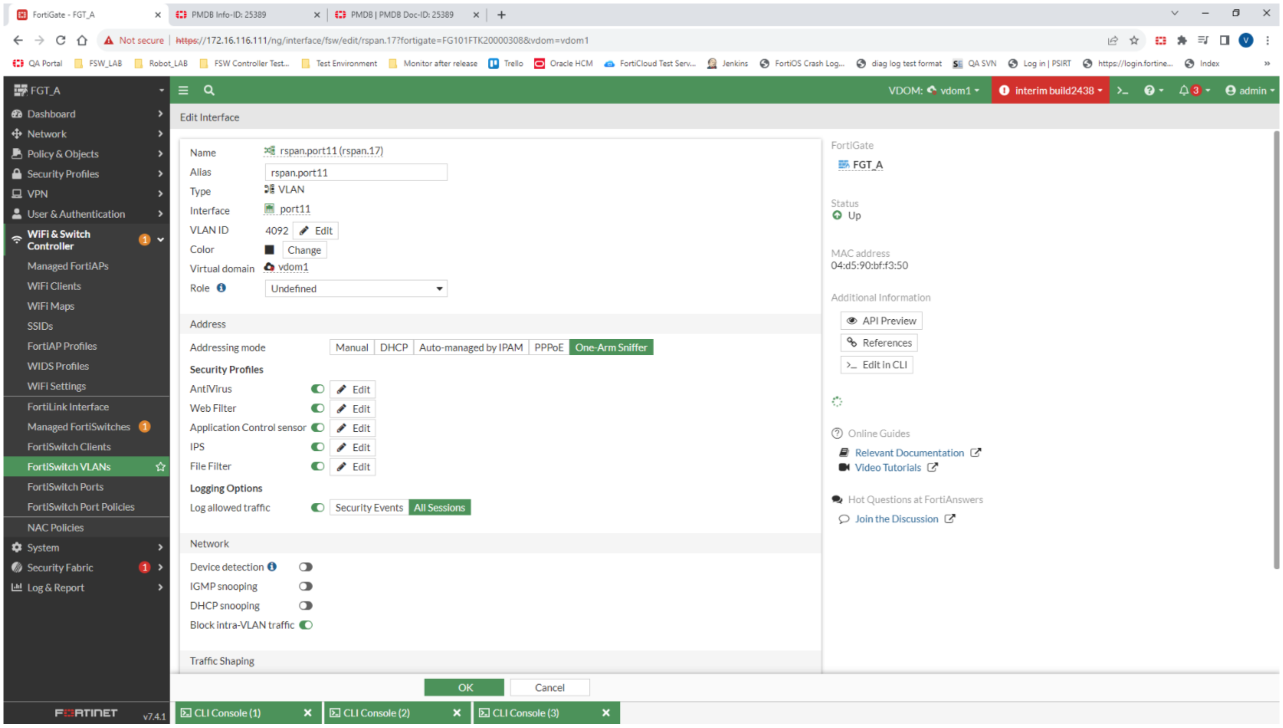
```
execute log display
```

Configuration example

The following example shows how a managed FortiSwitch unit mirrors traffic from a client and then sends the traffic to the FortiGate device for analysis. In this example, enable the FortiOS one-arm sniffer in the FortiOS CLI and then use the FortiOS GUI for the rest of the example.

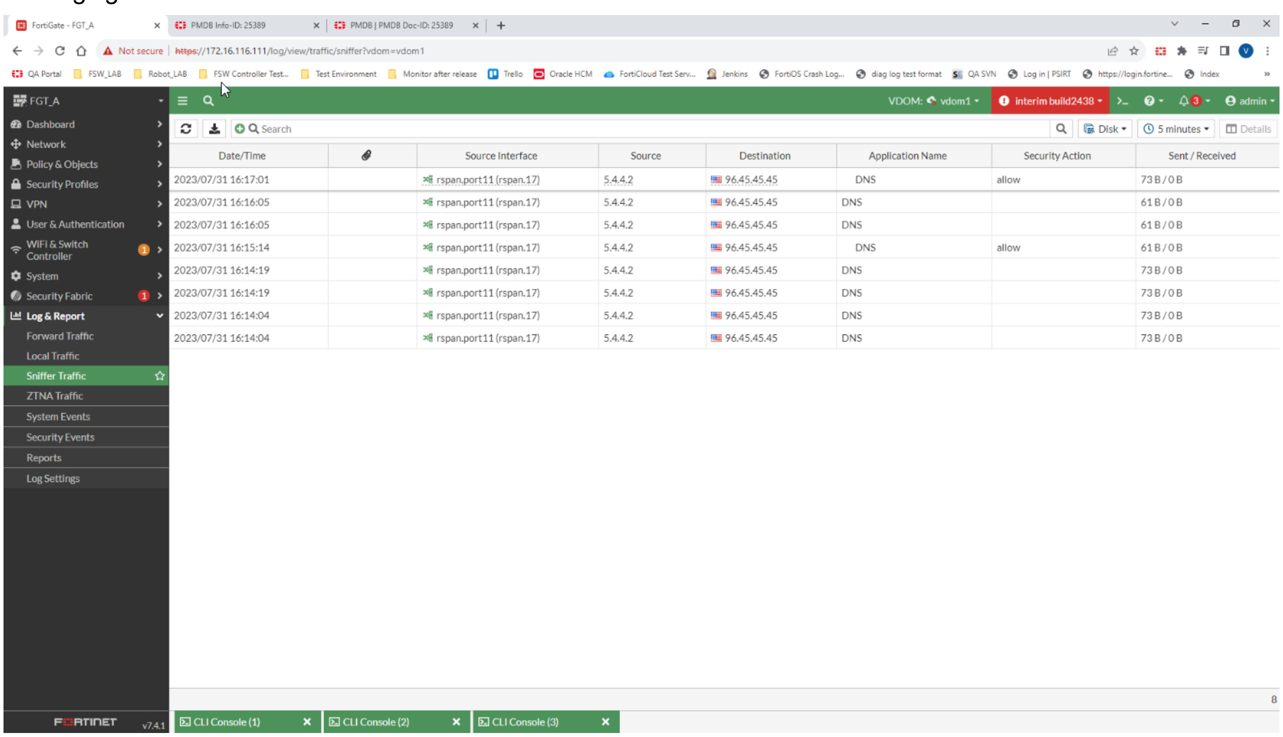


1. Enable the FortiOS one-arm sniffer.
```
config system interface
    edit "rspan.17"
        set ips-sniffer-mode enable
        set vdom root
        set interface port11
        set vlanid 4092
    next
end
```
2. Go to *Network > Interfaces*.
3. Select *rspan.17* (under *port11*) and click *Edit*.
4. Enable the security profiles that you want to use.



5. Click *OK*.
6. Generate traffic on the client.
7. Go to *Log & Report > Sniffer Traffic*.

   The logs generated from the mirrored traffic are listed.



In the FortiOS CLI, use the `execute log display` command to view the logs:

```
784 logs found.
```

```
10 logs returned.
1: date=2023-07-31 time=16:28:13 eventtime=1690846092971957519 tz="-0700"
     logid="0004000017" type="traffic" subtype="sniffer" level="notice" vd="vdom1"
     srcip=5.4.4.2 srcport=51293 srcintf="rspan.17" srcintfrole="undefined"
     dstip=96.45.45.45 dstport=53 dstintf="rspan.17" dstintfrole="undefined"
     srccountry="Germany" dstcountry="United States" sessionid=784 proto=17
     action="accept" policyid=1 policytype="sniffer" service="DNS" trandisp="snat"
     transip=0.0.0.0 transport=0 duration=180 sentbyte=70 rcvdbyte=0 sentpkt=1 rcvdpkt=0
     appid=16195 app="DNS" appcat="Network.Service" apprisk="elevated" utmaction="allow"
     countapp=1 sentdelta=70 rcvddelta=0 mastersrcmac="00:0c:29:38:2a:c6"
     srcmac="00:0c:29:38:2a:c6" srcserver=0 masterdstmac="04:d5:90:bf:f3:50"
     dstmac="04:d5:90:bf:f3:50" dstserver=0
2: date=2023-07-31 time=16:27:39 eventtime=1690846059062169260 tz="-0700"
     logid="0004000017" type="traffic" subtype="sniffer" level="notice" vd="vdom1"
     srcip=5.4.4.2 srcport=37800 srcintf="rspan.17" srcintfrole="undefined"
     dstip=96.45.45.45 dstport=53 dstintf="rspan.17" dstintfrole="undefined"
     srccountry="Germany" dstcountry="United States" sessionid=782 proto=17
     action="accept" policyid=1 policytype="sniffer" service="DNS" trandisp="snat"
     transip=0.0.0.0 transport=0 duration=180 sentbyte=70 rcvdbyte=0 sentpkt=1 rcvdpkt=0
     appid=16195 app="DNS" appcat="Network.Service" apprisk="elevated" utmaction="allow"
     countapp=1 sentdelta=70 rcvddelta=0 mastersrcmac="00:0c:29:38:2a:c6"
     srcmac="00:0c:29:38:2a:c6" srcserver=0 masterdstmac="04:d5:90:bf:f3:50"
     dstmac="04:d5:90:bf:f3:50" dstserver=0 utmref=0-6524
3: date=2023-07-31 time=16:27:39 eventtime=1690846059062027560 tz="-0700"
     logid="0004000017" type="traffic" subtype="sniffer" level="notice" vd="vdom1"
     srcip=5.4.4.2 srcport=52702 srcintf="rspan.17" srcintfrole="undefined"
     dstip=96.45.45.45 dstport=53 dstintf="rspan.17" dstintfrole="undefined"
     srccountry="Germany" dstcountry="United States" sessionid=780 proto=17
     action="accept" policyid=1 policytype="sniffer" service="DNS" trandisp="snat"
     transip=0.0.0.0 transport=0 duration=180 sentbyte=61 rcvdbyte=0 sentpkt=1 rcvdpkt=0
     appid=16195 app="DNS" appcat="Network.Service" apprisk="elevated" utmaction="allow"
     countapp=1 sentdelta=61 rcvddelta=0 mastersrcmac="00:0c:29:38:2a:c6"
     srcmac="00:0c:29:38:2a:c6" srcserver=0 masterdstmac="04:d5:90:bf:f3:50"
     dstmac="04:d5:90:bf:f3:50" dstserver=0 utmref=0-6510
```

Support new commands for Precision Time Protocol configuration - 7.4.1

The CLI commands for configuring Precision Time Protocol (PTP) transparent-clock mode have changed. FortiOS supports the previous CLI commands, as well as the new ones.

Use the following steps to configure PTP transparent-clock mode:

1. Configure a PTP profile or use the `default` profile.
2. Configure the PTP settings.
   By default, PTP is disabled. Enable PTP and select which PTP profile will use these PTP settings. The default profile is automatically selected.
3. Configure the default PTP policy or create a custom PTP policy.
   Select which VLAN will use the PTP policy and the priority of the VLAN. The default PTP policy is applied to all ports. If you want to select which ports to apply the PTP policy to, you need to create a custom PTP policy.
4. If you are not using the default PTP policy, select which port to apply your custom PTP policy to.
   By default, the PTP status is enabled.

To configure a PTP profile:

```
config switch-controller ptp profile
  edit {default | name_of_PTP_profile}
    set description <description_of_PTP_profile>
```

```
      set mode {transparent-e2e | transparent-p2p}
      set ptp-profile C37.238-2017
      set transport l2-mcast
      set domain <0-255> // the default is 254
      set pdelay-req-interval {1sec | 2sec | 4sec | 8sec | 16sec | 32sec} // 1sec default
   next
end
```

For example:

```
config system ptp profile
   edit newPTPprofile
      set description "New PTP profile"
      set mode transparent-p2p
      set ptp-profile C37.238-2017
      set transport l2-mcast
      set domain 1
      set pdelay-req-interval 2sec
   next
end
```

To configure the PTP settings:

```
config switch-controller managed-switch
   edit <FortiSwitch_serial_number>
      set ptp-status {enable | disable} // the default is disable
      set ptp-profile {default | name_of_PTP_profile} // the default is "default"
   next
end
```

For example:

```
config switch-controller managed-switch
   edit S524DF4K15000024
      set ptp-status enable
      set ptp-profile newPTPprofile
   next
end
```

To configure the default PTP policy or create a custom PTP policy:

```
config switch-controller ptp interface-policy
   edit {default | <policy_name>}
      set description <description_of_PTP_policy>
      set vlan <VLAN_name> //no default
      set vlan-pri <0-7> // the default is 4
   next
end
```

For example:

```
config switch-controller ptp interface-policy
   edit ptppolicy1
      set description "New custom PTP policy"
      set vlan vlan10
      set vlan-pri 3
   next
end
```

To apply your custom PTP policy to a port:

```
config switch-controller managed-switch
   edit <FortiSwitch_serial_number>
      config ports
         edit <port_name>
            set ptp-status {enable | disable} // the default is enable
            set ptp-policy {default | <policy_name>} // the default is "default"
         end
      end
```

For example:

```
config switch-controller managed-switch
   edit S524DF4K15000024
      config ports
         edit port5
            set ptp-status enable
            set ptp-policy ptppolicy1
         end
      end
```

Support inter-VLAN routing by managed FortiSwitch units - 7.4.1

Starting in FortOS 7.4.1 with FortiSwitchOS 7.4.1, managed FortiSwitch units can perform inter-VLAN routing. The FortiGate device can program the FortiSwitch unit to do the layer-3 routing of trusted traffic between specific VLANs. In this case, the traffic flows are trusted by the user and do not need to be inspected by the FortiGate device.

Inter-VLAN routing offload is applied to the supported FortiSwitch model located closest to FortiGate device in the topology. Refer to the FortiLink Compatibility table to find which FortiSwitchOS models support this feature.

You can use an MCLAG with inter-VLAN routing.

- If you use an MCLAG, you can have two FortiSwitch units per stack.
  NOTE: To use an MCLAG, you need VRRP, which requires an advanced features license. For more information, refer to Adding a license.
- If you do not use an MCLAG, you can have only one FortiSwitch unit per stack.

To configure inter-VLAN routing:

1. Configure both VLANs for routing offload.
2. Configure the switches for routing offload.

Configure both VLANs for routing offload

By default, `switch-controller-offload` and `switch-controller-offload-gw` are disabled.

The `switch-controller-offload-ip` option is available only when `switch-controller-offload` is enabled.

The `set allowaccess ping` command is configured automatically if it is not already specified.

Enable `switch-controller-offload-gw` on a single VLAN interface. The clients can use the offload IP addresses (configured in the `set switch-controller-offload-ip` command) as the default gateway, which is executed on the FortiSwitch unit. If you are using a DHCP server on the offloaded FortiSwitch VLANs, adjust the DHCP gateway address to match the `switch-controller-offload-ip` address.

```
config system interface
   edit <VLAN_name>
      set ip <IP_address_netmask>
      set switch-controller-offload {enable | disable}
      set switch-controller-offload-ip <IP_address>
      set switch-controller-offload-gw {enable | disable}
   next
end
```

Configure the switches for routing offload

By default, `route-offload` and `route-offload-mclag` are disabled.

When you have an MCLAG configured, you need to enable `route-offload-mclag` and configure `config route-offload`.

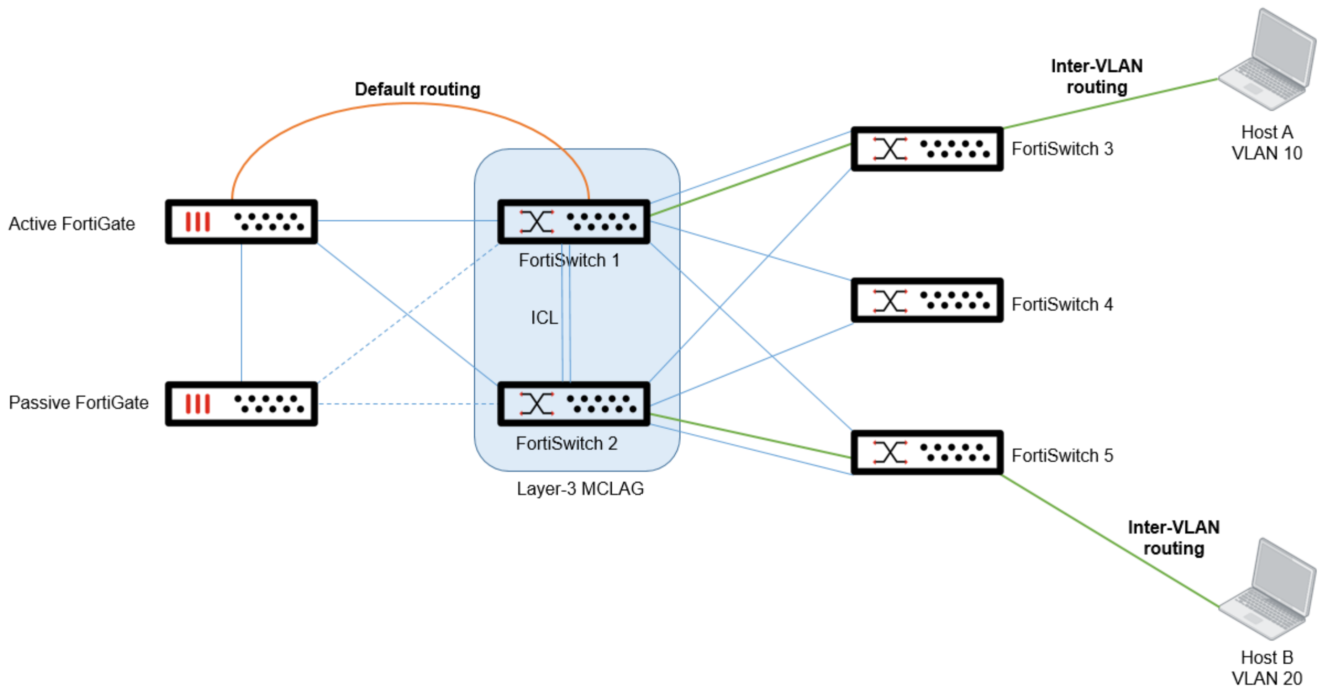
The `config route-offload` commands are available only when `route-offload-mclag` is enabled.

Use `router-ip` to specify the router IP address for VRRP.

```
config switch-controller managed-switch
   edit <FortiSwitch_serial_number>
      set route-offload {enable | disable}
      set route-offload-mclag {enable | disable}
      config route-offload
         edit <VLAN_name_1>
            set router-ip <IP_address_1>
         next
         edit <VLAN_name_2>
            set router-ip <IP_address_2>
         next
      end
   next
end
```

Configuration example

The following example shows how the default routing between Host A and Host B uses the active FortiGate device in HA mode. When inter-VLAN routing is enabled, VLAN10 on Host A routes through FortiSwitch 3, FortiSwitch 1, FortiSwitch 2, and FortiSwitch 5 to VLAN 20 on Host B.



1. Configure both VLANs for routing offloading

```
config system interface
   edit "vlan.10"
      set ip 192.168.10.1/24
      set switch-controller-offload enable
      set switch-controller-offload-ip 192.168.10.2
      set switch-controller-offload-gw enable
   next
   edit "vlan.20"
   set ip 192.168.20.1/24
      set switch-controller-offload enable
      set switch-controller-offload-ip 192.168.20.2
   next
end
```

2. Configure FortiSwitch 1 to route to Host A and Host B. Because this example uses MCLAG, you need to enable `route-offload-mclag` and configure `config route-offload`.

```
config switch-controller managed-switch
   edit ST1E24TF21000347
      set route-offload enable
      set route-offload-mclag enable
      config route-offload
         edit "vlan.10"
            set router-ip 192.168.10.3
         next
         edit "vlan.20"
            set router-ip 192.168.20.3
         next
      end
   next
end
```

3. Configure FortiSwitch 2 to route to route to Host A and Host B. Because this example uses MCLAG, you need to enable `route-offload-mclag` and configure `config route-offload`.

```
config switch-controller managed-switch
    edit ST1E24TF21000408
        set route-offload enable
        set route-offload-mclag enable
        config route-offload
            edit "vlan.10"
                set router-ip 192.168.10.4
            next
            edit "vlan.20"
                set router-ip 192.168.20.4
            next
        end
    next
end
```

Support security rating recommendations for tier-2 and tier-3 MCLAGs - 7.4.1

More tests have been added to the FortiSwitch recommendations to help optimize your network:

- When a connected tier-1 MCLAG peer group is detected and FortiOS detects a possible tier-2 MCLAG pair of switches, FortiOS recommends forming a tier-2 MCLAG.

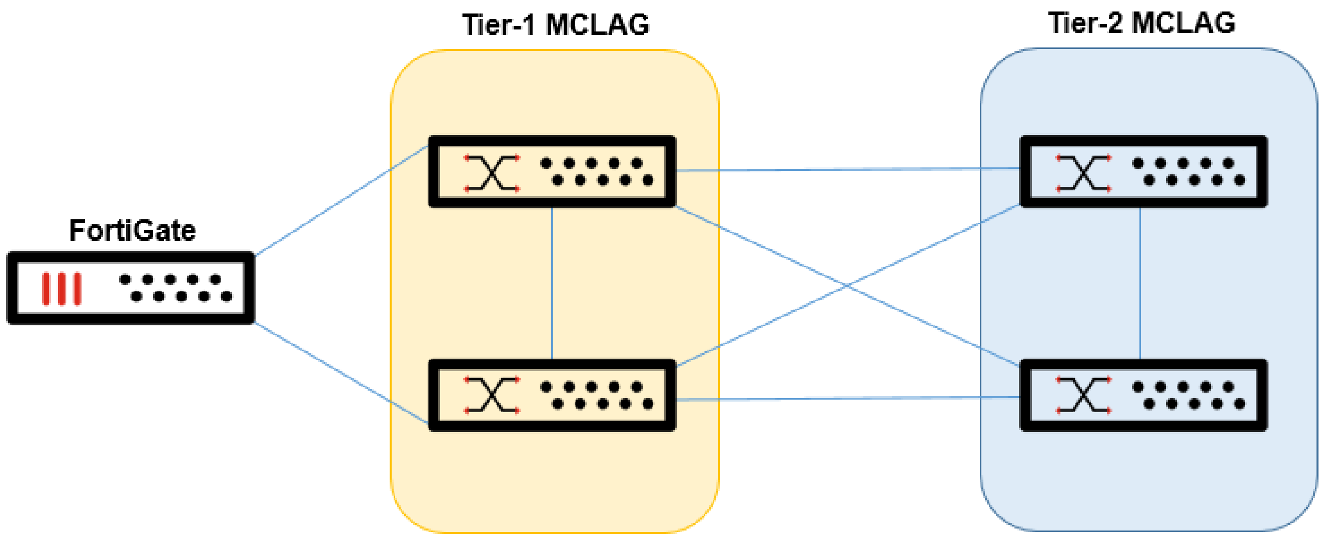
  After you accept the recommendation, the `set lldp-profile default-auto-mclag-icl` command is configured on the two switches with the recommended interchassis link (ICL) ports, and the `config switch auto-isl-port-group` command is configured on the parent MCLAG peer group.
- When a connected tier-2 MCLAG peer group is detected and FortiOS detects a possible tier-3 MCLAG pair of switches, FortiOS recommends forming a tier-3 MCLAG.

  After you accept the recommendation, the `set lldp-profile default-auto-mclag-icl` command is configured on the two switches with the recommended ICL ports, and the `config switch auto-isl-port-group` command is configured on the parent MCLAG peer group.

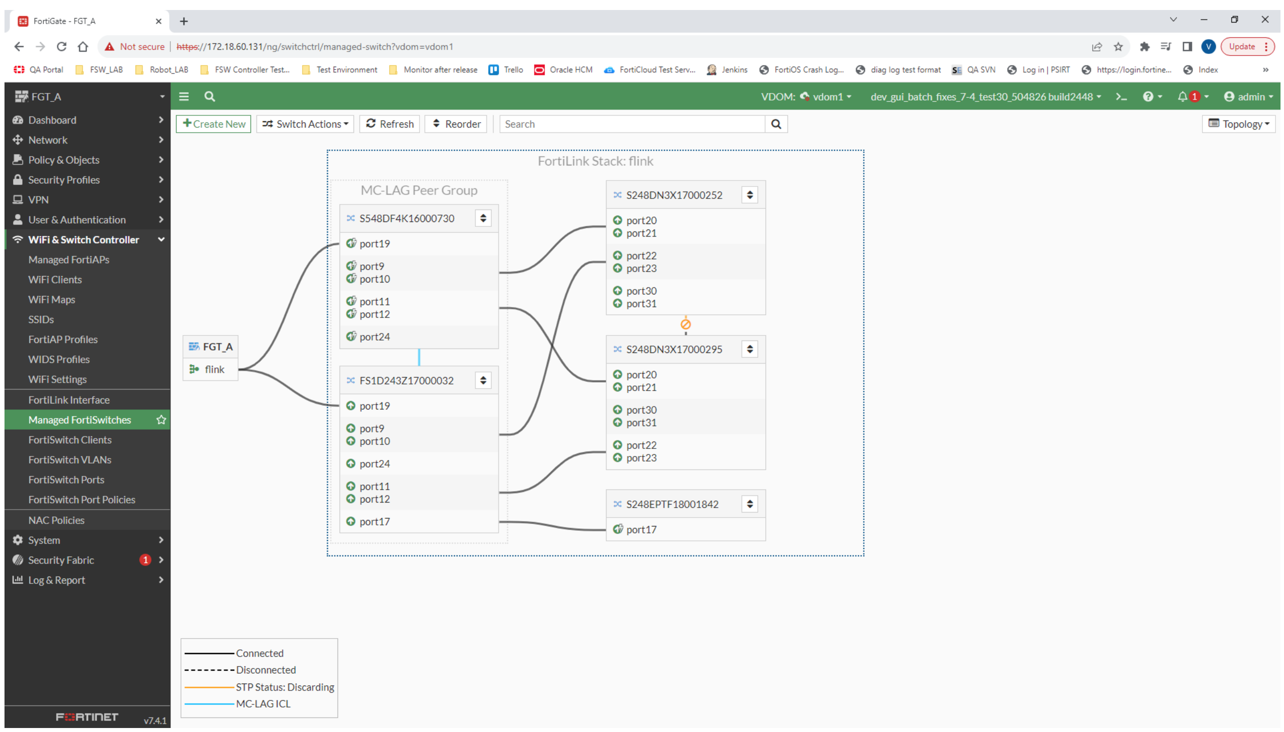
For detection to be successful, there must be fully meshed connection (each tier-2 FortiSwitcch unit must have a connection to each tier-1 FortiSwitch unit; each tier-3 FortiSwitch unit must have a connection to each tier-2 FortiSwitch unit.

Example

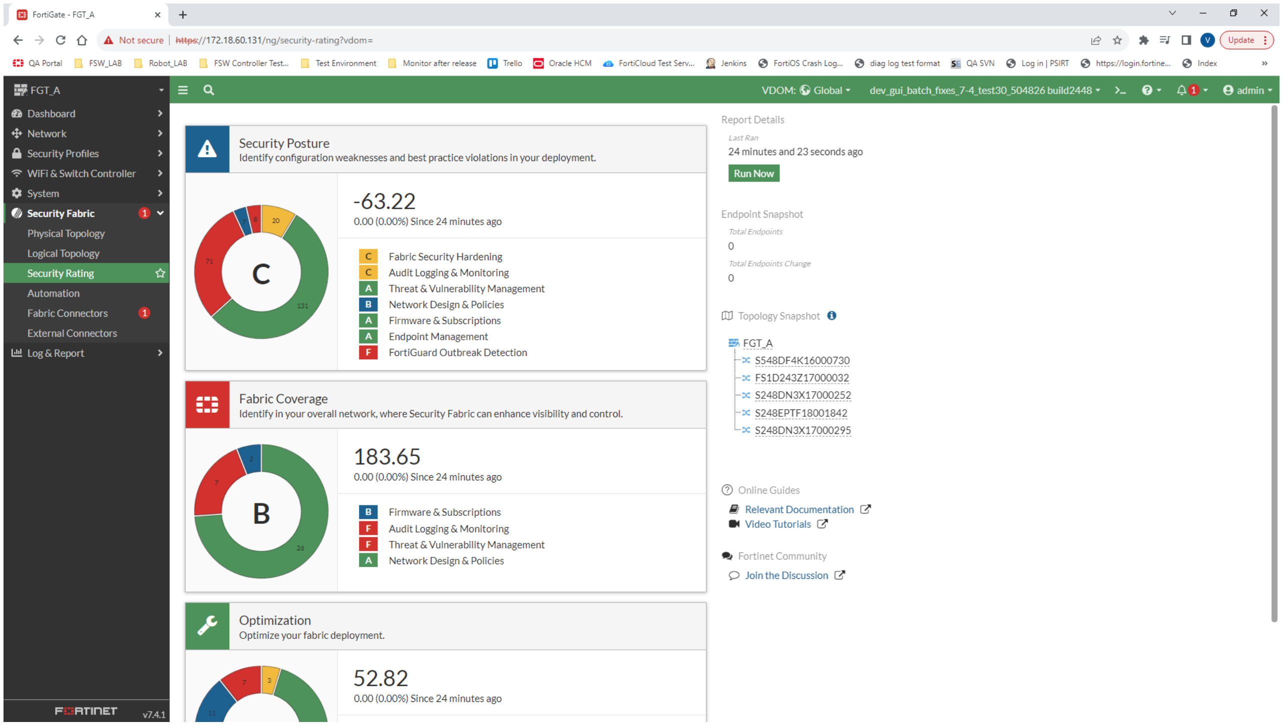
In this example, a FortiGate device manages four FortiSwitch units. Two of the switches already form an MCLAG, and the user wants a second MCLAG tier for redundancy.



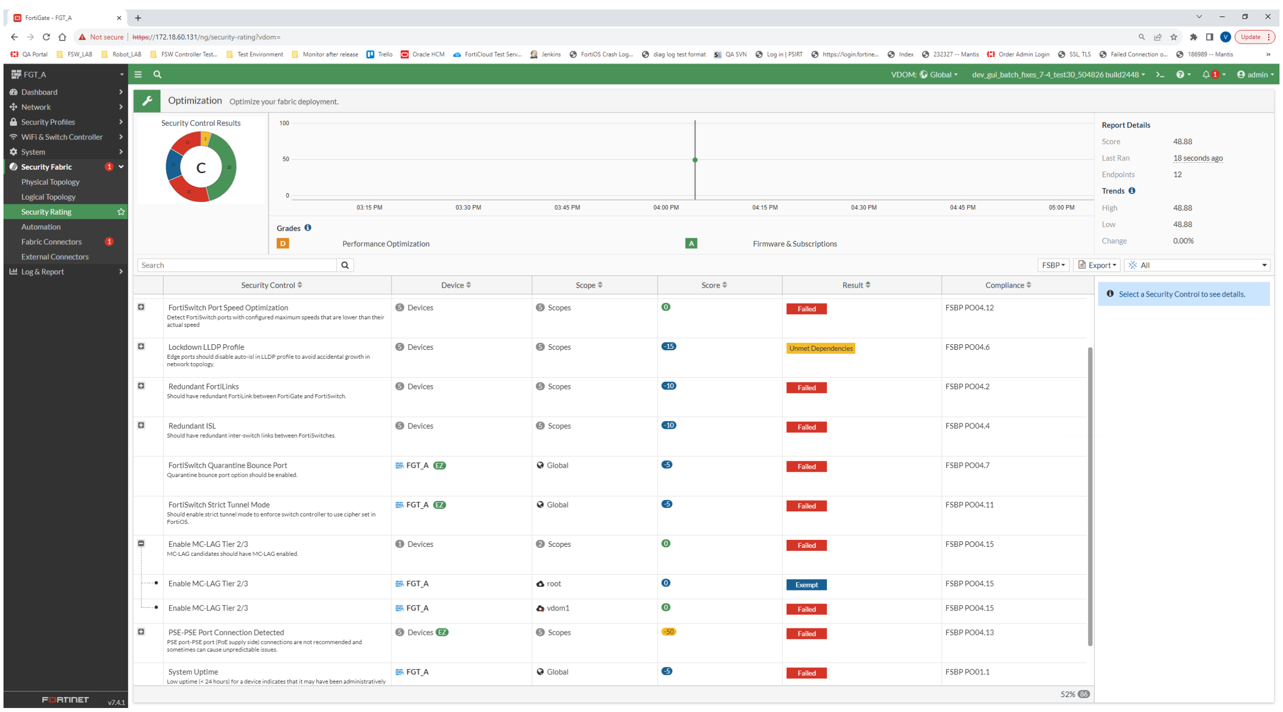
1. In the FortiOS GUI, go to *WiFi & Switch Controller > Managed FortiSwitches* and verify that the two tier-2 FortiSwitch units are the same model so that they can form an MCLAG.



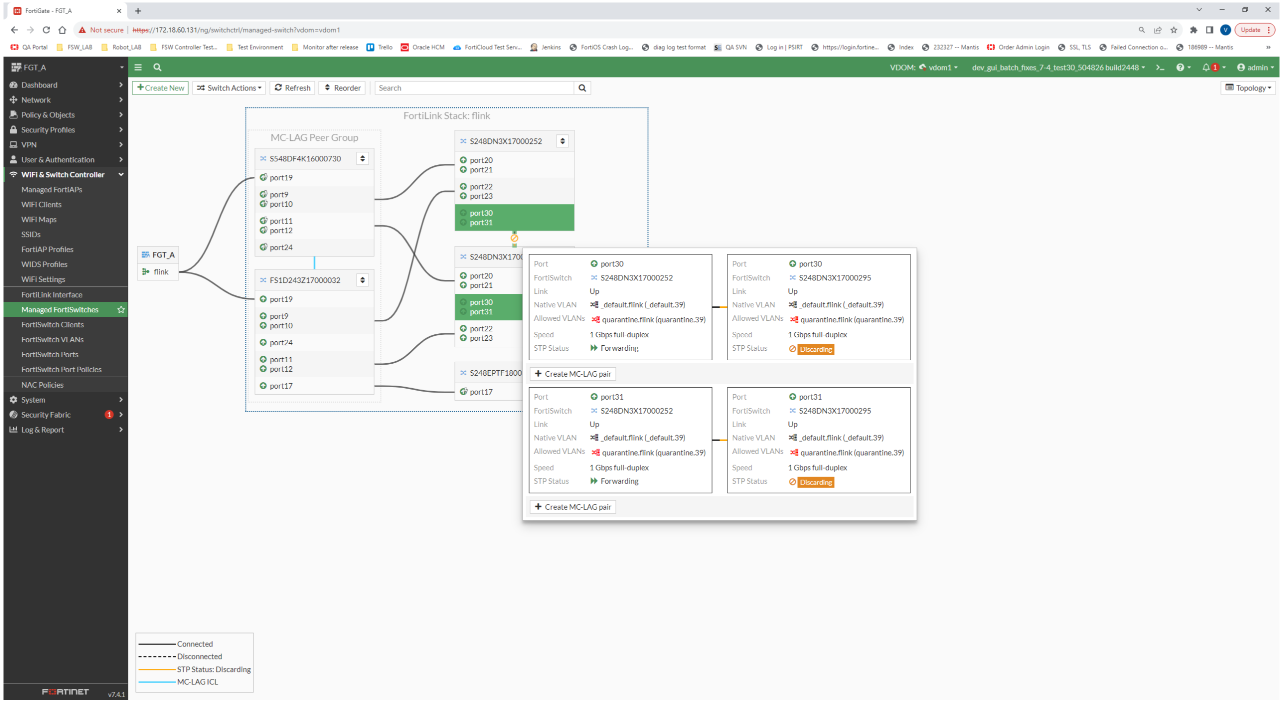
2. Go to *Security Fabric > Security Rating* and click *Run Now*.



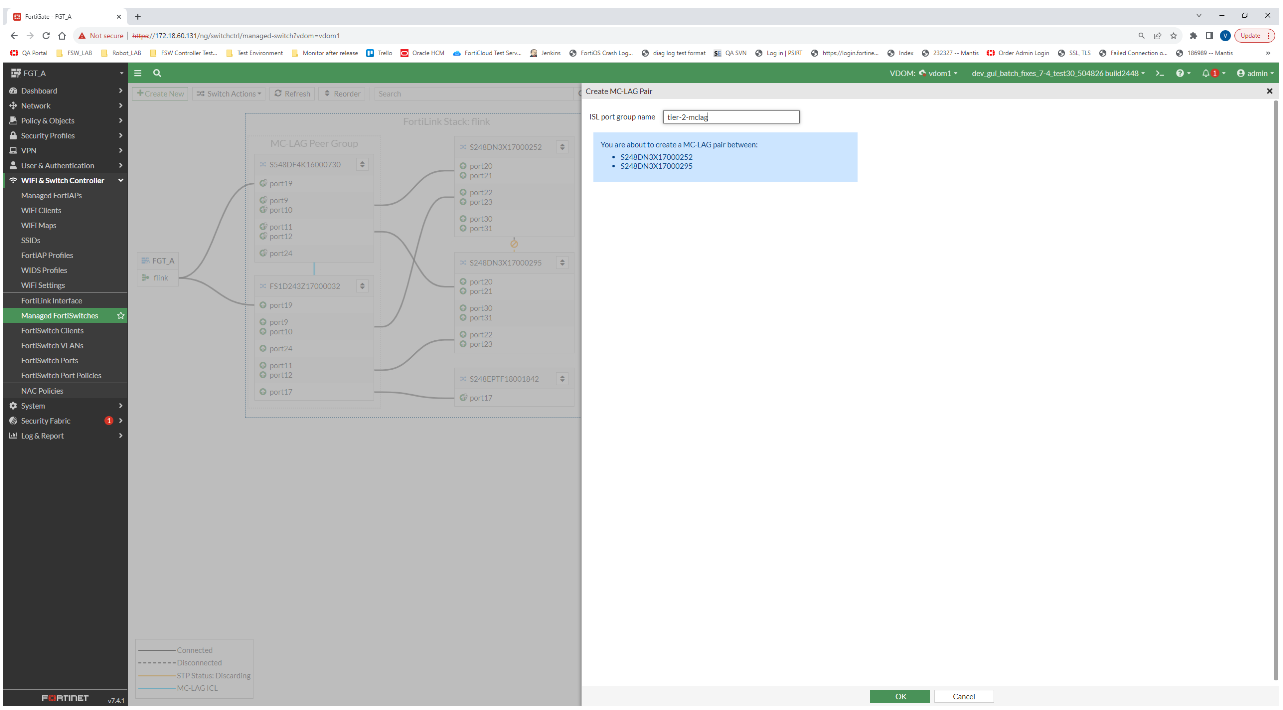
3. After the security rating report has run, expand the *Optimization* results to see *Enable MC-LAG Tier 2/3*.



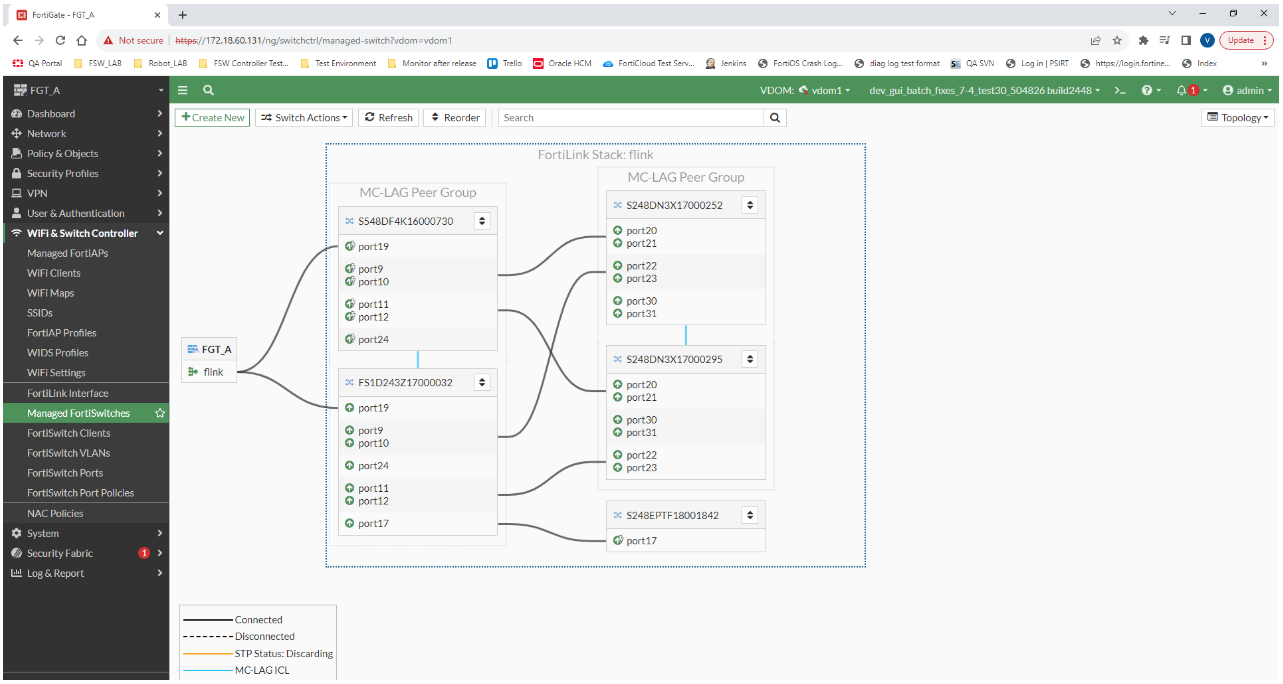
4. Go to *WiFi & Switch Controller > Managed FortiSwitches* and hover over the link connecting the two tier-2 FortiSwitch units. Click *Create MC-LAG pair*.



5. In the *Create MC-LAG Pair* panel, enter the ISL port group name.



6. The *Managed FortiSwitches* page shows that the MCLAG is formed for the tier-2 managed FortiSwitch units.



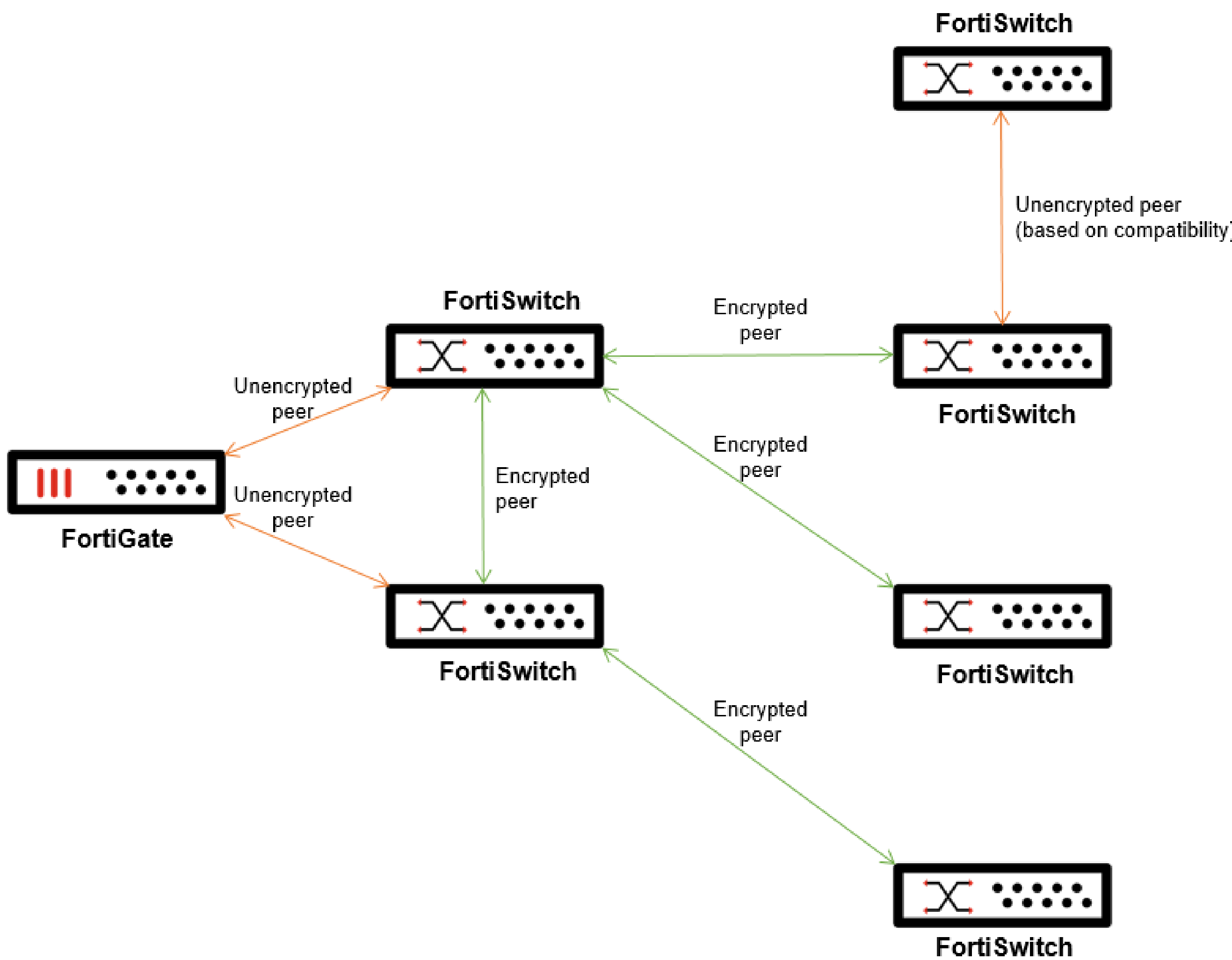
Support for the authentication and encryption of fabric links - 7.4.1

The FortiLink secured fabric provides authentication and encryption to all fabric links, wherever possible, making your Security Fabric more secure.

By default, authentication and encryption are disabled on the Security Fabric. After you specify the authentication mode and encryption mode for the FortiLink secured fabric in the LLDP profile:

1. FortiOS authenticates the connected LLDP neighbors.
2. FortiOS forms an authenticated secure inter-switch link (ISL) trunk.
3. Ports that are members of the authenticated secure ISL trunk are encrypted with Media Access Control security (MACsec) (IEEE 802.1AE-2018).
4. After the peer authentication (and MACsec encryption, if enabled) is complete, FortiOS configures the user VLANs.
5. If FortiOS detects a new FortiSwitch unit in the Security Fabric, one of the FortiSwitch peers validates whether the new switch has a Fortinet factory SSL certificate chain. If the new FortiSwitch unit has a valid certificate, it becomes a FortiSwitch peer in the Fortinet secured fabric.

The following figure shows the FortiLink secured fabric. The links between the FortiGate device and the managed FortiSwitch units are always unencrypted. The green links between FortiSwitch peers are encrypted ISLs. The orange links between FortiSwitch peers are unencrypted ISLs.



Authentication modes

By default, there is no authentication. You can select one of three authentication modes:

- *Legacy*—This mode is the default. There is no authentication.
- *Relax*—If authentication succeeds, FortiOS forms a secure ISL trunk. If authentication fails, FortiOS forms a restricted ISL trunk.

  A restricted ISL trunk is the same as a regular ISL trunk, but FortiOS does not add any user VLANs. The restricted ISL trunk allows limited access so that users can authenticate unauthenticated switches. Use a restricted ISL trunk for a new FortiSwitch unit that was just added to the Security Fabric or a FortiSwitch unit that does not support authentication or encryption.
- *Strict*—If authentication succeeds, FortiOS forms a secure ISL trunk. If authentication fails, no ISL trunk is formed.

Encryption modes

By default, there is no encryption. You must select the `strict` or `relax` authentication mode before you can select the `mixed` or `must` encryption mode.

- *None*—There is no encryption, and FortiOS does not enable MACsec on the ISL trunk members.
- *Mixed*—FortiOS enables MACsec on the ISL trunk ports that support MACsec; the ISL trunk members act as encrypted links. FortiOS disables MACsec on the ISL members that do not support MACsec; these ISL trunk members act as unencrypted links.
- *Must*—FortiOS enables MACsec on all ISL trunk members. If the port supports MACsec, the port acts as an encrypted link. If the port does not support MACsec, the port is removed from the ISL trunk, but the port still functions as a user port.

Configuring the FortiLink secured fabric

To configure the FortiLink secured fabric:

1. Configure the LLDP profile.
2. Assign the LLDP profile to a FortiSwitch physical port.

To configure the LLDP profile:

```
config switch-controller lldp-profile
   edit {LLDP_profile_name | default-auto-isl | default-auto-mclag-icl}
      set auto-isl-auth {legacy | relax | strict}
      set auto-isl-auth-user <string>
      set auto-isl-auth-identity <string>
      set auto-isl-auth-reauth <10-3600>
      set auto-isl-auth-encrypt {none | mixed | must}
      set auto-isl-auth-macsec-profile default-macsec-auto-isl
   next
end
```

Option	Description	Default
{LLDP_profile_name | default-auto-isl | default-auto-mclag-icl}	Select one of the two default LLDP profiles (`default-auto-isl` or `default-auto-mclag-icl`) or create your own LLDP profile.	No default
auto-isl-auth {legacy | relax | strict}	Select the authentication mode.	legacy
auto-isl-auth-user <string>	Select the user certificate, such as `Fortinet_Factory`. This option is available when `auto-isl-auth` is set to `relax` or `strict`.	No default
auto-isl-auth-identity <string>	Enter the identity, such as `fortilink`. This option is available when `auto-isl-auth` is set to `relax` or `strict`.	No default

Option	Description	Default
auto-isl-auth-reauth <10-3600>	Enter the reauthentication period in minutes. This option is available when `auto-isl-auth` is set to `relax` or `strict`.	3600
auto-isl-auth-encrypt {none | mixed | must}	Select the encryption mode. This option is available when `auto-isl-auth` is set to `strict` or `relax`.	none
auto-isl-auth-macsec-profile <string>	Use the `default-macsec-auto-isl` profile. This option is available when `auto-isl-auth-encrypt` is set to `mixed` or `must`.	default-macsec-auto-isl

Configuration example

```
config switch-controller lldp-profile
   edit customLLDPprofile
      set auto-isl-auth relax
      set auto-isl-auth-user Fortinet_Factory
      set auto-isl-auth-identity fortilink
      set auto-isl-auth-reauth 60
      set auto-isl-auth-encrypt mixed
      set auto-isl-auth-macsec-profile default-macsec-auto-isl
   next
end

config switch physical-port
   edit port49
      set lldp-profile customLLDPprofile
      set speed auto-module
      set storm-control-mode disabled
   next
end
```

Viewing the FortiLink secured fabric

To get information from the FortiGate device about which FortiSwitch units ports are authenticated, secured, or restricted:

```
execute switch-controller get-physical-conn {dot | standard} <FortiLink_interface>
```

To get the FortiLink authentication status for the port from the FortiSwitch unit:

```
diagnose switch fortilink-auth status <port_name>
```

To get the FortiLink authentication traffic statistics for the port from the FortiSwitch unit:

```
diagnose switch fortilink-auth statistics <port_name>
```

To delete the FortiLink authentication traffic statistics for the port from the FortiSwitch unit:

```
execute fortilink-auth clearstat physical-port <port_name>
```

To reauthenticate FortiLink secured fabric peers from the specified port from the FortiSwitch unit:

```
execute fortilink-auth reauth physical-port <port_name>
```

To reset the authentication for the FortiLink secured fabric from the FortiSwitch unit on the specified port:

```
execute fortilink-auth reset physical-port <port_name>
```

To display statistics and status of the FortiLink secured fabric for the port from the FortiSwitch unit:

```
get switch lldp auto-isl-status <port_name>
```

To display the status of the FortiLink secured fabric for the trunk from the FortiSwitch unit:

```
get switch trunk
```

Requirements and limitations

- FortiOS 7.4.1 or later and FortiSwitchOS 7.4.1 or later are required.
- FortiLink mode over a layer-2 network and FortiLink mode over a layer-3 network are supported.
- VXLAN is not supported.
- When a new FortiSwitch unit is added to the fabric, it must have a Fortinet factory SSL certificate before it is allowed to become an authenticated peer within the FortiLink secured fabric.
- When a new FortiSwitch unit is added to the FortiLink secured fabric with the `strict` authentication mode, the restricted ISL trunk is not formed. You must configure the FortiSwitch unit manually (under the `config switch lldp-profile` command).
- You need to manually import a custom certificate on the managed FortiSwitch units first; then you can specify the custom certificate on the FortiLink secured fabric with the `set auto-isl-auth-user` command under `config switch-controller lldp-profile`. After that, you can configure the custom certificate on the running Security Fabric.

Synchronize the FortiOS interface description with the FortiSwitch VLAN description - 7.4.1

Starting in FortiOS 7.4.1, the FortiOS switch controller supports the synchronization of the FortiGate system interface description to the switch VLAN description (up to the first 63 characters of FortiSwitch VLAN description field in FortiOS). This allows a more flexible use of the Tunnel-Private-Group-Id RADIUS attribute. To use the maximum length of 63 characters, set the `vlan-identity` command to `description` (under `config switch-controller global`).

Configuration example

To synchronize the FortiGate system interface description to the switch VLAN description:

1. Configure the FortiSwitch VLAN on the FortiGate device:
```
config system interface
   edit "vlan11"
   set vdom "vdom1"
      set ip 6.6.6.1 255.255.255.0
      set allowaccess ping https ssh http fabric
      set description "Test VLAN"
      set device-identification enable
      set role lan
      set snmp-index 45
      set interface "port11"
      set vlanid 111
   next
end
```
2. On the FortiSwitch unit, check that the FortiLink interface name is stored in the value for the `set description` command.
```
config switch vlan
   edit 11
      set description "Test VLAN"
   next
end
```

Support FortiSwitch management using HTTPS - 7.4.2

Starting in FortiOS 7.4.2 with FortiSwitchOS 7.4.2, you can use FortiLink with HTTPS to manage FortiSwitch units. Using FortiLink with HTTPS simplifies the management process and improves the user experience and efficiency.

The FortiGate device supports using both the CAPWAP protocol and HTTPS at the same time. Each FortiSwitch unit supports using the CAPWAP protocol or HTTPS; you cannot use both protocols to manage the same FortiSwitch unit.

FortiLink with HTTPS uses the same technology as FortiLAN Cloud to operate over both layer 2 and layer 3.

When you are using FortiLink with HTTPS to manage FortiSwitch units, the same FortiLink features are supported as when you are using FortiLink with the CAPWAP protocol.

To use FortiLink with HTTPS:

1. On the FortiSwitch unit, enable the FortiLink HTTPS management mode (CAPWAP remains enabled):
```
config switch-controller global
   set mgmt-mode https
end
```

2. On the FortiSwitch unit, set the FortiLAN Cloud service to FortiLink with HTTPS, enter the FortiLink IPv4 address, and enable the status:
```
config system flan-cloud
   set service-type fortilink-https
   set name <FortiLink_IPv4_addresss>
   set status enable
end
```
3. On the FortiGate device, authorize the FortiSwitch unit if it has not already been authorized:
```
config switch-controller managed-switch
   edit <FortiSwitch_serial_number>
      set fsw-wan1-admin enable
   next
end
```
4. On the FortiGate device, check that the tunnel has been established to allow FortiLink with HTTPS:
```
execute switch-controller get-conn-status
```
For example:
```
FGT_A (vdom1) (Interim)# execute switch-controller  get-conn-status
Managed-devices in current vdom vdom1:

FortiLink interface : port11
SWITCH-ID          VERSION            STATUS          FLAG   ADDRESS         JOIN-TIME
     SERIAL
S524DN4K16000116  v7.4.0 (0796)     Authorized/Up   2T   10.255.1.2      Mon Dec 18
15:41:34 2023    S524DN4K16000116
S248EPTF18001384  v7.4.1 (787)      Authorized/Up   2    10.255.1.5      Mon Dec 18
15:41:43 2023    S248EPTF18001384
S248EPTF18001827  N/A               Discovered/Down 2                    N/A
            S248EPTF18001827
S124EN5918003682  N/A               Discovered/Down 2                    N/A
            S124EN5918003682

       Flags: C=config sync, U=upgrading, S=staged, D=delayed reboot pending, E=config sync
error, 2=L2, 3=L3, V=VXLAN, T=tunnel, X=External
       Managed-Switches: 4 (UP: 2 DOWN: 2 MAX: 72)
```
5. On the FortiSwitch unit, check that FortiLAN Cloud has established the FortiLink connection:
```
S224DF3X15000367 # get system flan-cloud-mgr connection-info
```
For example:
```
S524DN4K16000116 # get system flan-cloud-mgr connection-info

Service Name:           : FortiLink
User Account-ID         : 0
SSL verify Code         : ok
Access Service          : IP= 10.255.1.1, Port= 443, Connected on: 2023-12-18 15:41:33
Bootstrap Service       : hostname= , Port= 0

State-Machine           : State= FLAN_MGR_STATE_READY, Event= EV_READY_SSL_SESSION_ESTD

SSL Local End-Point     : Interface: internal,  IP: 10.255.1.2
SSL Tunnel Uptime       : Days: 0  Hours: 0 Mins: 2 [Connected @2023-12-18 15:41:33]
SSL Tunnel stats        : restart-count= 279, Restart Reason= Boot-Strap fails to setup
SSL to Cloud
```

```
Stats:
========
Switch  Keep Alive  Tx/Reply := 3 / 1
Manager Keep Alive  Rx/Error := 2 / 0

Socks   Req Rx/Last Stream-ID  := 1193 / 5
Reset   Req Rx/last Stream-ID  := 137 / 276
Goaway  Req Rx  := 0
Unknown Req Rx  := 0

Syslog FD/Tx/Err  := 10 / 62 / 0

FortiLink details
=======================
stream_id : 5
online state_id : 7
localSock fd : 11
stpTelSock fd : 12
dhcpTelSock fd : 13
igmpsTelSock fd : 14
macSock fd : 15
cmfSock fd : 16
FortiGate - no response counter : 0
FortiGate - [Last no response time @1969-12-31 16:00:00]
online TX counter : 6
online RX_ACK counter : 6
online RX_NACK counter : 0
topology req : 8
topology resp : 4
system telemetry req : 8
system telemetry resp : 3
interface telemetry req : 2
interface telemetry resp : 2
mac telemetry req : 0
mac telemetry resp : 0
dot1x user req : 0
dot1x user resp : 0
lldp nbr req : 0
lldp nbr resp : 0
mac cache req : 0
mac cache resp : 0
trunk state req : 21
trunk state resp : 7
port state req : 4
port state resp : 2
poe status req : 0
poe status resp : 0

Used SOCKS stream-id:
=======================
SID       SockFd    Proxy-Ports            State         Description

__________________________________________________________________
1         0         UNKNOWN:0<-->0         DATA         BOOTSTRAP
3         0         UDP:9514<-->0          DATA         SYSLOG DATA
5         0         UNKNOWN:0<-->0         DATA         FORTILINK
```

To log in from the FortiGate device to a switch managed by FortiLink with HTTPS:

```
execute switch-controller ssh <FortiSwitch_user_name> <FortiSwitch_serial_number>
```

For example:

```
execute switch-controller ssh admin S524DF4K15000024
```

Set the priority for dynamic or egress VLAN assignment - 7.4.2

Starting in FortiOS 7.4.2 with FortiSwitchOS 7.4.2, you can change how a managed FortiSwitch unit searches for VLANs with names (specified in the `set description` command) that match the Tunnel-Private-Group-Id or Egress-VLAN-Name attribute.

Before FortiOS 7.4.2 and FortiSwitchOS 7.4.2, if there was more than one VLAN with the same name (specified in the `set description` command), the managed FortiSwitch unit selected the VLAN with the lowest VLAN ID that matched the Tunnel-Private-Group-Id or Egress-VLAN-Name attribute.

In the following example, the Tunnel-Private-Group-Id attribute is set to `testVLAN`, and three VLANs have the same name of `testVLAN`. The managed FortiSwitch unit matches the Tunnel-Private-Group-Id attribute with the VLAN with the lowest ID, VLAN 4.

VLAN ID	VLAN name
4	`testVLAN`
5	`testVLAN`
6	`testVLAN`

In FortiOS 7.4.2 with FortiSwitchOS 7.4.2, you can assign a priority to each VLAN. If there is more than one VLAN with the same name (specified in the `set description` command), the managed FortiSwitch unit selects the VLAN with the lowest `assignment-priority` value (which is the highest priority) of the VLANs with names that match the RADIUS Tunnel-Private-Group-Id or Egress-VLAN-Name attribute. The `assignment-priority` value can be 1-255. By default, the `assignment-priority` is 128. The lowest `assignment-priority` value gets the highest priority.

In the following example, the Tunnel-Private-Group-Id attribute is set to `localVLAN`, and four VLANs have the same name of `localVLAN`. The managed FortiSwitch unit matches the Tunnel-Private-Group-Id attribute with the VLAN with the lowest priority, VLAN 5.

VLAN ID	VLAN name	VLAN priority
4	`localVLAN`	50
5	`localVLAN`	25
6	`localVLAN`	75
7	`localVLAN`	100

To set the priority on the managed FortiSwitch unit for matching VLAN names:

```
config switch-controller managed-switch
   edit <FortiSwitch_serial_number>
      config vlan
         edit <VLAN_name>
```

```
            set assignment-priority <1-255>
         next
      end
   next
end
```

For example:

```
config switch-controller managed-switch
   edit "S524DF4K15000024"
      config vlan
         edit vlan5
            set assignment-priority 200
         next
      end
   next
end
```

Specify how RADIUS request attributes are formatted - 7.4.2

Starting in FortiOS 7.4.2 with FortiSwitchOS 7.4.1, you can specify how the following RADIUS request attributes are formatted when they are sent to the RADIUS server:

- User-Name

  You can select a colon, hyphen, or single hyphen to use as a delimiter, or you can select `none` for no delimiter. By default, you can use a hyphen as the delimiter.
- User-Password

  You can select a colon, hyphen, or single hyphen to use as a delimiter, or you can select `none` for no delimiter. By default, you can use a hyphen as the delimiter.
- Called-Station-Id

  You can select a colon, hyphen, or single hyphen to use as a delimiter, or you can select `none` for no delimiter. By default, you can use a hyphen as the delimiter.
- Calling-Station-Id

  You can select a colon, hyphen, or single hyphen to use as a delimiter, or you can select `none` for no delimiter. By default, you can use a hyphen as the delimiter.

The following are examples of MAC addresses with the different delimiters:

- Using a colon as a delimiter: `00:11:22:33:44:55`
- Using a hyphen as a delimiter: `00-11-22-33-44-55`
- Using a single hyphen as a delimiter: `001122-334455`
- Using `none` for no delimiter: `001122334455`

You can also select whether to use lowercase or uppercase letters in MAC addresses. By default, lowercase letters are used.

To specify how RADIUS request attributes are formatted:

```
config switch-controller managed-switch
   edit <FortiSwitch_serial_number>
      config 802-1X-settings
         set local-override enable
         set mac-username-delimiter {colon| hyphen | none | single-hyphen}
```

```
        set mac-password-delimiter {colon| hyphen | none | single-hyphen}
        set mac-calling-station-delimiter {colon| hyphen | none | single-hyphen}
        set mac-called-station-delimiter {colon| hyphen | none | single-hyphen}
        set mac-case {lowercase | uppercase}
      end
    next
end
```

FortiExtender

This section includes information about FortiExtender related new features:

Fast failover of CAPWAP control channel between two uplinks

This information is also available in the FortiExtender 7.4 Admin Guide (FGT-Managed):

- Fast failover of CAPWAP control channel between two uplinks

When a FortiExtender is configured as a FortiGate LAN extension and has two uplinks to the FortiGate access controller (AC), the system is able to perform a fast failover of the CAPWAP LAN extension control channel. Two CAPWAP sessions are established between the FortiGate and the FortiExtender: one is active and the other is standby. When the active uplink goes down, the CAPWAP LAN extension control channel changes to use the other standby uplink quickly. When the previously active uplink comes back up, the CAPWAP LAN extension control channel continues to use the previously standby uplink used for the failover event as the control channel.

To display the active and standby sessions for the CAPWAP LAN extension control channel on the FortiGate:

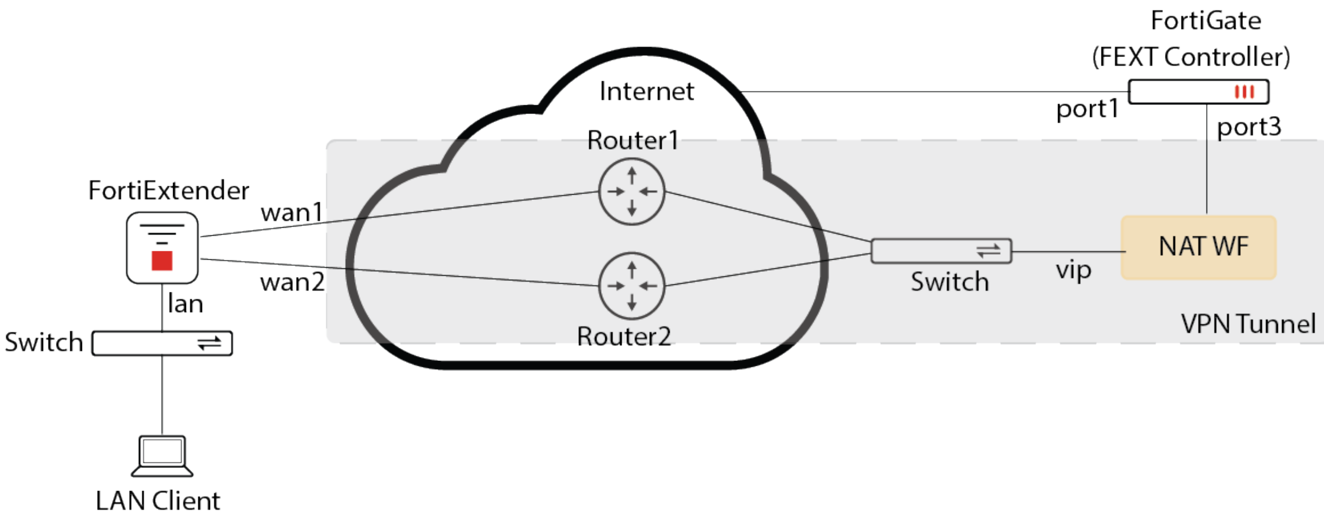
1. Execute the CLI command `get extender session-info`.
   In the CLI output, the active session is marked as `lan-extension` and the standby session is marked as `secondary`.

To display the active and standby sessions for the CAPWAP LAN extension control channel on the FortiExtender:

1. Execute the CLI command `get extender status`.
   In the CLI output, the active and standby sessions and the uplink ports are displayed when both uplinks are up; only the active session and the uplink port are displayed when a single uplink is up.

Topology

In the following diagram, the FortiGate (FEXT controller) port3 has the CAPWAP control channel to the FortiExtender (uplinks wan1 and wan2). The FortiExtender-200F port1 and port2 stand for wan1 and wan2.



CLI

The following CLI outputs show the configuration of the uplink failover event and how this new feature works.

1. Once the FortiExtender has two uplinks (port1, port2) that can reach the FortiGate, two CAPWAP sessions are established. One of them is the CAPWAP control channel (5246).

```
*** FGT console displays two extender sessions, one of which works as lan-extension
control channel.

FortiGate-501E # get extender session-info
Total 2 WS sessions, 0 AS sessions:
fg connectors:
extender sessions:
FX0035919000000 : 3.3.3.1:60440 (dport 65535) seconadry,  running,  install,
data-enable, refcnt 5, miss_echos -1, up-time 363 secs, change 1
FX0030000000000 : 3.3.3.1:5246 (dport 47997) lan-extension,  running,  install,
 data-enable, refcnt 7, miss_echos -1, up-time 2216 secs, change 0


*** FEXT console displays CAPWAP channel with active session (port1) and standby
session (port2):

FX200F0000000000 # get extender status
Extender Status
    name                 : FX200F0000000000
    mode                 : CAPWAP
    session              : active
      fext-addr          : 5.5.5.1
      ingress-intf       : port1
      controller-addr    : 1.1.1.10:5246
      controller-name    : FG5H1E5818904105
      uptime             : 0 days, 0 hours, 36 minutes, 31 seconds
```

```
      management-state   : CWWS_RUN
    session              : standby
      fext-addr          : 6.6.6.1
      ingress-intf       : port2
      controller-addr    : 1.1.1.10:5246
      controller-name    : FG5H1E5818904105
      uptime             : 0 days, 0 hours, 5 minutes, 38 seconds
      management-state   : CWWS_RUN
    base-mac             : E8:1C:BA:C4:4E:B1
    network-mode         : lan-extension
    fgt-backup-mode      : backup
    discovery-type       : static
    discovery-interval   : 5
    echo-interval        : 30
    report-interval      : 30
    statistics-interval  : 120
    mdm-fw-server        : fortiextender-firmware.forticloud.com
    os-fw-server         : fortiextender-firmware.forticloud.com
FX200F0000000000 #
```

2. Once the active uplink (port1) is down, the secondary session becomes the CAPWAP control channel (60440).

```
*** FGT console displays remaining extender session as lan-extension control
channel.

FortiGate-501E # get extender session-info

Total 1 WS sessions, 0 AS sessions:
fg connectors:
extender sessions:
FX0030000000000 : 3.3.3.1:60440 (dport 36583) lan-extension,  running,  install,
  data-enable, refcnt 7, miss_echos -1, up-time 481 secs, change 0


*** FEXT console displays CAPWAP channel with active session (port2):

FX200F0000000000 # get extender status
Extender Status
    name                 : FX200F0000000000
    mode                 : CAPWAP
    session              : standby
      fext-addr          : 0.0.0.0
      ingress-intf       :
      controller-addr    : 1.1.1.10:5246
      controller-name    : FG5H1E5818904105
      management-state   : CWWS_DISCOVERY
    session              : active
      fext-addr          : 6.6.6.1
```

```
      ingress-intf        : port2
      controller-addr     : 1.1.1.10:5246
      controller-name     : FG5H1E5818904105
      uptime              : 0 days, 0 hours, 7 minutes, 56 seconds
      management-state    : CWWS_RUN
    base-mac              : E8:1C:BA:C4:4E:B1
    network-mode          : lan-extension
    fgt-backup-mode       : backup
    discovery-type        : static
    discovery-interval    : 5
    echo-interval         : 30
    report-interval       : 30
    statistics-interval   : 120
    mdm-fw-server         : fortiextender-firmware.forticloud.com
    os-fw-server          : fortiextender-firmware.forticloud.com
FX200F0000000000 #
```

3. Once the uplink (port1) is recovered, the FortiGate console displays two extender sessions. The lan-extension control channel has no change (still via port2 on FEXT).

```
*** FGT console displays two extender sessions, one of which works as lan-extension
control channel.

FortiGate-501E # get extender session-info

Total 2 WS sessions, 0 AS sessions:
fg connectors:
extender sessions:
FX0030000000000 : 3.3.3.1:5246 (dport 65535) seconadry,  running,  install,
data-enable, refcnt 5, miss_echos -1, up-time 201 secs, change 1
FX0030000000000 : 3.3.3.1:60440 (dport 36583) lan-extension,  running,  install,
  data-enable, refcnt 7, miss_echos -1, up-time 1904 secs, change 0


*** FEXT console displays CAPWAP channel with active session (port2) and standby
session (port1):

FX200F0000000000 # get extender status
Extender Status
    name                  : FX200F0000000000
    mode                  : CAPWAP
    session               : standby
      fext-addr           : 5.5.5.1
      ingress-intf        : port1
      controller-addr     : 1.1.1.10:5246
      controller-name     : FG5H1E5818904105
      uptime              : 0 days, 0 hours, 1 minutes, 55 seconds
      management-state    : CWWS_RUN
    session               : active
      fext-addr           : 6.6.6.1
```

```
        ingress-intf        : port2
        controller-addr     : 1.1.1.10:5246
        controller-name     : FG5H1E5818904105
        uptime              : 0 days, 0 hours, 30 minutes, 18 seconds
        management-state    : CWWS_RUN
      base-mac              : E8:1C:BA:C4:4E:B1
      network-mode          : lan-extension
      fgt-backup-mode       : backup
      discovery-type        : static
      discovery-interval    : 5
      echo-interval         : 30
      report-interval       : 30
      statistics-interval   : 120
      mdm-fw-server         : fortiextender-firmware.forticloud.com
      os-fw-server          : fortiextender-firmware.forticloud.com
  FX200F0000000000 #
```

System

This section includes information about system related new features:

General

This section includes information about general system related new features:

Display warnings for supported Fabric devices passing their hardware EOS date

This information is also available in the FortiOS 7.4 Administration Guide:

- Downloading the EOS support package for supported Fabric devices

FortiGates, FortiSwitches, FortiAPs, and FortiExtenders can download an EOS (end of support) package automatically from FortiGuard during the bootup process or by using manual commands. Based on the downloaded EOS package files, when a device passes the EOS date, a warning message is displayed in the device's tooltip. The device is also highlighted in the following GUI locations:

- *System* > *Firmware & Registration* page
- *Security Fabric* > *Physical Topology* and *Logical Topology* pages
- *Security Fabric* > *Security Rating* page
- *Dashboard* > *Status* > *System Information* widget

The End-of-Support security rating check rule audits the EOS of FortiGates and Fabric devices. This allows administrators to have clear visibility of their Security Fabric, and helps to prevent security gaps or vulnerabilities that may arise due to devices passing their hardware EOS date.

FortiGuard updates

The EOS packages can be downloaded automatically from FortiGuard, but they can also be downloaded manually.

To manually download the EOS package from the FortiGuard server:

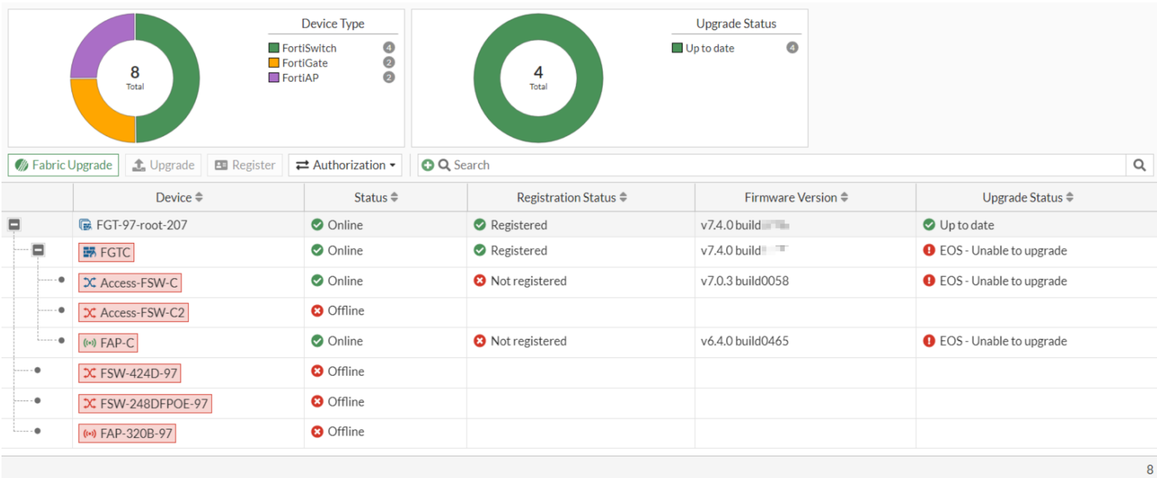
```
# diagnose fortiguard-resources update <product>-end-of-support
```

Product	Description
`fortigate-end-of-support`	FortiGate product life cycle information.
`fortiswitch-end-of-support`	FortiSwitch product life cycle information.
`fortiap-end-of-support`	FortiAP product life cycle information.
`fortiextender-end-of-support`	FortiExtender product life cycle information.

In the event the EOS package files are not downloaded due to a connection issue, use `diagnose fortiguard-resources update <product>-end-of-support` to download the package files.

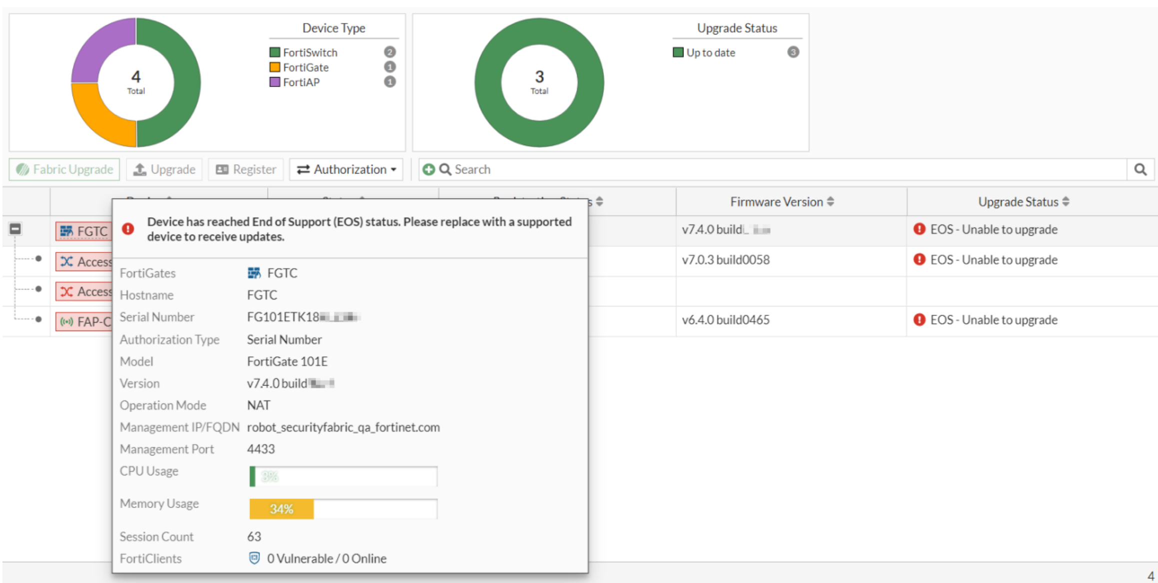
GUI warnings

On the *System* > *Firmware & Registration* page, devices that have reached EOS are highlighted in red, and their *Status* is *EOS - Unable to upgrade*.

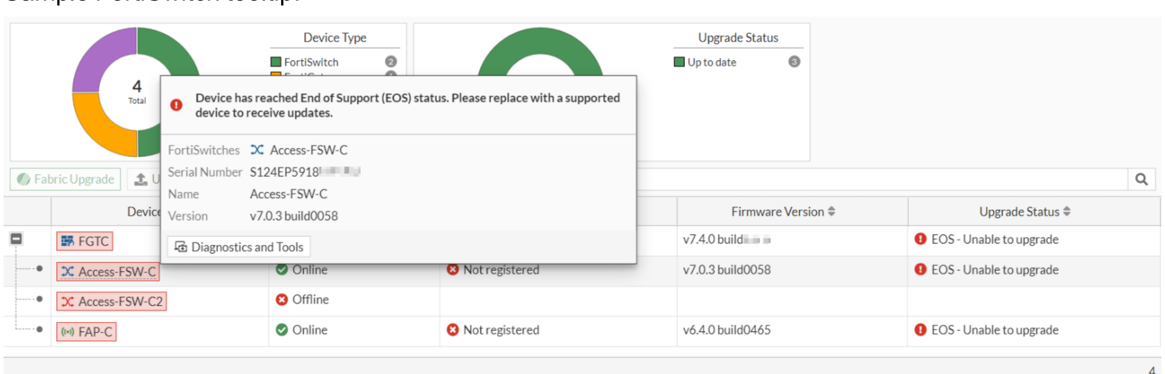


Hover over a device name to view the tooltip, which includes an EOS warning.

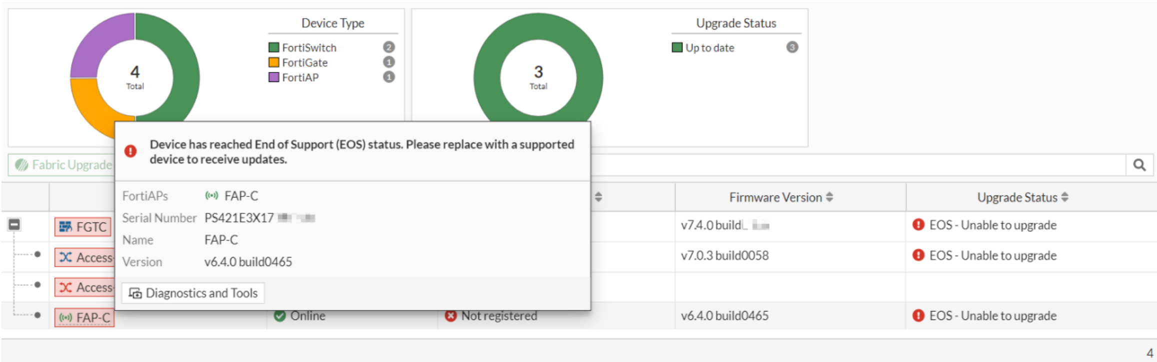
- Sample FortiGate tooltip:



- Sample FortiSwitch tooltip:

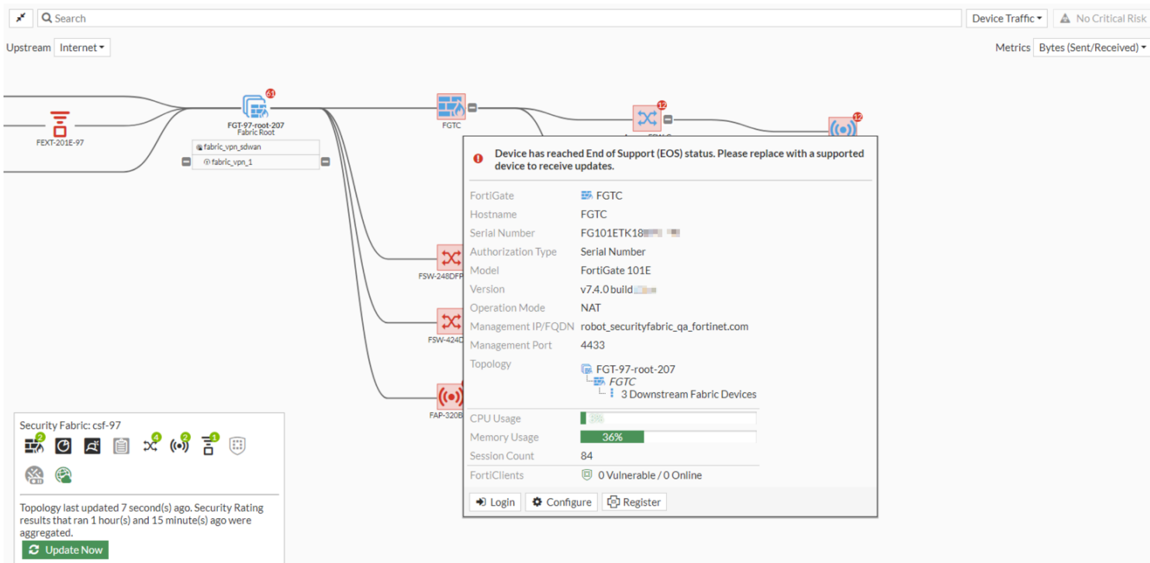


- Sample FortiAP tooltip:

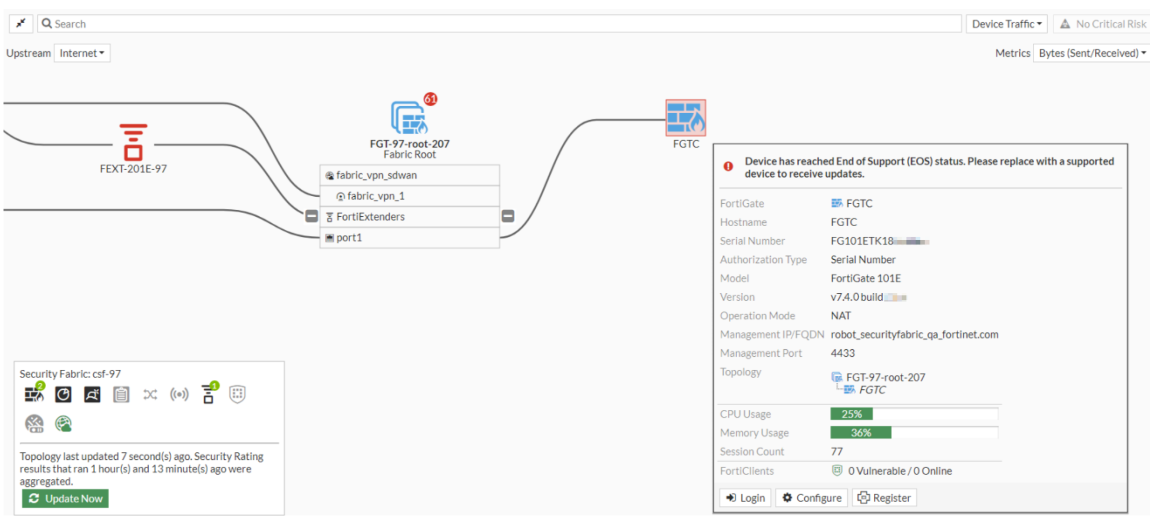


On the *Security Fabric > Physical Topology* and *Logical Topology* pages, devices that have reached EOS are highlighted in red. The device tooltips also include an EOS warning.

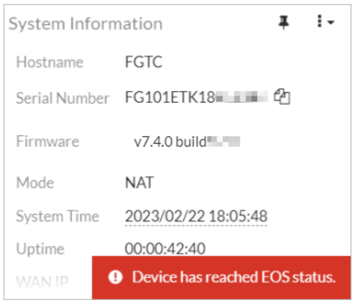
- Sample *Security Fabric > Physical Topology* page with tooltip:



- Sample *Security Fabric > Logical Topology* page with tooltip:

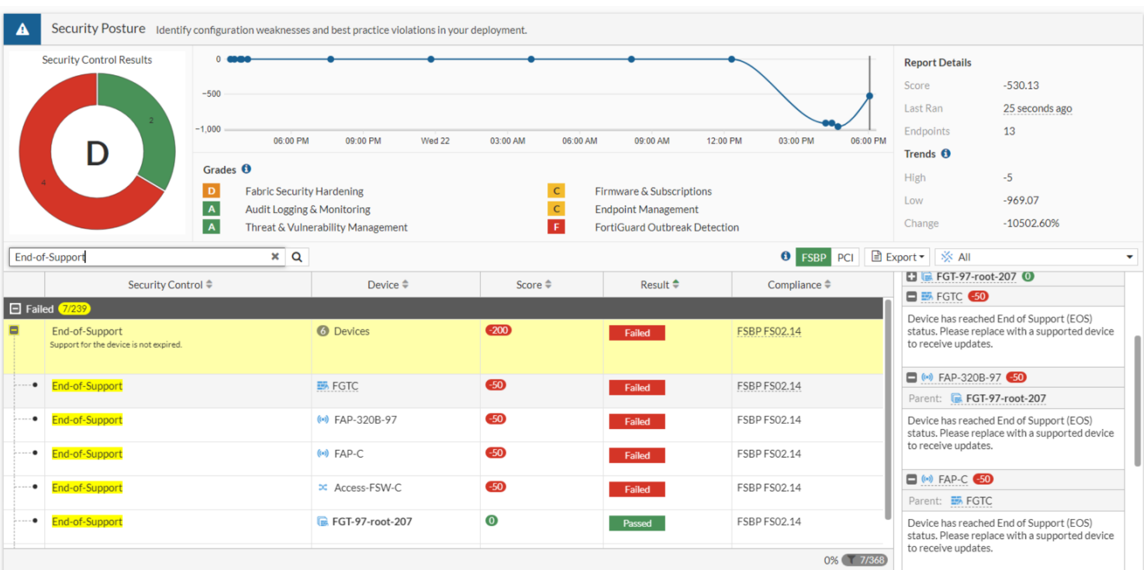


The *Dashboard* > *Status* > *System Information* widget includes a warning at the bottom of the widget that the *Device has reached EOS status*.



Security rating check

The End-of-Support security rating check rule audits the EOS of FortiGates and Fabric devices. In this result, the test is marked as *Failed* because several of the Fabric devices have reached EOS. The notice in the *Recommendations* section for the EOS devices displays the following message: *Device has reached End of Support (EOS) status. Please replace with a supported device to receive updates.*



Add setting to control the upper limit of the FQDN refresh timer

This information is also available in the FortiOS 7.4 Administration Guide:

- fqdn-max-refresh

The `fqdn-max-refresh` setting is used to control the global upper limit of the FQDN refresh timer. FQDN entries with a time to live longer than the maximum refresh value will have their refresh timer reduced to this upper limit. This allows the FortiGate to dictate the upper limit in querying for DNS updates for its FQDN addresses.

By default, the `fqdn-max-refresh` time is 3600 seconds, and the configurable range is 3600 to 86400 seconds.

```
config system dns
    set fqdn-max-refresh <integer>
end
```

Command to compute file hashes

This information is also available in the FortiOS 7.4 Administration Guide:

- Computing file hashes

This command computes the SHA256 file hashes for all of the files in a directory or directories:

```
# diagnose sys filesystem hash <paths> -d [depth]
```

`<paths>`	Add up to 25 paths to show only the hash for the files at those paths.
`-d [depth]`	Specify the maximum depth of the traversal.

This command can be used for troubleshooting and debugging the system. The file hashes of system files can be compared against known good system files to help identify any compromises made on the system files.

To hash all filesystems:

```
# diagnose sys filesystem hash
Hash contents: /bin
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/syslogd ->
/bin/init
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/acd ->
/bin/init
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/httpsnifferd ->
/bin/init
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/merged_daemons
-> /bin/init
...
/bin/init
6e2e07782dc17b8693268989f8ba1a8858a73d5291fb521e315011731cefe412        /bin/setpci
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/wad_csvc_cs ->
/bin/init
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/fds_notify ->
/bin/init
...
Hash contents: /lib
3dae8f9c15da465ffda24cebc1328725e98ee7c94a20e54af6ead7eaada45d9d        /lib/libusb-1.0.so.0
e50c6b5cad36b200d4903e4d7d5e5eac1f5c618d27fd6961011e28a892ed8866
/lib/libk5crypto.so.3
b021ad6fb16ce1e881ca586036687c1b2ae9555805817ef394284528d9e71612        /lib/libgomp.so.1
...
```

To hash specific filesystem, add the name of the filesystem:

```
# diagnose  sys  filesystem  hash  /sbin
Hash contents: /sbin
```

```
c1f81e67a53bcf70720748fe31c2380e95b4c3dfdb96957fd116fcf702bd797b        /sbin/init
Filesystem hash complete. Hashed 1 files.
```

To hash multiple filesystems, add the names of the filesystems:

Up to 25 file systems can be added.

```
# diagnose sys filesystem hash /sbin /bin
Hash contents: /sbin
c1f81e67a53bcf70720748fe31c2380e95b4c3dfdb96957fd116fcf702bd797b        /sbin/init
Hash contents: /bin
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/syslogd ->
/bin/init
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/acd ->
/bin/init
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/httpsnifferd ->
/bin/init
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/merged_daemons
-> /bin/init
```

To specify the maximum depth of the traversal:

```
# diagnose sys filesystem hash /data2 -d 1
Hash contents: /data2
a0166e804dc3d9a68fcc8015cb2d214ec40f0609e8e2aecc0eb2e5bdffc45524        /data2/new_alert_msg
7270b43899e0f72c7b9c94e66d64fd0e19881d91f74bd5ae6556eba045222e84        /data2/vir
8092e73c6a68f3cb02c86155bf3e55b2c1ab793eafcdd538beb5aa998d4b6b82        /data2/vir.x
2e29084d86f3925a0fb6bf96c4d83a6d3025fdd9cf8059ebcfc307153b9fd63b        /data2/virext
48ac27b0b5b10b3b0f3ab2f847406d524709c32117f6b721bb10448742bd5eb6        /data2/virext.x
2e29084d86f3925a0fb6bf96c4d83a6d3025fdd9cf8059ebcfc307153b9fd63b        /data2/virexdb
601316a029b28757c44515e37f48de2985d9fe8ef5c318e5f67e51369cba09f0        /data2/virexdb.x
7270b43899e0f72c7b9c94e66d64fd0e19881d91f74bd5ae6556eba045222e84        /data2/virfldb
896b71b3d9b209d339213f9d4af4088d3addd891cd292e93b5168eddb36b599a        /data2/virfldb.x
0af98283f9bcb7dff4974197f1c7f1b1013ec741c8cc6c1425119fb88f9a351b        /data2/ffdb_map_
default_res
627d2aed79770f698dbfc2bc0889f8285d1ea596c2dace8e6d3e7f00e040d990        /data2/madb.dat
96a296d224f285c67bee93c30f8a309157f0daa35dc5b87e410b78630a09cfc7        /data2/signature_
result
ceab5e70a5368aa834842973241e1ae6ca49ff5c88afb6199e5d87e1749caeb1        /data2/revision_
info_db
7eb70257593da06f682a3ddda54a9d260d4fc514f645237f5ca74b08f8da61a6        /data2/alci.dat
5840dfcf66d296be775e4e4d08bcdd014d1c91bd45e070587907d9eedab53e3e        /data2/uwdb
dc64fb8a291c7fc6d655474d00e2c42e7bb2b466de4489d33301f3ba82f64794        /data2/ffdb_
pkg.tgz.x
c66a6ccc586ce29d38854a6afee49c0464fdc0064b59c4a104544325fd1ff03f        /data2/afdb
Filesystem hash complete. Hashed 17 files.
```

```
# diagnose sys filesystem hash /data2 -d 2
Hash contents: /data2
a0166e804dc3d9a68fcc8015cb2d214ec40f0609e8e2aecc0eb2e5bdffc45524        /data2/new_alert_msg
7270b43899e0f72c7b9c94e66d64fd0e19881d91f74bd5ae6556eba045222e84        /data2/vir
8092e73c6a68f3cb02c86155bf3e55b2c1ab793eafcdd538beb5aa998d4b6b82        /data2/vir.x
2e29084d86f3925a0fb6bf96c4d83a6d3025fdd9cf8059ebcfc307153b9fd63b        /data2/virext
48ac27b0b5b10b3b0f3ab2f847406d524709c32117f6b721bb10448742bd5eb6        /data2/virext.x
2e29084d86f3925a0fb6bf96c4d83a6d3025fdd9cf8059ebcfc307153b9fd63b        /data2/virexdb
601316a029b28757c44515e37f48de2985d9fe8ef5c318e5f67e51369cba09f0        /data2/virexdb.x
7270b43899e0f72c7b9c94e66d64fd0e19881d91f74bd5ae6556eba045222e84        /data2/virfldb
```

```
896b71b3d9b209d339213f9d4af4088d3addd891cd292e93b5168eddb36b599a        /data2/virfldb.x
0af98283f9bcb7dff4974197f1c7f1b1013ec741c8cc6c1425119fb88f9a351b        /data2/ffdb_map_
default_res
627d2aed79770f698dbfc2bc0889f8285d1ea596c2dace8e6d3e7f00e040d990        /data2/madb.dat
96a296d224f285c67bee93c30f8a309157f0daa35dc5b87e410b78630a09cfc7        /data2/signature_
result
5ce22b4398f63fea2b47b7c1f00813a29851714993aee1269d3e95cbf43f4252        /data2/geodb/geoip.1
81ad258e278019dbd34fd07ba33966a6ff04e3fa352dddfe9ff362ac26d3cc88
/data2/config/cfg0000000001
e0067eb3d67b21cf39f27cb3558c5fbdafbc2c17c2afc29ab776b08e9c777a13
/data2/config/cfg0000000002
e77ad7c6b5d620d49f0f11933baf633335621de848a4229c3724152fff9aa4fa
/data2/config/cfg0000000003
228a7ed52779ba23f41a2423bfa7dbe858f24433f1702161f27678df4894f358
/data2/config/cfg0000000004
fe9e7afe7a6ccb739cb45c8d8f3b985377242ab61cc8199fa33dd475db49420f
/data2/config/cfg0000000005
b632b77348a54a2479453ab0f2c9f8e3c1e910badc8fbfb3fb841acf8eb4e35e
/data2/config/cfg0000000006
baeccb81d75f1f31503d42d3526f8831044144051f562486a89f1c5e4dd46d9c
/data2/config/cfg0000000007
ceab5e70a5368aa834842973241e1ae6ca49ff5c88afb6199e5d87e1749caeb1        /data2/revision_
info_db
7eb70257593da06f682a3ddda54a9d260d4fc514f645237f5ca74b08f8da61a6        /data2/alci.dat
5840dfcf66d296be775e4e4d08bcdd014d1c91bd45e070587907d9eedab53e3e        /data2/uwdb
dc64fb8a291c7fc6d655474d00e2c42e7bb2b466de4489d33301f3ba82f64794        /data2/ffdb_
pkg.tgz.x
c66a6ccc586ce29d38854a6afee49c0464fdc0064b59c4a104544325fd1ff03f        /data2/afdb
Filesystem hash complete. Hashed 25 files.
```

An error message is shown if an incorrect value is entered:

```
# diagnose sys filesystem hash /test-path
ERROR: Could not fetch info for path /test-path (No such file or directory)
Filesystem hash complete. Hashed 0 files.
```

```
# diagnose sys filesystem hash /bin -d 0
ERROR: depth must be greater than zero. (0)
Command fail. Return code -651
```

Support checking for firmware updates daily when auto firmware upgrade is enabled

This information is also available in the FortiOS 7.4 Administration Guide:

- Enabling automatic firmware updates

When automatic firmware update is enabled, the FortiGate will check for firmware upgrades daily between a configured time interval. When a new patch release is available, a firmware upgrade will be scheduled. By actively searching for patch updates and performing patch upgrades, the system quality is improved as new security fixes are implemented and released.

You can define the installation delay using the `auto-firmware-upgrade-delay` command. This allows you to set the number of days before installing an automatic patch-level firmware upgrade from FortiGuard. The default delay is three days.

```
config system fortiguard
    set auto-firmware-upgrade {enable | disable}
    set auto-firmware-upgrade-day {sunday monday tuesday wednesday thursday friday saturday}
    set auto-firmware-upgrade-delay <integer>
    set auto-firmware-upgrade-start-hour <integer>
    set auto-firmware-upgrade-end-hour <integer>
end
```

The `auto-firmware-upgrade-delay` command overrides the `auto-firmware-upgrade-day` command. Disable `auto-firmware-upgrade-delay` by setting it to zero if you would rather use the `auto-firmware-upgrade-day` command to select a day of the week for automatic installation, regardless of when the patch release is detected.

After the patch release is successfully installed, an email is sent to the FortiCloud account that the FortiGate is registered to.

This feature is related to the previous Enable automatic firmware updates feature from the FortiOS 7.2.0 New Features Guide. However, this feature supersedes the previous feature where applicable.

For example, the original feature does not actively search for a firmware upgrade daily. It searches for the latest patch and builds an upgrade path to that patch if there has been one or more patches since the last firmware upgrade. In contrast, this new feature enhancement will check for firmware updates daily so that the firmware is never more than one patch behind.

Example

The following example demonstrates setting automatic firmware upgrades after a delay of three days.

To demonstrate the functionality of this feature, this example uses FortiGates that are running and upgrading to fictitious build numbers.

To configure automatic firmware upgrades:

```
config system fortiguard
    set auto-firmware-upgrade enable
    set auto-firmware-upgrade-delay 3
    set auto-firmware-upgrade-start-hour 2
    set auto-firmware-upgrade-end-hour 4
end
```

The FortiGate will perform a check between the start and end hours set for the firmware upgrade to review if there is an upgrade available.

To review the available firmware upgrade check schedule:

```
# diagnose test application forticldd 13
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled.
        Next upgrade check scheduled at (local time) Thu Mar 29 03:10:56 2023
```

When an available patch upgrade is detected, the automatic firmware update will be scheduled based on the set upgrade delay.

Sample event log after a new patch upgrade is detected:

```
date=2023-03-29 time=03:10:56 eventtime=1679336380720695924 tz="-0700"
logid="0100032263" type="event" subtype="system" level="notice" vd="vdom1"
logdesc="Automatic firmware upgrade schedule changed" user="system"
msg="System patch-level auto-upgrade new image installation scheduled
     between local time Sat Apr 01 03:10:56 2023 and local time Sat Apr 01 04:00:00 2023."
```

To review the installation window of new patch releases:

```
# diagnose test application forticldd 13
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled.
        Next upgrade check scheduled at (local time) Mon Mar 30 03:10:56 2023
        New image 7.4.1b2305(07004000FIMG0021204001) installation is scheduled to
                start at Sat Apr 01 03:10:56:21 2023
                end by Sat Apr 01 04:00:00 2023
```

Once the firmware patch is successfully installed, an event log is created to track the change and an email is sent to the FortiCloud account under which the FortiGate is registered.

Sample event log after successfully updating firmware:

```
date=2023-04-01 time=03:13:04 devid="FG3H1E5819904039" devname="D"
eventtime=1679590383750408029 tz="-0700"
logid="0100022094" type="event" subtype="system" level="information" vd="vdom1"
logdesc="A federated upgrade was completed by the root FortiGate"
msg="Federated upgrade complete" version="7.4.1"
```

FortiConverter in the GUI

This information is also available in the FortiOS 7.4 Administration Guide:
- Migrating a configuration with FortiConverter

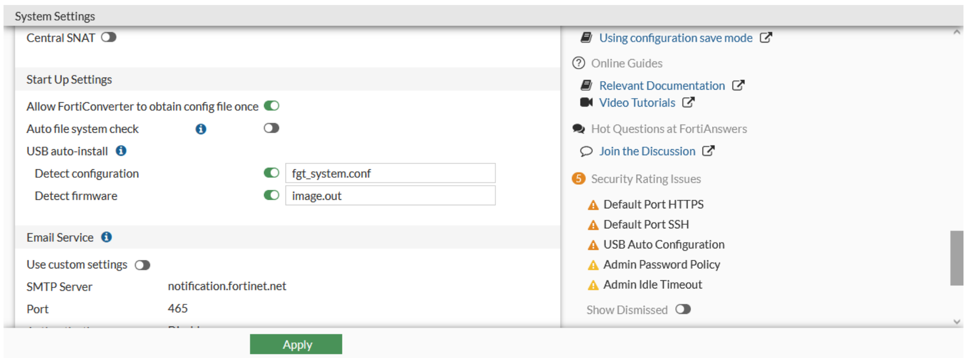
A configuration can be migrated from an older FortiGate device to a new FortiGate device directly from the FortiGate GUI, without having to access the FortiConverter portal.

Both the source and target FortiGates must be registered under the same FortiCare account and have internet connectivity to reach the FortiConverter server. The target FortiGate must also have a valid FortiConverter license.

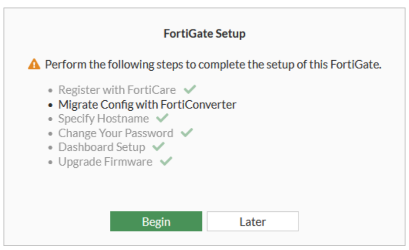
In this example, FortiGate A (FGTA) is replacing FortiGate B (FGTB). The configuration is migrated using FortiConverter, but without accessing the FortiConverter portal.

To migrate the configuration from FGTB to FGTA in the GUI:

1. On FGTB, go to *System > Settings*, enable *Allow FortiConverter to obtain config file once*, then click *Apply*.



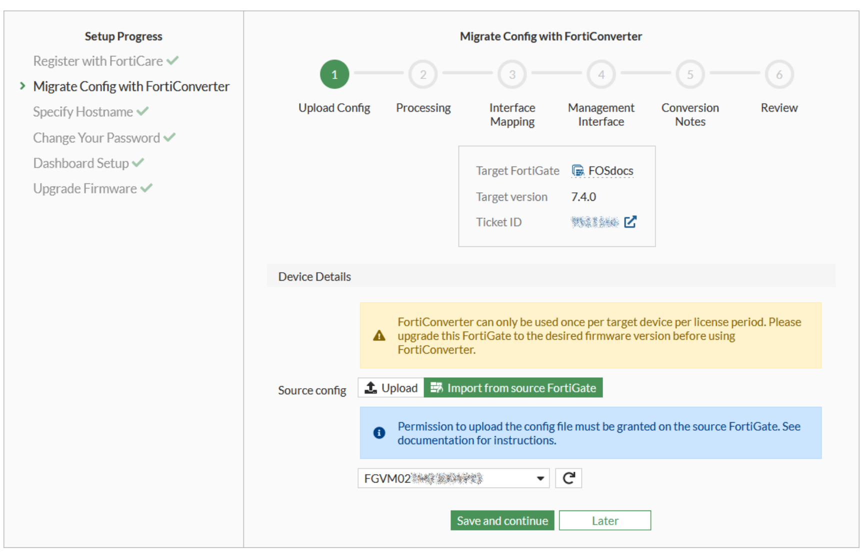
2. Log in to FGTA and on the GUI startup menu click *Begin* to start *Migrate Config with FortiConverter*.



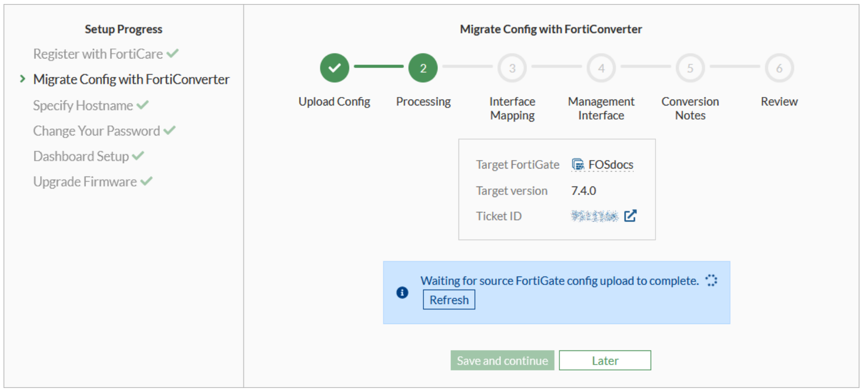
3. Click *Convert* to create a FortiConverter ticket.

   You can toggle the *Don't show again* option to turn off reminders about the migration process.

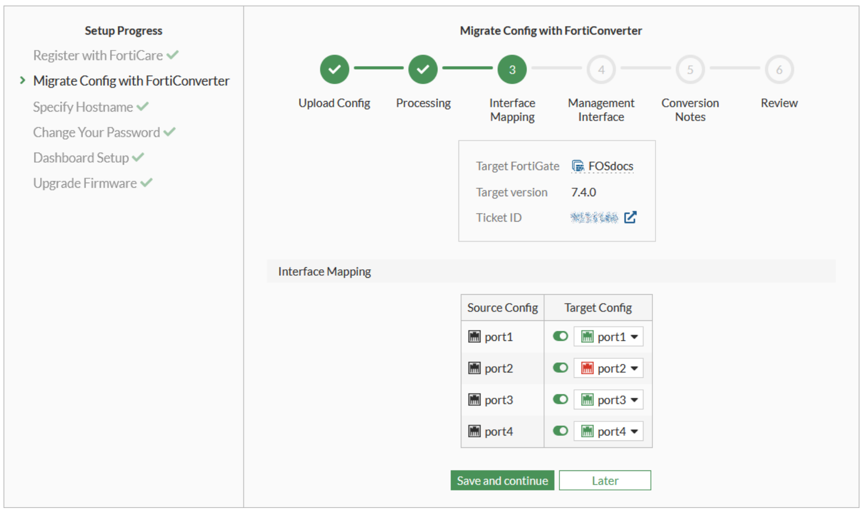
4. Select *Import from source FortiGate*, then select the source FortiGate.



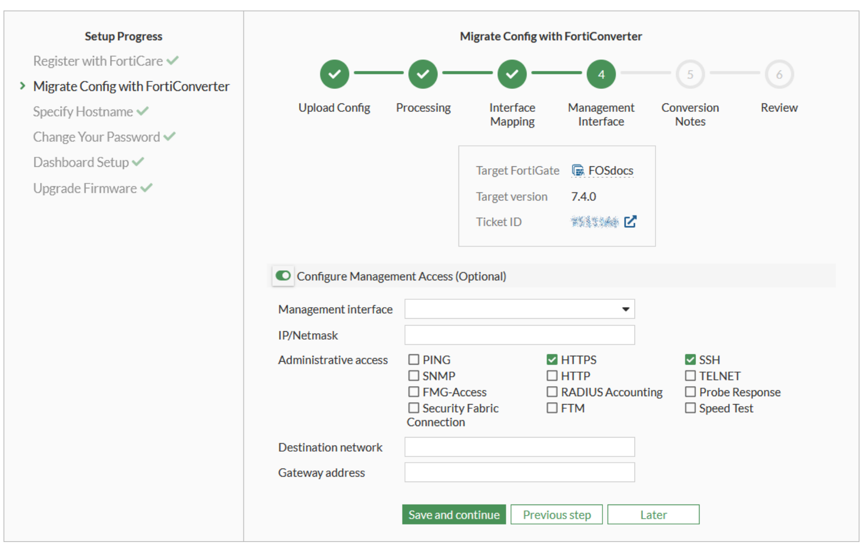
5. Click *Save and continue*, then wait for the FGTB configuration file to be uploaded to the ticket. After the configuration is uploaded, the *Allow FortiConverter to obtain config file once* is automatically disabled on FGTB.



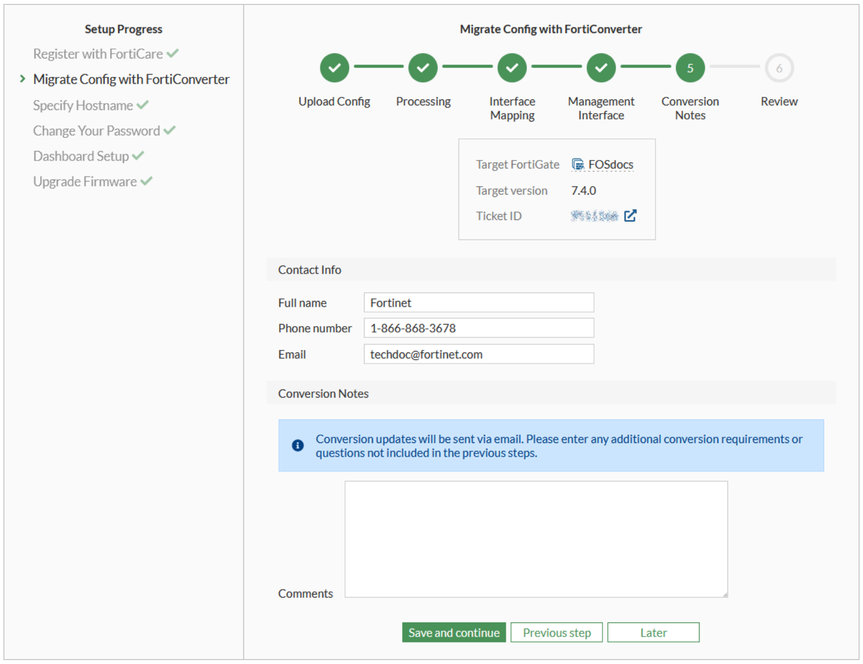
6. Define the interface mapping between the source and target FortiGates, then click *Save and continue*.



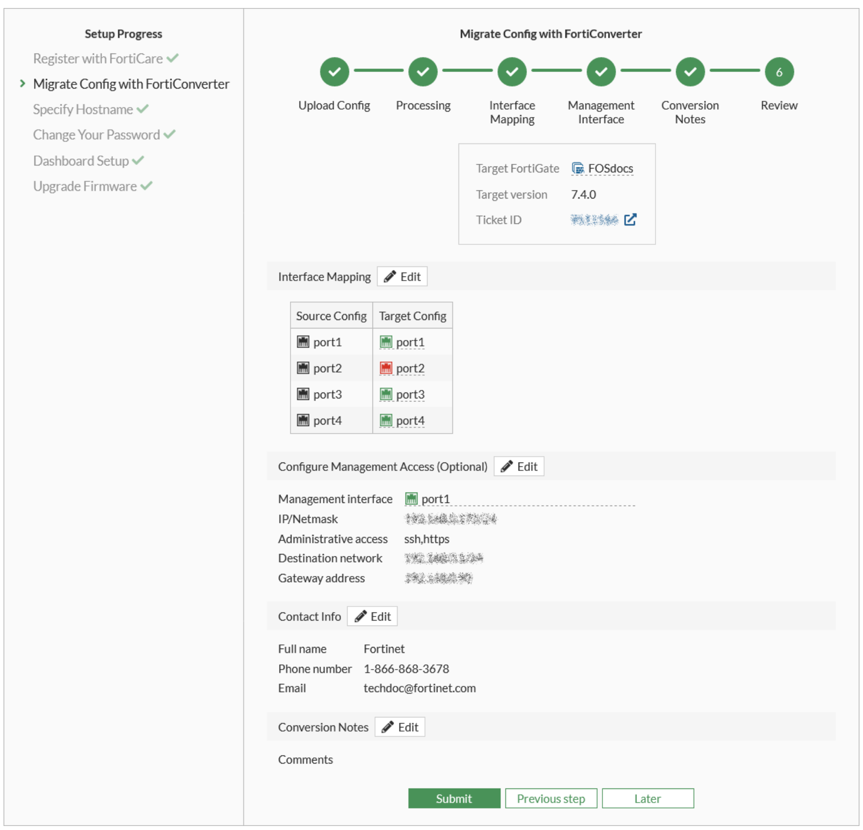
7. Optionally, configure management access on the target FortiGate (FGTA), then click *Save and continue*.



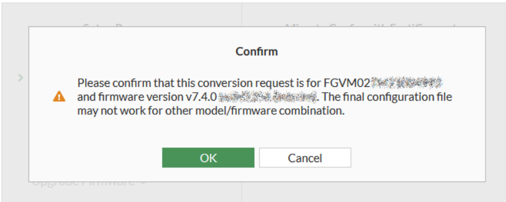
8. Specify the contact information for the ticket and enter conversion notes, then click *Save and continue*.



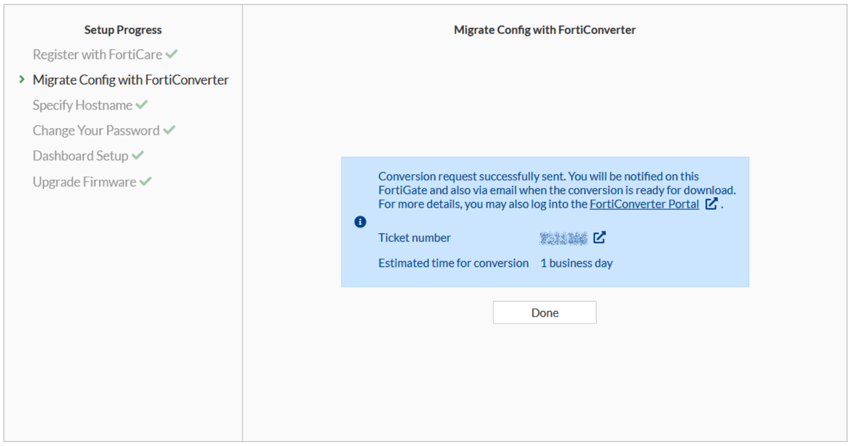
9. Review the ticket content, then click *Submit*.



10. Confirm the conversion process by clicking *OK*.



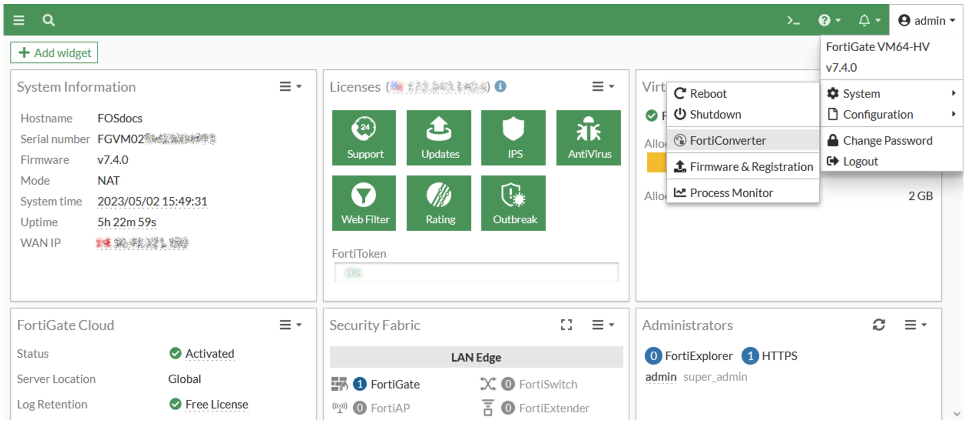
The conversion request is sent, and an email is sent to confirm that the conversion process has started in FortiConverter.



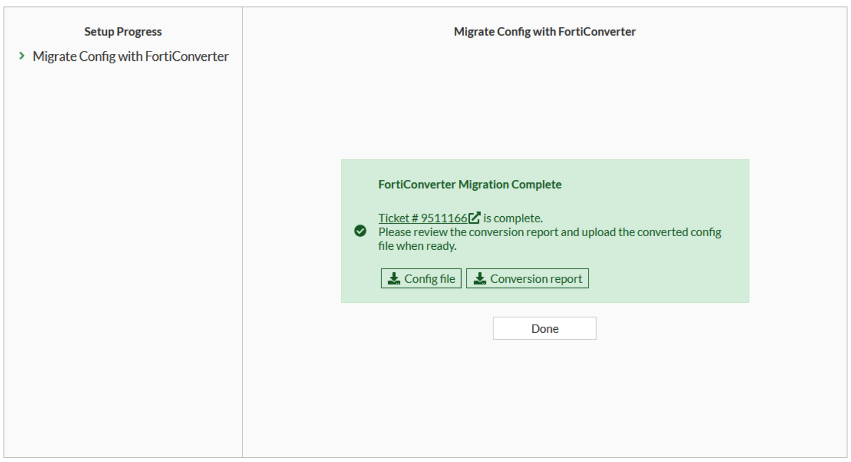
Sample email message:

```
*** This is a System Generated Message. Please do not reply to this email! ***

Dear FortiConverter User,

A new ticket XXXXXXX has been created. You can expect an initial response with one
business day.
https://service.forticonverter.com/ticket/detail/XXXXXXX

Ticket#:       XXXXXXX
Subject:       FortiConverter Service for FortiGate [FGVM02XX00000000]

Regards,
Fortinet Converter Service and Support
```

11. Click *Done*. The conversion can take a few days.
12. To check the status of the conversion process, click your administrator name and select *System* > *FortiConverter*.



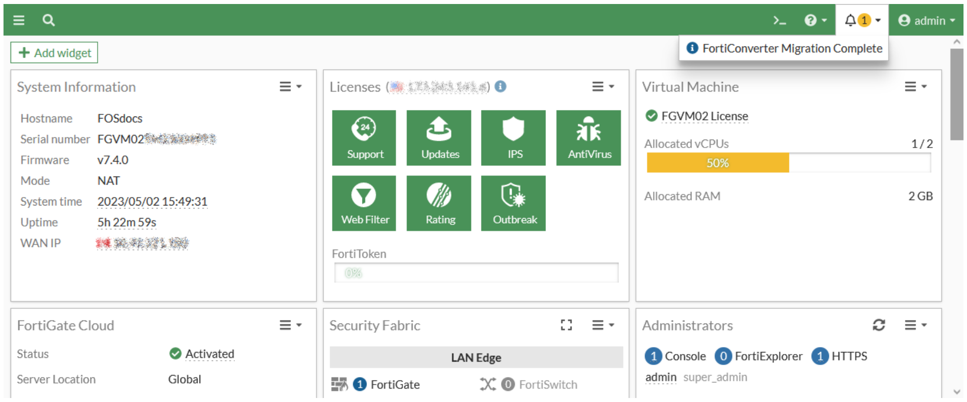
If the conversion is complete, the configuration file and conversion report can be downloaded.



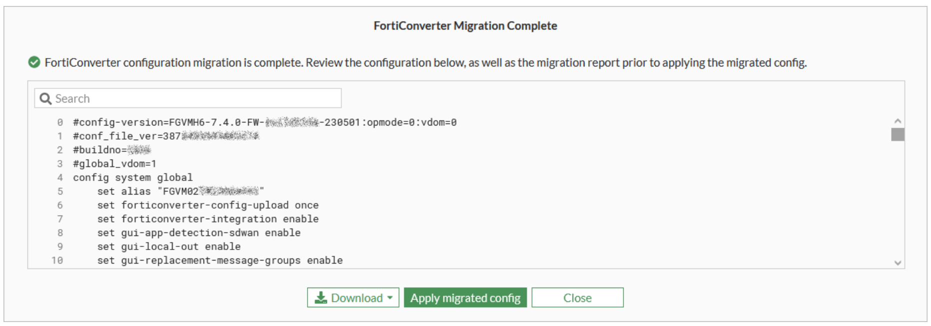
13. When the conversion process completes, you will receive an email and a notifications in the FortiGate GUI.

Sample email message:

```
*** This is a System Generated Message. Please do not reply to this email! ***

Dear FortiConverter User,

https://service.forticonverter.com/ticket/detail/XXXXXXX

Ticket#:       XXXXXXX
Subject:       FortiConverter Service for FortiGate [FGVM02XX00000000]
Ticket Status: Service Delivered

The converted config file and a summary report of the configuration conversion have
been uploaded to the ticket 9511166 under the "Converted Config File" section.
Please login to the FortiConverter Service Portal to download the files for review
and let us know if you have any questions about the conversion.

Your opinion matters to us and we would love to hear more about your experience with
FortiConverter service. The survey can be accessed on the right side of the ticket
in our service portal. Your feedback will help us to improve FortiConverter service
and we look forward to hearing from you.

Regards,
Fortinet Converter Services and Support
```

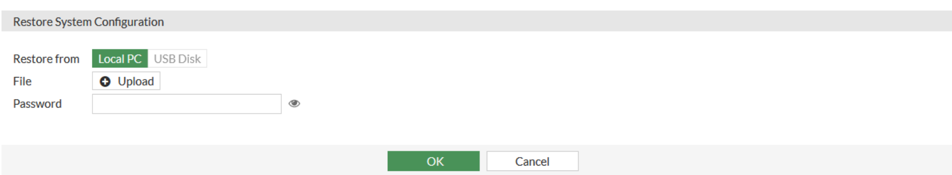


14. Click the notification to review the configuration file, download the conversion report and the migrated configuration, or apply the configuration to the FortiGate.



a. Select *Download* > *Conversion report* to download a PDF version of the conversion report.
b. Select *Download* > *Migrated config* to download the new configuration file.
c. Click *Apply migrated config* to immediately apply the new configuration file. This will cause the device to reboot.
d. Click *Close* if you need to review the configuration file and manually apply it later.

15. To manually load to configuration file, click your administrator name and select *Configuration* > *Restore*.



16. Upload the converted configuration file, then click *OK*. This will cause the device to reboot.

Prevent FortiGates with an expired support contract from upgrading to a major or minor firmware release

This information is also available in the FortiOS 7.4 Administration Guide:
- Preventing FortiGates with an expired support contract from upgrading to a major or minor firmware release

If the FortiGate support contract has expired, you will be unable to upgrade the firmware to a higher major version, such as from FortiOS 6.0 to 7.0, or to a higher minor version, such as from FortiOS 7.0 to 7.2. However, you can upgrade the firmware of a FortiGate with an expired support contract to a higher patch build, such as from FortiOS 7.4.0 to 7.4.1, to allow for security updates.

You can confirm the *Firmware & General Updates* (FMWR) contract expiry date in the *System* > *FortiGuard* page or by using the `diagnose test update info contract` command.

Updates in the GUI have been implemented for this new feature in 7.4.1. See Prevent firmware upgrades when the support contract is expired using the GUI 7.4.1 on page 613 for more information.

Example

The following example demonstrates what occurs when upgrading the firmware to a patch build and to a higher version with an expired license. The patch upgrade successfully upgrades the firmware from FortiOS 7.4.0 to 7.4.3. The major upgrade attempts and fails to upgrade the firmware from FortiOS 7.4.0 to 7.6.3.

To demonstrate the functionality of this feature, this example uses FortiGates that are running and upgrading to fictitious build numbers.

To upgrade the firmware to a higher patch build:

1. Confirm the current firmware version:

```
# get system status
Version: FortiGate-301E v7.4.0,build2303,230307 (interim)
```

2. Upgrade the firmware:

```
# execute restore image tftp v743-B2400-GA-M_B230309_FGT_301E.out 172.16.200.55
This operation will replace the current firmware version!
Do you want to continue? (y/n)y
Please wait...
Connect to tftp server 172.16.200.55 ...
......
Firmware upgrade in progress ...
Done.
```

3. Confirm the new firmware version:

```
# get system status
Version: FortiGate-301E v7.4.3,build2400,230309 (GA.M)
```

To upgrade the firmware to a higher major version:

1. Confirm the current firmware version:

```
# get system status
Version: FortiGate-301E v7.4.0,build2303,230307 (interim)
```

2. Upgrade the firmware:

```
# execute restore image tftp  v763-B1505-GA-F_B234847_FGT_301E.out 172.16.200.55
......
Firmware update licence is expired! Please update to a valid licence.
Command fail. Return code -180
```

Prevent firmware upgrades when the support contract is expired using the GUI - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

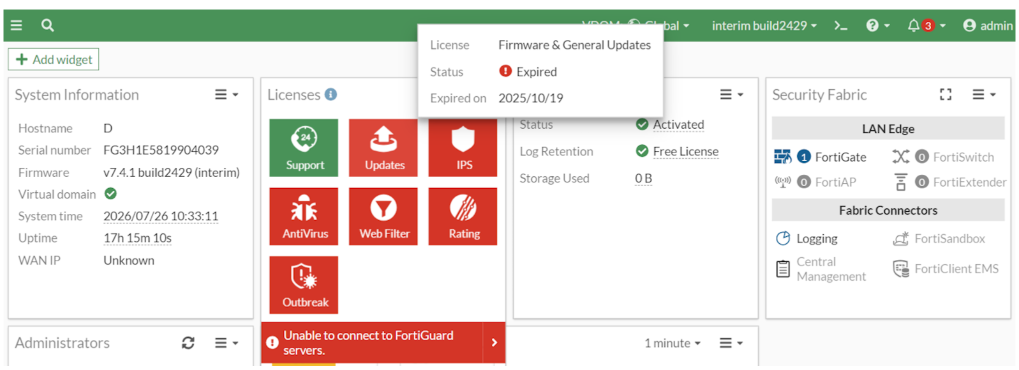
- Prevent FortiGates with an expired support contract from upgrading to a major or minor firmware release

The GUI supports the prevention of major and minor version firmware upgrades if your FortiGate has an expired support contract.

FortiGates with expired support contracts cannot upgrade the firmware to a higher major or minor version. However, the FortiGate can still be upgraded to a higher patch build, such as FortiOS 7.4.1 to 7.4.3, to allow for security updates.

This new feature is an expansion of a 7.4.0 new feature. See Prevent FortiGates with an expired support contract from upgrading to a major or minor firmware release on page 611 for more information on upgrading to major and minor versions, and firmware prevention in the CLI.

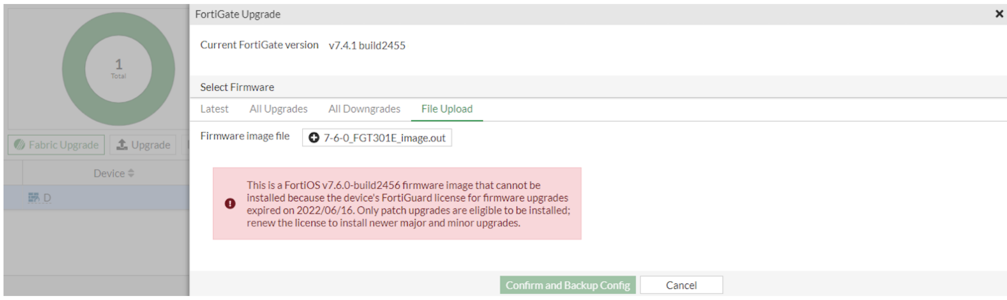
The status of the FortiGate support contract can be viewed in the *Licenses* widget from *Dashboard > Status*.



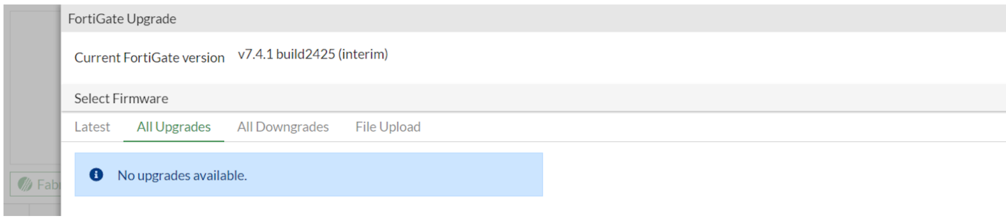
To demonstrate the functionality of this feature, this example uses FortiGates that are running and upgrading to fictitious build numbers. For more information on upgrading device firmware using the GUI, see Upgrading individual device firmware.

If the contract is expired, the following upgrade attempts will be blocked in the GUI *System > Firmware & Registration* page:

- If a higher, major or minor version firmware is uploaded to the FortiGate, the upgrade cannot be processed and a warning will display.
  In the following example, a firmware image file is uploaded in an attempt to upgrade the FortiGate from 7.4.1 to 7.6.0. However, since the license is expired, the upgrade is denied and a warning is displayed.



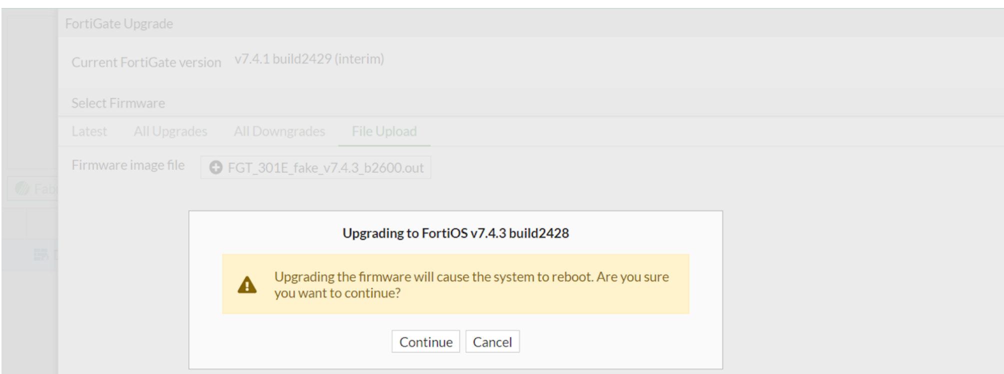
- FortiGuard upgrades will be unavailable until the support contract is renewed.



When the support contract is expired, the following actions are still available in the GUI *System > Firmware & Registration* page:

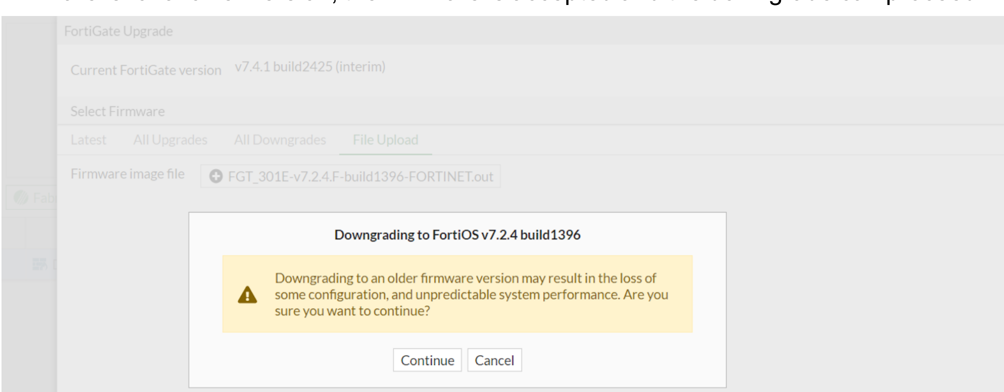
- The FortiGate firmware can be upgraded to a higher patch build to allow for necessary security updates.

  In the following example, a firmware image file is uploaded in an attempt to upgrade from 7.4.1 to 7.4.3. Since it is a patch release, the file is accepted and the upgrade can proceed.



- The FortiGate firmware can be downgraded to lower major and minor versions.

  In the following example, a firmware image file is uploaded in an attempt to downgrade from 7.4.1 to 7.2.4. Since the firmware is for a lower version, the firmware is accepted and the downgrade can proceed.



Automatic firmware upgrade enhancements - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:
- Enabling automatic firmware updates

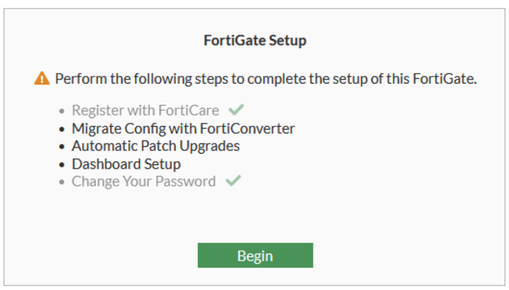
Several automatic firmware upgrade enhancements are added:

- Automatic patch upgrades are available in the FortiGate Setup wizard.
- Automatic patch upgrades can be enabled or disabled from *System > Firmware & Registration*.
- By default, entry-level FortiGates (lower than 100 series) have automatic firmware upgrades enabled.
- FortiGates belonging to a Security Fabric or FortiGates under management by a FortiManager cannot enable automatic firmware upgrade.

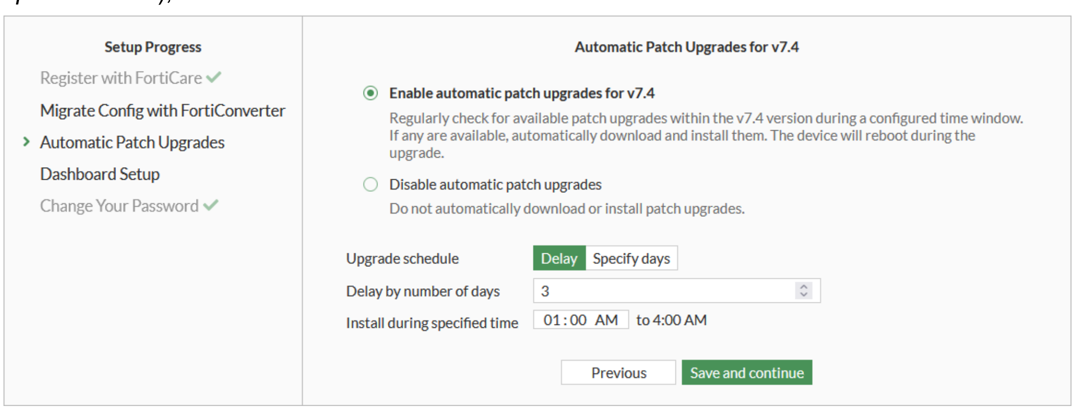
On FortiOS 7.4.2 and FortiOS 7.4.3, automatic firmware upgrade only allows upgrading to a Mature build. For information about firmware maturity, see Firmware maturity levels.

To configure automatic firmware upgrades from the GUI:

1. Log in to the FortiGate GUI and click *Begin*.

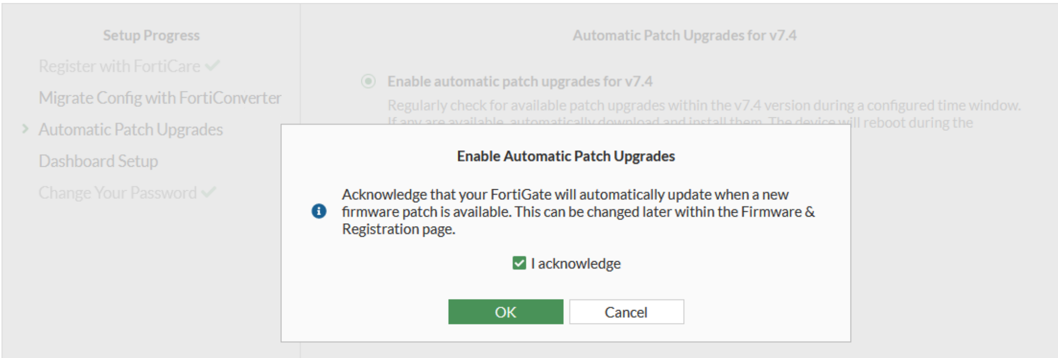


2. Select *Enable automatic patch upgrades for v7.4* (default setting).
3. Edit the upgrade and installation settings as needed (*Upgrade schedule*, *Delay by number of days*, *Install during specified time*), then click *Save and continue*.



If *Disable automatic patch upgrades* is selected, this can be changed later from the *System > Firmware & Registration* page by clicking the *Disable automatic patch upgrades* notification.

4. The *Enable Automatic Patch Upgrades* dialog opens. Select *I acknowledge* and click *OK* to proceed.



The FortiGate will be updated based on the configured schedule when a new patch is available.

5. An email is sent to alert the administrator that the firmware upgrade schedule has changed.
6. Once a patch is detected, an email is sent to alert the administrator that a new image installation is scheduled.
7. After the image installation is completed, an email is sent to alert the administrator that the federated upgrade is complete.

To view the default firmware upgrade settings:

1. Verify the FortiGuard firmware update settings:

```
show full system fortiguard | grep firmware
    set auto-firmware-upgrade enable
    unset auto-firmware-upgrade-day
    set auto-firmware-upgrade-delay 3
    set auto-firmware-upgrade-start-hour 2
    set auto-firmware-upgrade-end-hour 4
```

2. Verify the patch update schedule:

```
# diagnose test application forticldd 13
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled.
        Next upgrade check scheduled at (local time) Wed Jul 26 03:26:33 2023
```

If the FortiGate is part of a Fabric or managed by FortiManager, the `Automatic image upgrade` option is set to `disabled`.

```
# diagnose test application forticldd 13
...
Automatic image upgrade: disabled.
```

To verify the update schedule after a new patch is detected:

```
# diagnose test application forticldd 13
...
Automatic image upgrade: Enabled.
        Next upgrade check scheduled at (local time) Fri Jul 21 13:50:15 2023
        New image 7.4.2b2600(07004000FIMG0019704002) installation is scheduled to
                start at Sat Jul 22 13:03:56 2023
                end by Sat Jul 22 14:00:00 2023
```

Sample email after configuring automatic firmware upgrades:

```
From: DoNotReply@notification.fortinet.net <DoNotReply@notification.fortinet.net>
Sent: Tuesday, July 25, 2023 11:08 AM
To: ********** <*****@fortinet.com>
Subject: Automatic firmware upgrade schedule changed

date=2023-07-25 time=11:07:34 devid="FG81EPTK19000000" devname="FortiGate-81E-POE"
eventtime=1690308454221334719 tz="-0700" logid="0100032263" type="event" subtype="system"
level="notice" vd="root" logdesc="Automatic firmware upgrade schedule changed" user="system"
msg="System patch-level auto-upgrade regular check enabled."
```

Sample email after a new image installation is scheduled:

```
From: DoNotReply@notification.fortinet.net <DoNotReply@notification.fortinet.net>
Sent: Friday, July 21, 2023 1:17 PM
To: ********** <*****@fortinet.com>
Subject: Automatic firmware upgrade schedule changed

date=2023-07-21 time=13:16:50 devid="FG81EPTK19000000" devname="FortiGate-81E-POE"
eventtime=1689970609076391174 tz="-0700" logid="0100032263" type="event" subtype="system"
level="notice" vd="root" logdesc="Automatic firmware upgrade schedule changed" user="system"
msg="System patch-level auto-upgrade new image installation scheduled between local time Sat
Jul 22 13:03:56 2023 and local time Sat Jul 22 14:00:00 2023."
```

Sample event logs after the federated upgrade is complete:

```
date=2023-07-22 time=13:55:37 eventtime=1689972938126416979 tz="-0700" logid="0100032138"
type="event" subtype="system" level="critical" vd="root" logdesc="Device rebooted"
ui="sfupgraded" action="reboot" msg="User rebooted the device from sfupgraded. The reason is
'upgrade firmware'"

date=2023-07-22 time=13:55:37 eventtime=1689972938126337130 tz="-0700" logid="0100032202"
type="event" subtype="system" level="critical" vd="root" logdesc="Image restored"
ui="sfupgraded" action="restore-image" status="success" msg="User restored the image from
sfupgraded (v7.4.1,build2425 -> v7.4.2,build2426)"
```

Sample email after the federated upgrade is complete:

```
From: DoNotReply@notification.fortinet.net <DoNotReply@notification.fortinet.net>
Sent: Friday, July 22, 2023 2:00 PM
To: ********** <*****@fortinet.com>
Subject: A federated upgrade was completed by the root FortiGate

date=2023-07-22 time=14:00:09 devid="FG81EPTK19000000" devname="FortiGate-81E-POE"
```

```
eventtime=1689973183346851869 tz="-0700" logid="0100022094" type="event" subtype="system"
level="information" vd="root" logdesc="A federated upgrade was completed by the root
FortiGate" msg="Federated upgrade complete" version="7.4.2"
```

Introduce selected availability (SA) version and label - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Selected availability (SA) versions

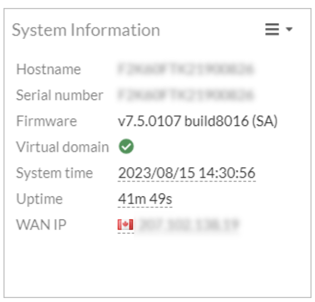
A selected availability (SA) version and label identifies special builds that are provided to customers to use for a long time. The SA version uses an odd number as the minor version and a four digit number for the patch version. The SA version and label are visible in the GUI and CLI.

SA builds are dual-signed by the Fortinet CA and a third-party CA.

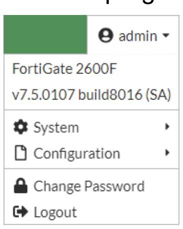
In the following example, special build 0107 is based on FortiOS 7.4.0 build 8016 and is labeled *v7.5.0107 build8016 (SA)*.

To view the SA version and label in the GUI:

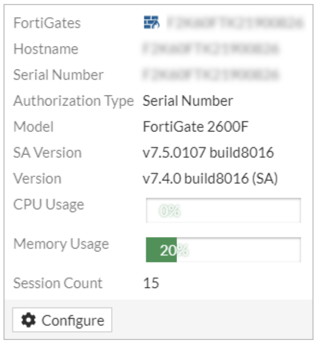
1. Go to *Dashboard* > *Status* > *System Information*. The *Firmware* option displays the SA version and label of *v7.5.0107 build8016 (SA)*.



2. On the top-right of the banner, click <*administrator name*>, such as *admin*. The SA version and label is displayed.



3. On the top-left corner of the banner, click the FortiGate name. A tooltip displays the SA version and label.



To view the SA version and label in the CLI:

```
# get system  status
Version: FortiGate-2600F v7.4.0,build8016,230711 (SA)
SA Version: v7.5.0107,build8016
Security Level: 0
Firmware Signature: certified
...
```

The `SA Version` is displayed as `v7.5.0107, build8016`.

View batch transaction commands through the REST API - 7.4.1

The commands of an uncommitted batch transaction can be viewed through the REST API from an API client with the `transaction-show` option. Previously administrators could only view commands of a batch transaction through the CLI.

Example

In this example, use the REST API to change the admin timeout of the FortiGate. Before committing the change, check the cached commands to view the pending changes. After committing the change, you cannot view the commands because the transaction is complete.

To view batch transaction commands with the REST API:

1. From an API client, start a transaction with FortiGate.

   In this example, the transaction ID is `1`.

```
user@test:~$ curl -k -X 'POST' 'https://<ip address>/api/v2/cmdb?action=transaction-start&vdom=vdom1&access_token=j8Gcs836dQsqbrd9637Qs770s0f13Q' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "timeout": 60
}'

response:
{
```

```
  "http_method":"POST",
  "revision":"df4217a73f57e09b766605b683fb5caf",
  "revision_changed":false,
  "results":{
    "transaction-id":1
  },
  "vdom":"vdom1",
  "action":"transaction-start",
  "status":"success",
  "http_status":200,
  "serial":"<serial number>",
  "version":"v7.4.2",
  "build":2484

}
```

2. Change the admin timeout on the FortiGate for the started transaction.

   For transaction ID `1`, the `admintimeout` is set to `123`.

```
user@test:~$ curl -k -X 'PUT' 'https://<ip address>/api/v2/cmdb/system/global?access_
token=j8Gcs836dQsqbrd9637Qs770s0f13Q' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -H 'X-TRANSACTION-ID: 1' \
  -d '{
  "admintimeout": 123
}'

response:
{
  "http_method":"PUT",
  "revision":"c8263664d73eeff0e47db5e142fa5306",
  "revision_changed":false,
  "status":"success",
  "http_status":200,
  "vdom":"vdom1",
  "path":"system",
  "name":"global",
  "serial":"<serial number>",
  "version":"v7.4.2",
  "build":2484
}
```

3. Before committing the commands, check the cached commands.

   The `transaction-show` results for transaction ID `1` show the uncommitted changes to `admintimeout` of `123`.

```
user@test:~$ curl -k -X 'GET'  'https://<ip address>/api/v2/cmdb?action=transaction-
show&transaction-id=1&access_token=j8Gcs836dQsqbrd9637Qs770s0f13Q' \
  -H 'accept: application/json'

response:
{
  "http_method":"GET",
  "revision":"df4217a73f57e09b766605b683fb5caf",
  "results":[
    " config global",
    "  config system global",
```

```
    "      set admintimeout 123",
    "  end",
    " end"
  ],
  "vdom":"vdom1",
  "action":"transaction-show",
  "status":"success",
  "http_status":200,
  "serial":"<serial number>",
  "version":"v7.4.2",
  "build":2484

}
```

4. Commit transaction ID 1:

```
user@test:~$ curl -k -X 'POST'   'https://<ip address>/api/v2/cmdb?action=transaction-commit&vdom=vdom1?access_token=j8Gcs836dQsqbrd9637Qs770s0f13Q'   -H 'accept: application/json'   -H 'Content-Type: application/json'   -d '{
  "transaction-id": 1
}'

response:
{
  "http_method":"POST",
  "revision":"df4217a73f57e09b766605b683fb5caf",
  "revision_changed":false,
  "status":"success",
  "http_status":200,
  "vdom":"vdom1",
  "action":"transaction-commit",
  "serial":"<serial number>",
  "version":"v7.4.2",
  "build":2484
}
```

5. Check the commands for transaction 1. An error is returned as expected because transaction 1 is complete. No cached commands are available to be viewed.

```
user@test:~$ curl -k -X GET'   'https://<ip address>/api/v2/cmdb?action=transaction-show&transaction-id=1&access_token=j8Gcs836dQsqbrd9637Qs770s0f13Q'   -H 'accept: application/json'

response:

{
  "http_method":"GET",
  "revision":"df4217a73f57e09b766605b683fb5caf",
  "error":-651,
  "status":"error",
  "http_status":500,
  "vdom":"vdom1",
  "action":"transaction-show",
  "serial":"<serial number>",
  "version":"v7.4.2",
  "build":2484
}
```

Separate the SSHD host key from the administration server certificate - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- Separating the SSHD host key from the administration server certificate

Separating the SSHD host key from the administration server certificate addresses the issue where the administration server key tends to overwrite one of the key files, which can lead to complications. This resolves the problem where the SSH module regenerates the host key files after a factory reset. This action previously prompted a warning message when an older SSH client attempted to log in to the FortiGate using SSH.

```
config system global
    set ssh-hostkey-override {enable | disable}
    set ssh-hostkey-password <password>
    set ssh-hostkey <encrypted_private_key>
end
```

The `ssh-hostkey-algo` option under `config system global` supports ECDSA 384 and ECDSA 256, allowing the SSHD to accommodate the most commonly used host key algorithms.

To configure SSH host key override in SSHD:

1. Using the ssh-keygen tool, generate the host key (ecdsa-sha2-nistp384 is used in this example).
2. Configure the SSH host key override settings:

```
config system global
    set ssh-hostkey-override enable
    set ssh-hostkey-algo ecdsa-sha2-nistp384
    set ssh-hostkey-password **********
    set ssh-hostkey <encrypted_private_key>
end
```

3. On a PC, attempt to log in to the FortiGate with the defined ecdsa-sha2-nistp384 algorithm:

```
root@PC05:~# ssh admin@172.16.200.1
The authenticity of host '172.16.200.1 (172.16.200.1)' can't be established.
ECDSA key fingerprint is SHA256:mcrMXSjtN/YjY3zQgZpxk77ezxPVGGGOL/GUOG8Oijs.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.200.1' (ECDSA) to the list of known hosts.
```

4. Verify the server host key algorithms:

```
root@PC05:~# nmap -sV --script ssh2-enum-algos 172.16.200.1
Starting Nmap 7.01 ( https://nmap.org ) at 2023-11-07 15:47 PST
Nmap scan report for FGT_A (172.16.200.1)
Host is up (0.00013s latency).
Not shown: 995 filtered ports
PORT    STATE  SERVICE      VERSION
22/tcp  open   ssh          (protocol 2.0)
| ssh2-enum-algos:
|   kex_algorithms: (8)
|       diffie-hellman-group14-sha256
|       diffie-hellman-group16-sha512
|       diffie-hellman-group18-sha512
|       diffie-hellman-group-exchange-sha256
```

```
|       curve25519-sha256@libssh.org
|       ecdh-sha2-nistp256
|       ecdh-sha2-nistp384
|       ecdh-sha2-nistp521
|   server_host_key_algorithms: (1)
|       ecdsa-sha2-nistp384
|   encryption_algorithms: (3)
```

FortiOS REST API enhances FortiManager interaction with FortiExtender - 7.4.2

The FortiOS REST API enables FortiManager firmware upgrade templates for FortiExtender modems to:

- Query the modem firmware version utilized by FortiExtender.
- Direct FortiExtender to install modem firmware updates from FortiCloud.

This feature enhances the interaction between FortiGate, FortiManager, and FortiExtender to ensure that FortiExtender firmware is always up-to-date.

The following prerequisites are required to use this feature:

- FortiExtender must be registered in FortiCloud.
- FortiExtender firmware version must be 7.4 on build 231 or later.
- FortiExtender must be connected to the internet.
- FortiExtender is managed by FortiGate, its status is Online, and the FortiExtender IP address is shown in FortiGate interfaces.

Example

In this example, a FortiManager administrator creates a firmware upgrade template for FortiExtender modem and assigns the template to the managed FortiGate with attached FortiExtender. When the FortiManager administrator uses the template to initiate an upgrade to the FortiExtender modem firmware, the template uses the FortiOS REST API to:

- Query the FortiGate for the current modem firmware version of the attached FortiExtender and the firmware versions available for FortiExtender on FortiCloud
- Direct FortiExtender to install a specific version of firmware from FortiCloud.

To use FortiManager to update FortiExtender modem firmware:

1. In FortiManager create a firmware upgrade template for FortiExtender modem and assign it to the managed FortiGate with attached FortiExtender. For details, see the FortiManager 7.4 New Features.
2. In FortiManager, use the template to initiate a FortiExtender modem firmware upgrade. The template uses the FortiOS REST API to query FortiExtender for the current modem firmware version.

```
https://<ip address>/api/v2/monitor/extender-controller/extender/modem-
firmware?serial=<number>

{
  "http_method":"GET",
  "results":{
    "available":[
      "FEM_EM06A-22-1-1"
    ],
```

```
    "current":"FEM_EM06A-22-1-1"
  },
  "vdom":"root",
  "path":"extender-controller",
  "name":"extender",
  "action":"modem-firmware",
  "status":"success",
  "serial":"<number>",
  "version":"v7.4.2",
  "build":2566
}
```

After receiving the API call, the following FortiOS command is run to provide the current and available FortiExtender firmware versions to FortiManager:

```
execute extender query-forticloud-mdmpkg-image all <serial number>

Local Modem Package:
FEM_07A-22-1-0-AMERICA

Versions on Cloud:
FEM_07A-22-2-0-AMERICA
```

3. After receiving the response from FortiGate, the FortiManager template automatically uses the FortiOS REST API to direct FortiExtender to download a specific firmware version from FortiCloud and install it.

```
POST /api/v2/monitor/extender-controller/extender/upgrade-modem-firmware
{
    "serial": <fext_serial>,
    "firmware-name": <name>
}
```

After receiving the API call, the following FortiOS command is run to download a specific firmware version from FortiCloud and install it to FortiExtender.

```
execute extender install-forticloud-mdm-package FEM_07A-22-2-0-AMERICA <serial number>
```

After the command is run on FortiGate, you can also use the FortiExtender console to view the progress of downloading and installing the modem firmware version.

```
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  229M  100  229M    0     0  2575k      0  0:01:31  0:01:31 --:--:-- 2744k

[MDM FW upgrade]: Decompress package...
Archive:  /tmp/upfile.zip
  inflating: SWI9X50C_01.14.20.00_VERIZON_002.058_000.nvu
  inflating: SWI9X50C_01.14.13.00.cwe
  inflating: SWI9X50C_01.14.20.00.cwe
  inflating: SWI9X50C_01.14.13.00_ATT_002.062_000.nvu
  inflating: SWI9X50C_01.14.03.00_US-CELLULAR_002.011_001.nvu
  inflating: SWI9X50C_01.14.13.00_GENERIC_002.048_000.nvu
  inflating: SWI9X50C_01.14.03.00_TMO_002.005_004.nvu
  inflating: SWI9X50C_01.14.03.00_TELUS_001.013_003.nvu
  inflating: SWI9X50C_01.14.03.00.cwe
  inflating: carrier_profile.conf
Starting modem firmware upgrade!
```

CLI system permissions - 7.4.2

Users now have the capability to exercise more granular control over CLI commands. This feature allows administrators to customize access to CLI commands based on their role, access level, or seniority, thereby enhancing both security and efficiency.

To configure CLI command access in administrative profiles:

```
config system accprofile
    edit <name>
        set cli-diagnose {enable | disable}
        set cli-get {enable | disable}
        set cli-show {enable | disable}
        set cli-exec {enable | disable}
        set cli-config {enable | disable}
    next
end
```

This command allows the administrator to configure the administrator profiles by enabling specific CLI commands as needed. The default setting for all the CLI command options is `disable`.

To edit an administrator profile, you must be logged in to an account with sufficient privileges, or as a super_admin user.

By default, the FortiGate has an administrator account that uses the super_admin profile. See Administrator profiles for more information.

Memory usage reduced on FortiGate models with 2 GB RAM - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- Fortinet Security Fabric

As part of improvements to reduce memory usage on FortiGate models with 2 GB RAM:

- FortiGate models with 2 GB RAM can be the root of the Security Fabric topology with a maximum of five downstream devices.
- FortiGate models can authorize a limited number of FortiExtender devices:
  - Two FortiExtenders for FortiGate 40F and 60E series devices and their variants
  - Six FortiExtenders for FortiGate 60F, 80E, and 90E series devices and their variants
- The memory footprint is reduced when running daemons, including Proxy/WAD, IPS engine, automation, and logging.
- The dynamic routing daemon only runs when required by the FortiGate configuration.

Models with reduced memory usage are the FortiGate 40F, 60E, 60F, 80E, and 90E series devices and their variants.

Prevent firmware upgrade depending on the current firmware license's expiration date - 7.4.2

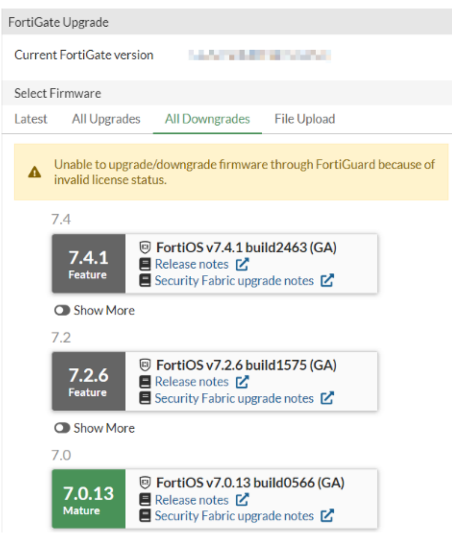
This information is also available in the FortiOS 7.4 Administration Guide:

- Prevent FortiGates with an expired support contract from upgrading to a major or minor firmware release

In FortiOS 7.4.2 and above, enforcement of an active FortiGate firmware license to allow firmware upgrades has been improved. Enforcement is based on the expiry date of the current firmware license compared to the release date of the first GA release of a major version. For example, for FortiOS 7.4.x firmware upgrades, enforcement is based on the expiry date of the current support contract compared to the release date of FortiOS 7.4.0 GA.

Therefore, upgrades between major, minor, and patch versions are only allowed if the firmware license is valid relative to the release date of the first GA release of a major version. If the firmware license expiry date is earlier than the firmware first GA major release date, then the firmware upgrade to that version will not be allowed. See the following Example on page 627.

In the *System > Firmware & Registration* page, until the support contract is renewed, FortiGuard upgrades will be unavailable; namely, the *Confirm and Backup Config* button will be grayed out. However, you will be able to view the FortiGate firmware images available on FortiGuard using *Latest*, *All Upgrades*, and *All Downgrades* tabs and this functionality will be restored upon support contract renewal.



Downgrades from one major version to another are not blocked because the FortiGate should have had a firmware expiry date that is later than the release date of the older firmware major version.

For example, if the firmware license expiry date was March 25, 2024, the FortiGate is currently running 7.4.2 and you wanted to downgrade to 7.2.7, since the release date of 7.2.0 GA was March 31, 2022 then this firmware downgrade would be allowed. The firmware license expiry date is later than the release date of the older firmware major version, 7.2.0 GA.

This new feature is an expansion of 7.4.0 and 7.4.1 new features. See Prevent FortiGates with an expired support contract from upgrading to a major or minor firmware release on page 611 and Prevent firmware upgrades when the support contract is expired using the GUI 7.4.1 on page 613 for more information on upgrading to major and minor versions.

Example

In this example, the release dates of major versions are as follows:

- 7.4.0 GA release on May 8, 2023
- 7.6.0 GA release on March 31, 2024
- 7.8.0 GA release on March 31, 2025

This example is using fictitious GA release dates of future versions for illustrative purposes only. These dates do not indicate the official FortiOS release schedule.

The following table demonstrates whether you can upgrade the target FortiGate firmware version depending on the current firmware license expiry date.

Firmware license expiry date	Is a FortiGate firmware upgrade allowed to the target version?		
	7.4.x	7.6.x	7.8.x
March 31, 2025 or later	Yes	Yes	Yes
March 25, 2025	Yes	Yes	No
March 25, 2024	Yes	No	No
May 2, 2023	No	No	No

High availability

This section includes information about HA related new features:

FGCP HA between FortiGates of the same model with different AC and DC PSUs

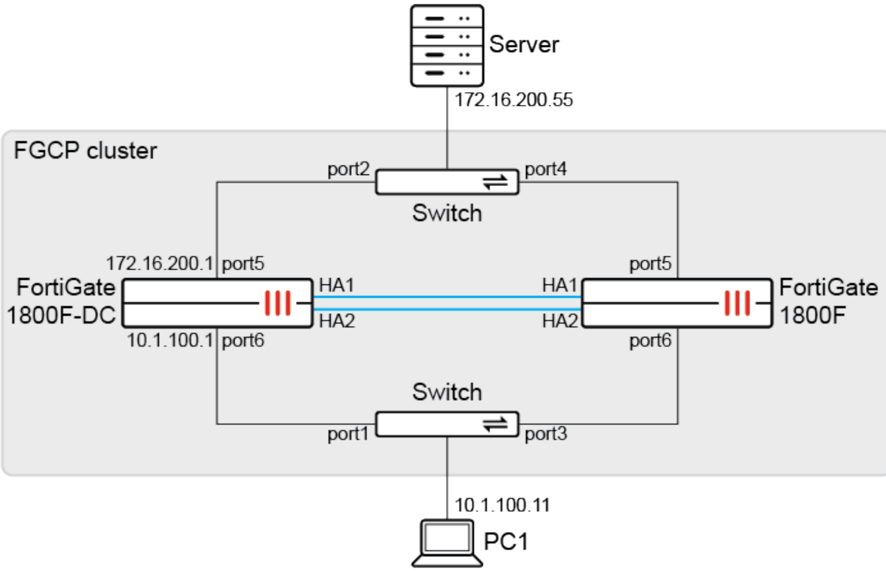
This information is also available in the FortiOS 7.4 Administration Guide:

- FGCP HA between FortiGates of the same model with different AC and DC PSUs

To improve power redundancy, FGCP HA clusters can support forming HA between units of the same model but with different AC PSU and DC PSU power supplies. This enables redundancy in a situation where power is completely lost on the AC grid, but traffic can fail over to a cluster member running on an independent DC grid.

The cluster members must be the same model with the same firmware installed, and must have the same hardware configuration other than the PSU.

In the following examples, there is an FGCP cluster with AC and DC PSU members: a FortiGate 1800F-DC (primary) and FortiGate 1800F (secondary).



Basic configuration

To configure the FGCP cluster in the GUI:

1. On the primary FortiGate (FG-1800F-DC), go to *System* > *HA*.
2. Configure the following settings:

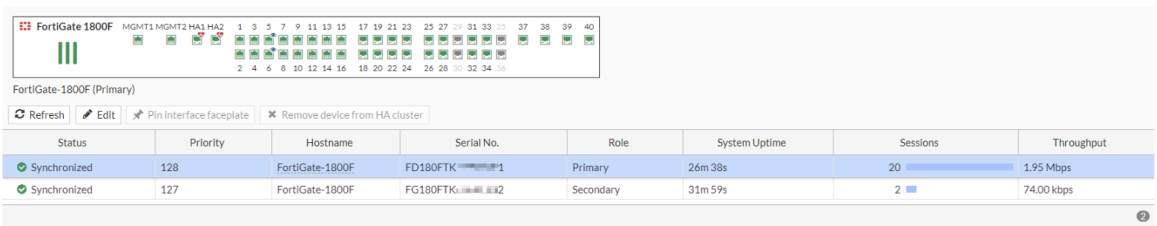
Mode	*Active-Passive*
Device priority	*128*
Group ID	*0*
Group name	*Example_cluster*
Password	Enter a password.
Session pickup	Enable this setting.
Monitor interfaces	Click the + to add *port5* and *port6*.
Heartbeat interfaces	Click the + to add *ha1* and *ha2*.

3. Click *OK*.
4. On the secondary FortiGate (FG-1800F), go to *System* > *HA*.
5. Configure the following settings:

Mode	*Active-Passive*

Device priority	*127*
Group ID	*0*
Group name	*Example_cluster*
Password	Enter a password.
Session pickup	Enable this setting.
Monitor interfaces	Click the + to add *port5* and *port6*.
Heartbeat interfaces	Click the + to add *ha1* and *ha2*.

6. Click *OK*.
7. Verify that the cluster status is *Synchronized*.



To configure the FGCP cluster in the CLI:

1. Configure the primary FortiGate (FG-1800F-DC):

```
config system ha
    set group-name "Example_cluster"
    set mode a-p
    set password **********
    set hbdev "ha2" 0 "ha1" 0
    set session-pickup enable
    set override disable
    set monitor "port5" "port6"
end
```

2. Configure the secondary FortiGate (FG-1800F):

```
config system ha
    set group-name "Example_cluster"
    set mode a-p
    set password **********
    set hbdev "ha2" 0 "ha1" 0
    set session-pickup enable
    set override disable
    set priority 127
    set monitor "port5" "port6"
end
```

3. Verify the cluster status on the primary FortiGate:

```
# get system ha status
HA Health Status: OK
Model: FortiGate-1800F
Mode: HA A-P
```

```
Group Name: Example_cluster
Group ID: 0
Debug: 0
Cluster Uptime: 0 days 0:56:11
Cluster state change time: 2023-05-29 19:11:14
Primary selected using:
    <2023/05/29 19:11:14> vcluster-1: FG180FTK*******1 is selected as the primary
because its uptime is larger than peer member FG180FTK*******2.
    <2023/05/29 18:59:45> vcluster-1: FG180FTK*******2 is selected as the primary
because its uptime is larger than peer member FG180FTK*******1.
    <2023/05/29 18:59:45> vcluster-1: FG180FTK*******1 is selected as the primary
because its override priority is larger than peer member FG180FTK*******2.
ses_pickup: enable, ses_pickup_delay=disable
override: disable
Configuration Status:
    FG180FTK*******1(updated 4 seconds ago): in-sync
    FG180FTK*******1 chksum dump: 95 4e 92 c3 39 75 8e 0e db 83 8d b7 b2 b1 9f 04
    FG180FTK*******2(updated 5 seconds ago): in-sync
    FG180FTK*******2 chksum dump: 95 4e 92 c3 39 75 8e 0e db 83 8d b7 b2 b1 9f 04
System Usage stats:
    FG180FTK*******1(updated 4 seconds ago):
        sessions=4, npu-sessions=0, average-cpu-user/nice/system/idle=0%/0%/0%/99%,
memory=22%
    FG180FTK*******2(updated 5 seconds ago):
        sessions=0, npu-sessions=0, average-cpu-user/nice/system/idle=0%/0%/0%/99%,
memory=22%
HBDEV stats:
    FG180FTK*******1(updated 4 seconds ago):
        ha2: physical/10000full, up, rx-bytes/packets/dropped/errors=18367581/33512/0/0,
tx=9563450/16609/0/0
        ha1: physical/10000full, up, rx-bytes/packets/dropped/errors=11543018/22166/0/0,
tx=12359673/22151/0/0
    FG180FTK*******2(updated 5 seconds ago):
        ha2: physical/10000full, up, rx-bytes/packets/dropped/errors=19133123/35087/0/0,
tx=10685583/18475/0/0
        ha1: physical/10000full, up, rx-bytes/packets/dropped/errors=17011332/25876/0/0,
tx=11919050/24991/0/0
MONDEV stats:
    FG180FTK*******1(updated 4 seconds ago):
        port5: physical/1000full, up, rx-bytes/packets/dropped/errors=988220/13742/0/0,
tx=106998000/73260/0/0
        port6: physical/1000full, up, rx-
bytes/packets/dropped/errors=107084264/73624/0/0, tx=953158/13611/0/0
    FG180FTK*******2(updated 5 seconds ago):
        port5: physical/1000full, up, rx-bytes/packets/dropped/errors=38194/128/0/0,
tx=0/0/0/0
        port6: physical/1000full, up, rx-bytes/packets/dropped/errors=99019/448/0/0,
tx=0/0/0/0
Primary     : FortiGate-1800F , FG180FTK*******1, HA cluster index = 1
Secondary   : FortiGate-1800F , FG180FTK*******2, HA cluster index = 0
number of vcluster: 1
vcluster 1: work 169.254.0.2
Primary: FG180FTK*******1, HA operating index = 0
Secondary: FG180FTK*******2, HA operating index = 1
```

4. Verify the cluster status on the secondary FortiGate:

```
# get system ha status
HA Health Status: OK
Model: FortiGate-1800F
Mode: HA A-P
Group Name: Example_cluster
Group ID: 0
Debug: 0
Cluster Uptime: 0 days 0:56:53
Cluster state change time: 2023-05-29 19:11:14
Primary selected using:
    <2023/05/29 19:11:14> vcluster-1: FG180FTK*******1 is selected as the primary
because its uptime is larger than peer member FG180FTK*******2.
    <2023/05/29 18:59:45> vcluster-1: FG180FTK*******2 is selected as the primary
because its uptime is larger than peer member FG180FTK*******1.
    <2023/05/29 18:55:03> vcluster-1: FG180FTK*******2 is selected as the primary
because it's the only member in the cluster.
    <2023/05/29 18:54:57> vcluster-1: FG180FTK*******2 is selected as the primary
because SET_AS_SECONDARY flag is set on peer member FG180FTK*******1.
ses_pickup: enable, ses_pickup_delay=disable
override: disable
...
Secondary   : FortiGate-1800F , FG180FTK*******2, HA cluster index = 0
Primary     : FortiGate-1800F , FG180FTK*******1, HA cluster index = 1
number of vcluster: 1
vcluster 1: standby 169.254.0.2
Secondary: FG180FTK*******2, HA operating index = 1
Primary: FG180FTK*******1, HA operating index = 0
```

Testing synchronization in the cluster

Based on the preceding example, the interface and firewall policy configurations are changed on the primary FortiGate. These configuration changes and sessions are synchronized to the secondary FortiGate. If the switch interface connected to the primary's port5 is down (port2), this triggers the monitor interface to be down, and the PC1 traffic will fail over to the secondary FortiGate.

To test configuration synchronization in the FGCP cluster:

1. Modify configurations on the primary FortiGate (FG-1800F-DC).
   a. Edit the interface settings:

   ```
   config system interface
       edit "port5"
           set ip 10.1.100.1 255.255.255.0
           set allowaccess ping https ssh http telnet
           set alias "To_Client_PC"
           config ipv6
               set ip6-address 2000:10:1:100::1/64
               set ip6-allowaccess ping https ssh http
           end
       next
       edit "port6"
           set ip 172.16.200.1 255.255.255.0
           set allowaccess ping https ssh http fgfm
           set alias "To_Server"
   ```

```
        config ipv6
            set ip6-address 2000:172:16:200::1/64
            set ip6-allowaccess ping https ssh http
        end
    next
end
```

b. Edit the firewall policy settings:

```
config firewall policy
    edit 1
        set name "to_server_policy"
        set srcintf "port5"
        set dstintf "port6"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set logtraffic-start enable
    next
end
```

2. On the secondary FortiGate (FG-1800F), verify that the settings were synchronized.

a. Verify the interface settings:

```
show system interface
    config system interface
        ...
        edit "port5"
            set vdom "root"
            set ip 10.1.100.1 255.255.255.0
            set allowaccess ping https ssh http telnet
            set type physical
            set alias "To_Client_PC"
            set snmp-index 9
            config ipv6
                set ip6-address 2000:10:1:100::1/64
                set ip6-allowaccess ping https ssh http
            end
        next
        edit "port6"
            set vdom "root"
            set ip 172.16.200.1 255.255.255.0
            set allowaccess ping https ssh http fgfm
            set type physical
            set alias "To_Server"
            set snmp-index 10
            config ipv6
                set ip6-address 2000:172:16:200::1/64
                set ip6-allowaccess ping https ssh http
            end
        next
    end
```

b. Verify the firewall policy settings:

```
show firewall policy
    config firewall policy
            edit 1
            set name "to_server_policy"
            set uuid 82a05e78-fe90-51ed-eb16-ee7bdea60de0
            set srcintf "port5"
            set dstintf "port6"
            set action accept
            set srcaddr "all"
            set dstaddr "all"
            set schedule "always"
            set service "ALL"
            set logtraffic-start enable
        next
    end
```

c. Verify the HA checksum:

```
# diagnose sys ha checksum show
is_manage_primary()=0, is_root_primary()=0
debugzone
global: 4e 15 af c3 c6 87 32 f5 69 5c b7 33 b1 8b 27 12
root: 4a 52 e4 f1 6a 2b eb 7d 84 7d f1 48 50 93 fe d9
all: 95 4e 92 c3 39 75 8e 0e db 83 8d b7 b2 b1 9f 04

checksum
global: 4e 15 af c3 c6 87 32 f5 69 5c b7 33 b1 8b 27 12
root: 4a 52 e4 f1 6a 2b eb 7d 84 7d f1 48 50 93 fe d9
all: 95 4e 92 c3 39 75 8e 0e db 83 8d b7 b2 b1 9f 04
```

To test session synchronization in the FGCP cluster:

1. On PC1, verify the IP address and gateway:

```
root@pc1:~# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:0c:29:a0:60:d6
          inet addr:10.1.100.11  Bcast:10.1.100.255  Mask:255.255.255.0
          ...

root@pc1:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.1.100.1      0.0.0.0         UG    0      0        0 eth1
10.1.100.0      0.0.0.0         255.255.255.0   U     0      0        0 eth1
10.6.30.0       0.0.0.0         255.255.255.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth0
```

2. Using Wget, initiate a large file download with HTTP that will maintain a long session:

```
root@pc1:~# wget http://172.16.200.55/big100MB.html   --keep-session-cookies --limit-
rate=128k --progress=dot -S -r --delete-after
--2023-05-29 14:55:33--  http://172.16.200.55/big100MB.html
Connecting to 172.16.200.55:80... connected.
HTTP request sent, awaiting response...
  HTTP/1.1 200 OK
  Date: Mon, 29 May 2023 21:55:41 GMT
  Server: Apache/2.4.18 (Ubuntu)
  Last-Modified: Thu, 01 Dec 2016 00:17:35 GMT
```

```
  ETag: "6126784-5428dbf967ad3"
  Accept-Ranges: bytes
  Content-Length: 101869444
  Vary: Accept-Encoding
  Keep-Alive: timeout=5, max=100
  Connection: Keep-Alive
  Content-Type: text/html
Length: 101869444 (97M) [text/html]
Saving to: '172.16.200.55/big100MB.html'

     0K .......... .......... .......... .......... ..........  0%  199K 8m18s
    50K .......... .......... .......... .......... ..........  0%  100K 12m26s
   100K .......... .......... .......... .......... ..........  0%  200K 11m3s
   150K .......... .......... .......... .......... ..........  0%  100K 12m25s
   200K .......... .......... .......... .......... ..........  0%  100K 13m14s
   250K .......... .......... .......... .......... ..........  0%  200K 12m24s
```

3. On the primary FortiGate (FG-1800F-DC), check the session information:

```
# diagnose sys session filter  dport  80
# diagnose sys session list

session info: proto=6 proto_state=01 duration=5 expire=3594 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu synced log-start
statistic(bytes/packets/allow_err): org=112/2/1 reply=60/1/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=13->14/14->13 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.1.100.11:54752->172.16.200.55:80(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.55:80->10.1.100.11:54752(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=15767 auth_info=0 chk_client_info=0 vd=0
serial=00000d80 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=9/9, ips_offload=0/0, epid=133/132, ipid=132/133,
vlan=0x0000/0x0000
vlifid=132/133, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=12/12
total session: 1
```

4. On the secondary FortiGate (FG-1800F), check that the session is synchronized:

```
# diagnose sys session filter dport 80
# diagnose sys session list

session info: proto=6 proto_state=01 duration=47 expire=3552 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=dirty may_dirty npu syn_ses
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
```

```
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=13->14/14->13 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.1.100.11:54752->172.16.200.55:80(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.55:80->10.1.100.11:54752(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=00000d80 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0,
vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:
total session: 1
```

To test failover in the FGCP cluster:

1. On the switch connected to port5 of the primary FortiGate, change port2's status to be down:

```
config switch physical-port
    edit port2
        set status down
    next
end
```

2. Check the HA status on the primary FortiGate (FG-1800F-DC), which now becomes the secondary device:

```
# get system ha status
HA Health Status:
    WARNING: FG180FTK*******1 has mondev down;
Model: FortiGate-1800F
Mode: HA A-P
Group Name: Example_cluster
Group ID: 0
Debug: 0
Cluster Uptime: 0 days 1:16:13
Cluster state change time: 2023-05-29 20:08:56
Primary selected using:
    <2023/05/29 20:08:56> vcluster-1: FG180FTK*******2 is selected as the primary
because the value 0 of link-failure + pingsvr-failure is less than peer member
FG180FTK*******1.
    <2023/05/29 19:11:14> vcluster-1: FG180FTK*******1 is selected as the primary
because its uptime is larger than peer member FG180FTK*******2.
    <2023/05/29 18:59:45> vcluster-1: FG180FTK*******2 is selected as the primary
because its uptime is larger than peer member FG180FTK*******1.
    <2023/05/29 18:59:45> vcluster-1: FG180FTK*******1 is selected as the primary
because its override priority is larger than peer member FG180FTK*******2.
ses_pickup: enable, ses_pickup_delay=disable
override: disable
...
Secondary   : FortiGate-1800F , FG180FTK*******1, HA cluster index = 1
Primary     : FortiGate-1800F , FG180FTK*******2, HA cluster index = 0
number of vcluster: 1
vcluster 1: standby 169.254.0.1
Secondary: FG180FTK*******1, HA operating index = 1
Primary: FG180FTK*******2, HA operating index = 0
```

3. Check the HA status on the new primary FortiGate (FG-1800F):

```
# get system ha status
HA Health Status:
    WARNING: FG180FTK*******1 has mondev down;
Model: FortiGate-1800F
Mode: HA A-P
Group Name: Example_cluster
Group ID: 0
Debug: 0
Cluster Uptime: 0 days 1:19:9
Cluster state change time: 2023-05-29 20:08:56
Primary selected using:
    <2023/05/29 20:08:56> vcluster-1: FG180FTK*******2 is selected as the primary
because the value 0 of link-failure + pingsvr-failure is less than peer member
FG180FTK*******1.
    <2023/05/29 19:11:14> vcluster-1: FG180FTK*******1 is selected as the primary
because its uptime is larger than peer member FG180FTK*******2.
    <2023/05/29 18:59:45> vcluster-1: FG180FTK*******2 is selected as the primary
because its uptime is larger than peer member FG180FTK*******1.
    <2023/05/29 18:55:03> vcluster-1: FG180FTK*******2 is selected as the primary
because it's the only member in the cluster.
ses_pickup: enable, ses_pickup_delay=disable
override: disable
...
Primary     : FortiGate-1800F , FG180FTK*******2, HA cluster index = 0
Secondary   : FortiGate-1800F , FG180FTK*******1, HA cluster index = 1
number of vcluster: 1
vcluster 1: work 169.254.0.1
Primary: FG180FTK*******2, HA operating index = 0
Secondary: FG180FTK*******1, HA operating index = 1
```

4. On PC1, verify that the HTTP traffic remains uninterrupted:

```
 ...
 74700K .......... .......... .......... .......... .......... 75%  100K 3m13s
 74750K .......... .......... .......... .......... .......... 75%  200K 3m13s
 74800K .......... .......... .......... .......... .......... 75%  100K 3m12s
 74850K .......... .......... .......... .......... .......... 75%  200K 3m12s
 74900K .......... .......... .......... .......... .......... 75%  100K 3m12s
 74950K .......... .......... .......... .......... .......... 75%  100K 3m11s
 75000K .......... .......... .......... .......... .......... 75%  200K 3m11s
 75050K .......... .......... .......... .......... .......... 75%  100K 3m10s
 75100K .......... .......... .......... .......... .......... 75%  200K 3m10s
 75150K .......... .......... .......... .......... .......... 75%  100K 3m10s
```

FGCP multi-version cluster upgrade - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- FGCP multi-version cluster upgrade

The FGCP multi-version cluster (MVC) upgrade mode allows manual control over the cluster member that is being upgraded. HA members can temporarily run in an MVC while administrators perform tests to confirm traffic can pass through the upgraded member smoothly.

The syntax of the existing upgrade mode has been updated:

```
config system ha
    set upgrade-mode {simultaneous | uninterruptible | local-only | secondary-only}
end
```

`upgrade-mode {simultaneous | uninterruptible | local-only | secondary-only}`	Set the mode to upgrade a cluster. • `simultaneous`: all HA members upgrade at the same time (previously `set uninterruptible-upgrade disable`). • `uninterruptible`: secondary HA members are upgraded first, followed by the primary member (previously `set uninterruptible-upgrade enable`). • `local-only`: only upgrade the local member in which the firmware is uploaded. • `secondary-only`: only upgrade the secondary members.

The `local-only` and `secondary-only` upgrade options are advanced configurations that should only be used to temporarily put the HA cluster in MVC operation mode. While in this operation, states and sessions (such as the session table and routing table) are synchronized, but configuration changes are not synchronized between cluster members in different builds. If more than two members are in the cluster, the configurations between members in the same builds will be synchronized. The configurations for the entire cluster will be synchronized once the upgrade process has completed.

How it works

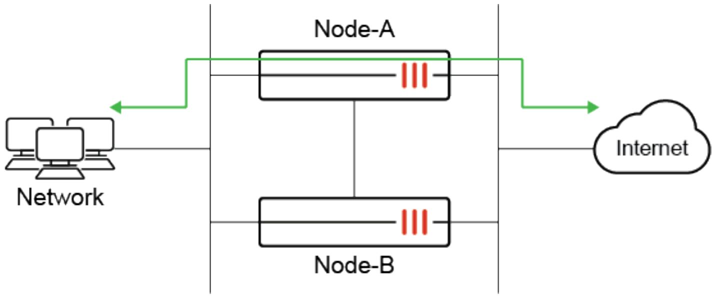
In `local-only` and `secondary-only` modes, the specific cluster member is upgraded and sessions are synchronized to it. The following tables show which members are upgraded based on the mode and where the upgrade is initiated.

local-only		
Upgrade method	**Outcome**	**Recommendation**
Initiate the upload or upgrade on the primary.	The primary member is upgraded.	Not recommended.
Initiate the upload or upgrade on the secondary member.	The secondary member where the image is uploaded is upgraded.	Recommended when selecting a specific HA member to upgrade.

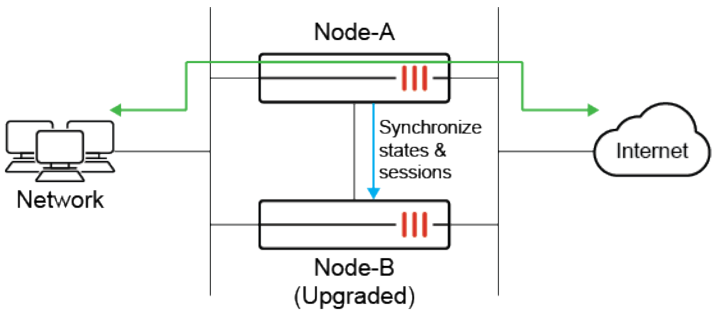
secondary-only		
Upgrade method	**Outcome**	**Recommendation**
Initiate the upload or upgrade on the primary.	All non-primary members are upgraded.	Recommended for scenarios where there is more than one secondary HA member.
Initiate the upload or upgrade on the secondary member.	The secondary member where the image is uploaded is upgraded.	Same result as initiating an upgrade on a secondary member in `local-only` mode.

This can apply to any HA clusters with two or more members. Administrators can initiate an upgrade on a secondary member by using its CLI console or accessing the device's GUI from its HA management interface.

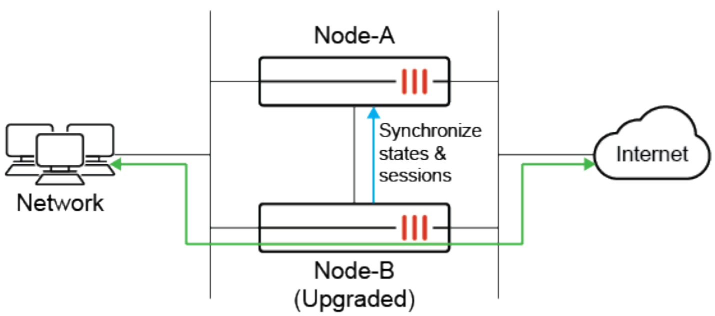
Initially, when you prepare an HA cluster in A-P mode for upgrade, traffic passes through the primary unit (Node-A) as the secondary unit (Node-B) sits on standby.



After the upgrade is completed on a secondary unit, states and sessions are synchronized. The members are now operating in MVC mode; however, traffic continues to pass through Node-A.



Administrators can manually trigger failover to make Node-B the new primary when ready. This can be done by resetting the HA uptime or changing device priorities, whichever method is desired. Traffic now passes through Node-B.

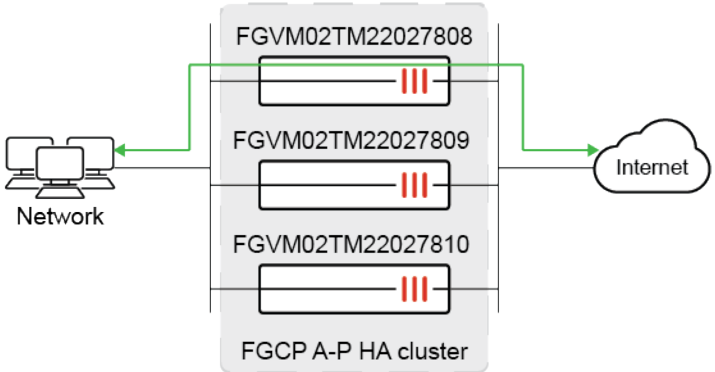


The upgraded system (Node-B) can be tested to verify that traffic can pass smoothly. If verification fails, administrators can trigger a failover to fail back to Node-A to avoid any downtime.

If verification is successful, administrators can manually trigger an upgrade on Node-A to bring the HA member up to the same version as Node-B to complete the HA upgrade procedure. This can be performed by accessing Node-A’s GUI from its HA management interface or using its CLI console.

Example 1: upgrade a single secondary member using the local-only upgrade option

In this example, three HA members are running in an FGCP A-P HA cluster.



The member FGVM02TM22027808 is acting as the primary and forwarding traffic. The member FGVM02TM22027810 is chosen for upgrade.

The cluster is originally running build 2456. The secondary unit is upgraded to build 2461. Fictitious build numbers are used in this example to demonstrate functionality of the feature.

To configure the HA cluster:

```
config system ha
    set group-id 260
    set group-name "kkk"
    set mode a-p
    set hbdev "port3" 0
    set session-pickup enable
    set upgrade-mode local-only
end
```

To perform the upgrade:

1. On the secondary member (FGVM02TM22027810), log in to the CLI console.
2. Execute a TFTP upgrade:

```
FGVM02TM22027810 # execute restore image tftp
/home/Images/FortiOS/v7.00/images/build2461/FGT_VM64-v7-build2461-FORTINET.out
172.16.100.71
This operation will replace the current firmware version!
Do you want to continue? (y/n)y

Please wait...

Connect to ftp server 172.16.100.71 ...
Get image from ftp server OK.
Verifying the signature of the firmware image.

Please wait for system to restart.
```

3. After the upgrade is complete, verify the version running on the secondary member:

```
FGVM02TM22027810 # get system status
Version: FortiGate-VM64 v7.4.1,build2461,230828 (interim)
…
```

4. On the primary unit, verify that HA is still formed between the three members:

```
FGVM02TM22027808 # diagnose sys ha dump-by group
<hatalk> vcluster_1: ha_prio=0(primary), state/chg_time/now=2
(work)/1692750721/1693262149
            HA information.
group-id=260, group-name='kkk'
has_no_aes128_gcm_sha256_member=0

gmember_nr=3
'FGVM02TM22027808': ha_ip_idx=2, hb_packet_version=10, last_hb_jiffies=0, linkfails=0,
weight/o=0/0, support_aes128_gcm_sha256=1
'FGVM02TM22027809': ha_ip_idx=1, hb_packet_version=12, last_hb_jiffies=51142842,
linkfails=3, weight/o=0/0, support_aes128_gcm_sha256=1
        hbdev_nr=1: port3(mac=000c..de, last_hb_jiffies=51142842, hb_lost=0),
'FGVM02TM22027810': ha_ip_idx=0, hb_packet_version=4, last_hb_jiffies=51142858,
linkfails=3, weight/o=0/0, support_aes128_gcm_sha256=1
        hbdev_nr=1: port3(mac=000c..1a, last_hb_jiffies=51142858, hb_lost=0),

vcluster_nr=1
vcluster-1: start_time=1692750718(2023-08-22 17:31:58), state/o/chg_time=2(work)/2
(work)/1692750721(2023-08-22 17:32:01)
        pingsvr_flip_timeout/expire=3600s/0s
        mondev: port1(prio=50,is_aggr=0,status=1) port7(prio=50,is_aggr=0,status=1)
port8(prio=50,is_aggr=0,status=1)
        'FGVM02TM22027808': ha_prio/o=0/0, link_failure=0, pingsvr_failure=0,
flag=0x00000001, mem_failover=0, uptime/reset_cnt=510868/0
        'FGVM02TM22027809': ha_prio/o=1/1, link_failure=0, pingsvr_failure=0,
flag=0x00000000, mem_failover=0, uptime/reset_cnt=510857/0
        'FGVM02TM22027810': ha_prio/o=2/2, link_failure=0, pingsvr_failure=0,
flag=0x00000000, mem_failover=0, uptime/reset_cnt=0/0
```

5. Fail over the HA cluster so that the secondary member, FGVM02TM22027810, becomes the primary. Since override is not enabled and the HA primary is determined by uptime, you can reset the HA uptime on the units that were not upgraded:

   ```
   # diagnose sys ha reset-uptime
   ```

6. Once verification on the upgraded member is successful, repeat step 2 to perform upgrades on the remaining units.

Example 2: upgrade multiple secondary members using the secondary-only upgrade option

Using the same topology as example 1, the three HA cluster members are originally running build 2456. Both secondary units are upgraded using the `secondary-only` upgrade option. Fictitious build numbers are used in this example to demonstrate functionality of the feature.

To configure the HA cluster:

```
config system ha
    set group-id 260
    set group-name "kkk"
    set mode a-p
    set hbdev "port3" 0
    set session-pickup enable
    set upgrade-mode secondary-only
end
```

To perform the upgrade:

1. On the primary unit (FGVM02TM22027808), log in to the CLI console.
2. Execute a TFTP upgrade:

```
FGVM02TM22027808 # execute restore image tftp
/home/Images/FortiOS/v7.00/images/build2461/FGT_VM64-v7-build2461-FORTINET.out
172.16.100.71
```

3. After the upgrade is complete, verify the version running on the secondary members.
   a. Member 1:

   ```
   FGVM02TM22027809 # get system status
   Version: FortiGate-VM64 v7.4.1,build2461,230828 (interim)
   …
   ```

   b. Member 2:

   ```
   FGVM02TM22027810 # get system status
   Version: FortiGate-VM64 v7.4.1,build2461,230828 (interim)
   …
   ```

4. On the primary unit, verify that HA is still formed between the three members:

```
FGVM02TM22027808 # diagnose sys ha dump-by group
            HA information.
group-id=260, group-name='kkk'
has_no_aes128_gcm_sha256_member=0

gmember_nr=3
'FGVM02TM22027808': ha_ip_idx=2, hb_packet_version=19, last_hb_jiffies=0, linkfails=0,
weight/o=0/0, support_aes128_gcm_sha256=1
'FGVM02TM22027809': ha_ip_idx=1, hb_packet_version=4, last_hb_jiffies=51358055,
linkfails=3, weight/o=0/0, support_aes128_gcm_sha256=1
        hbdev_nr=1: port3(mac=000c..de, last_hb_jiffies=51358055, hb_lost=0),
'FGVM02TM22027810': ha_ip_idx=0, hb_packet_version=5, last_hb_jiffies=51358057,
linkfails=3, weight/o=0/0, support_aes128_gcm_sha256=1
        hbdev_nr=1: port3(mac=000c..1a, last_hb_jiffies=51358057, hb_lost=0),

vcluster_nr=1
vcluster-1: start_time=1692750718(2023-08-22 17:31:58), state/o/chg_time=2(work)/2
(work)/1692750721(2023-08-22 17:32:01)
        pingsvr_flip_timeout/expire=3600s/0s
        mondev: port1(prio=50,is_aggr=0,status=1) port7(prio=50,is_aggr=0,status=1)
port8(prio=50,is_aggr=0,status=1)
        'FGVM02TM22027808': ha_prio/o=0/0, link_failure=0, pingsvr_failure=0,
flag=0x00000001, mem_failover=0, uptime/reset_cnt=512775/0
        'FGVM02TM22027809': ha_prio/o=2/2, link_failure=0, pingsvr_failure=0,
flag=0x00000000, mem_failover=0, uptime/reset_cnt=0/0
        'FGVM02TM22027810': ha_prio/o=1/1, link_failure=0, pingsvr_failure=0,
flag=0x00000000, mem_failover=0, uptime/reset_cnt=1/0
```

Enhance IPv6 VRRP state control - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- Ignore VRRP default route

State control for IPv6 Virtual Router Redundancy Protocol (VRRP) is enhanced. Previously, the VRRP state would be *Primary* as long as any route, including the default route, could reach the IPv6 VRRP destination. Now administrators can choose whether to exclude the default route from the calculation of available routes to the IPv6 VRRP destination to better manage and control the VRRP states.

```
config system interface
    edit < name >
        config ipv6
        config vrrp6
            edit < id >
                set ignore-default-route {enable | disable}
            next
        end
    end
end
```

`set ignore-default-route {enable | disable}`	Set the default route to be ignored: • `enable`: Ignore the default route when checking the VRRP destination. • `disable`: Include the default route when checking the VRRP destination.

Example

In this example, the IPv6 VRRP destination (`vrdst6`) is set with an IPv6 address of `2000:172:22:20::22`, and `ignore-default-route` is enabled for the destination. As long as non-default routes exist to the VRRP destination, the VRRP state is `Primary`. When only the default route to the VRRP destination exists, the VRRP state changes to `Backup`.

To ignore the default route when checking the IPv6 VRRP destination:

1. Enable the default route to be ignored for IPv6 VRRP.

   In the following example, the IPv6 VRRP destination (`vrdst6`) is set with an IPv6 address of `2000:172:22:20::22`, and `ignore-default-route` is enabled for the destination.

   ```
   config system interface
       edit "port2"
           config ipv6
               set vrrp-virtual-mac6 enable
               set vrip6_link_local fe80::926c:acff:2222:2222
               config vrrp6
                   edit 100
                       set vrgrp 100
                       set vrip6 2000:10:1:100::222
                       set priority 200
                       set vrdst6 2000:172:22:20::22
                       set ignore-default-route enable
                   next
               end
           end
       next
   end
   ```

2. Check the route for IPv6 VRRP destination.

The following example, the routing table shows an active route through `port1` to the IPv6 VRRP destination of `2000:172:22:20::22`. The active route is not a default route.

```
# get router info6 routing-table 2000:172:22:20::22
Routing entry for 2000:172:22:20::/80
  Known via "static", distance 10, metric 0
  Last update 00:00:15 ago
    via 2000:172:16:200::55, port1
```

3. Check VRRP group information for IPv6.

   In the following example, the VRRP state is `Primary` because non-default routes to the IPv6 VRRP destination exist as shown in the previous step.

```
# get router info6 vrrp
Interface: port2, primary IPv6 address: 2000:10:1:100::1
link-local IPv6 address: fe80::96f3:92ff:fe15:1ecd
Virtual link-local IPv6 address: fe80::926c:acff:2222:2222
  UseVMAC: 1, SoftSW: 0, EmacVlan: 0 BrPortIdx: 0, PromiscCount: 1
  HA mode: primary (0:0:1)
  VRT primary count: 1
  VRID: 100 version: 3
    vrip: 2000:10:1:100::222, priority: 200, state: PRIMARY
    adv_interval: 1, preempt: 1, ignore_dft: 0, start_time: 3
    primary_adv_interval: 100, accept: 1
    vrmac: 00:00:5e:00:02:64
    vrdst: 2000:172:22:20::22
    vrgrp: 100
```

4. Delete the non-default routes to the IPv6 VRRP destination (`vrdst6`), and check the routes again.

   In the following example, the routing table shows only the default route (`::/0`) is available to the IPv6 VRRP destination of `2000:172:22:20::22`.

```
# get router info6 routing-table 2000:172:22:20::22
Routing entry for ::/0
  Known via "static", distance 10, metric 0, best
  Last update 02:02:09 ago
  * via 2000:172:16:200::254, port1
```

5. Check VRRP group information for IPv6.

   In the following example, the VRRP state is `Backup` because only the default route is available to the IPv6 VRRP destination as shown in the previous step.

```
#get router info6 vrrp
Interface: port2, primary IPv6 address: 2000:10:1:100::1
link-local IPv6 address: fe80::96f3:92ff:fe15:1ecd
Virtual link-local IPv6 address: fe80::926c:acff:2222:2222
  UseVMAC: 1, SoftSW: 0, EmacVlan: 0 BrPortIdx: 0, PromiscCount: 0
  HA mode: primary (0:0:1)
  VRT primary count: 0
  VRID: 100 version: 3
    vrip: 2000:10:1:100::222, priority: 0, state: BACKUP
    adv_interval: 1, preempt: 1, ignore_dft: 1, start_time: 3 but
    primary_adv_interval: 100, accept: 1
    vrmac: 00:00:5e:00:02:64
    vrdst: 2000:172:22:20::22
    vrgrp: 100
```

SNMP

This section includes information about SNMP related new features:

- Add SNMP trap for memory usage on FortiGates 7.4.2 on page 644
- Add SNMP trap for PSU power restore 7.4.2 on page 646

Add SNMP trap for memory usage on FortiGates - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- Important SNMP traps

Both free memory usage and freeable memory of FortiGate devices can be monitored through the Simple Network Management Protocol (SNMP).

SNMP object identifier (OID) entries are available in Fortinet MIB files to show the percentage of free memory usage and freeable memory in an SNMP manager:

- `1.3.6.1.4.1.12356.101.4.1.36 .fgSysFreeMemUsage`
- `1.3.6.1.4.1.12356.101.4.1.37 .fgSysFreeableMemUsage`

The following commands are available to configure memory thresholds to trigger SNMP traps:

```
config system snmp sysinfo
    set trap-free-memory-threshold <integer>
    set trap-freeable-memory-threshold <integer>
end
```

`set trap-free-memory-threshold <integer>`	Use an integer from 1 to 100 (default 5) to identify what percentage of free memory usage will trigger an SNMP trap. SNMP traps are sent when the free memory is *lower* than the specified threshold. For example, the free memory threshold is set to 5, and SNMP traps are sent when free memory is lower than 5%.
`set trap-freeable-memory-threshold <integer>`	Use an integer from 1 to 100 (default 60) to identify what percentage of freeable memory will trigger an SNMP trap. SNMP traps are sent when the freeable memory is *higher* than the specified threshold. For example, the freeable memory threshold is set to 60, and SNMP traps are sent when freeable memory is higher than 60%.

Example

In this example, the SNMP agent is configured to monitor FortiGate memory and send traps. The `trap-free-memory-threshold` is set to `10`, and the `trap-freeable-memory-threshold` is set to `50`. SNMP traps are triggered for both thresholds because:

- The free memory on the FortiGate is 9%, which is lower than the threshold of 10.
- The freeable memory on the FortiGate is 56%, which is higher than the threshold of 50.

This example describes how to use the new commands to configure SNMP agents. It does not describe how to fully configure SNMP. For information about configuring SNMP, see the FortiOS 7.4 Administration Guide:

- Basic configuration

To configure SNMP for monitoring memory usage on FortiGates:

1. Configure the SNMP agent to monitor FortiGate memory usage and freeable memory.
   In this example, the `trap-free-memory-threshold` is set to `10`, and the `trap-freeable-memory-threshold` is set to `50`.
   ```
   config system snmp sysinfo
       set status enable
       set engine-id <string for local SNMP engine ID>
       set description <string>
       set contact-info <string>
       set location <string>
       set trap-high-cpu-threshold 60
       set trap-free-memory-threshold 10
       set trap-freeable-memory-threshold 50
   end
   ```
2. Verify that the SNMP manager can successfully query and receive a response on the current memory status of the FortiGate.
   In the following example, the free memory on the FortiGate is reported as 9%, and the freeable memory on the FortiGate is reported as 56%.
   ```
   # snmpwalk -v2c -c REGR-SYS 172.16.200.1 1.3.6.1.4.1.12356.101.4.1.36
   FORTINET-FORTIGATE-MIB::fgSystemInfo.36.0 = Gauge32: 9
   fosqa@pc05:~$ snmpwalk -v2c -c REGR-SYS 172.16.200.1 1.3.6.1.4.1.12356.101.4.1.37
   FORTINET-FORTIGATE-MIB::fgSystemInfo.37.0 = Gauge32: 56
   ```
3. Use the SNMP manager to monitor memory usage on the FortiGate.
   Following is an example of the SNMP trap messages sent when thresholds are surpassed for freeable memory and free memory usage on FortiGates:
   ```
   2023-12-08 19:53:14 172.16.200.1(via UDP: [172.16.200.1]:162->[172.16.200.55]:162) TRAP,
   SNMP v1, community REGR-SYS
           FORTINET-FORTIGATE-MIB::fgModel.1001 Enterprise Specific Trap (102) Uptime: 1
   day, 9:49:42.35
           FORTINET-CORE-MIB::fnSysSerial.0 = STRING: FG101FTK20006858     SNMPv2-
   MIB::sysName.0 = STRING: FGT_A    FORTINET-CORE-MIB::fnGenTrapMsg = STRING: freeable
   memory percentage is too high
   2023-12-08 19:56:33 <UNKNOWN> [UDP: [172.16.200.1]:162->[172.16.200.55]:162]:
   DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (12198187) 1 day, 9:53:01.87   SNMPv2-
   MIB::snmpTrapOID.0 = OID: FORTINET-CORE-MIB::fnTrapMemThreshold       FORTINET-CORE-
   MIB::fnSysSerial.0 = STRING: FG101FTK20006858     SNMPv2-MIB::sysName.0 = STRING: FGT_A
         FORTINET-CORE-MIB::fnGenTrapMsg = STRING: free memory percentage is too low
   ```

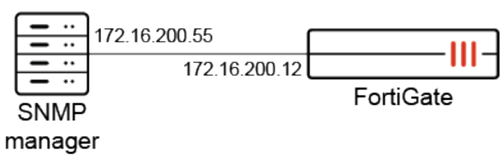
Add SNMP trap for PSU power restore - 7.4.2

An SNMP trap has been added for when power is restored to the power supply unit (PSU) on a FortiGate. When the PSU regains power after an outage, an SNMP trap should be triggered. This enhances the monitoring capabilities of the FortiGate.

In the GUI, the *snmp-event::power-supply-failure* event has been renamed to *snmp-event::power-supply*. In the CLI, the `power-supply-failure` event option has been renamed to `power-supply`.

Example

In this example, the `power-supply` event is applied in the SNMP community configuration. The SNMP trap messages are observed when the PSU cable is disconnected and reconnected.



To configure the SNMP community:

```
config system snmp community
    edit 1
        set name "1"
        config hosts
            edit 1
                set ip 1.1.1.1 255.255.255.255
            next
        end
        set events power-supply
    next
end
```

Sample log after the PSU cable is disconnected:

```
2: date=2023-11-06 time=11:34:03 eventtime=1699299242317192852 tz="-0800" logid="0100022106"
type="event" subtype="system" level="information" vd="vdom1" logdesc="Optional power supply
not detected" action="ipmc-sensor-monitor" status="failure" msg="PS2 Status not detected:
9.00"
```

Sample SNMP trap message after the PSU cable is disconnected:

```
2023-11-06 11:33:59 172.16.200.12(via UDP: [172.16.200.12]:162->[172.16.200.55]:162) TRAP,
SNMP v1, community REGR-SYS
        FORTINET-FORTIGATE-MIB::fgt2601F Enterprise Specific Trap (106) Uptime: 0:25:56.56
        FORTINET-CORE-MIB::fnSysSerial.0 = STRING: F2K61FTK22901112 SNMPv2-MIB::sysName.0 =
STRING: FGT_G FORTINET-CORE-MIB::fnGenTrapMsg = STRING: PS2 Status: not detected
2023-11-06 11:33:59 <UNKNOWN> [UDP: [172.16.200.12]:162->[172.16.200.55]:162]:
DISMAN-EXPRESSION-MIB::sysUpTimeInstance = Timeticks: (155656) 0:25:56.56 SNMPv2-
MIB::snmpTrapOID.0 = OID: FORTINET-CORE-MIB::fnTrapPowerSupplyFORTINET-CORE-
```

```
MIB::fnSysSerial.0 = STRING: F2K61FTK22901112 SNMPv2-MIB::sysName.0 = STRING: FGT_G
FORTINET-CORE-MIB::fnGenTrapMsg = STRING: PS2 Status: not detected
```

Sample log after the PSU cable is reconnected:

```
2: date=2023-11-06 time=11:28:52 eventtime=1699298932826382671 tz="-0800" logid="0100022115"
type="event" subtype="system" level="notice" vd="vdom1" logdesc="Power supply restored
notification" action="ipmc-sensor-monitor" status="success" msg="PS1 Status is normal"
```

Sample SNMP trap message after the PSU cable is reconnected:

```
2023-11-06 11:28:50 172.16.200.12(via UDP: [172.16.200.12]:162->[172.16.200.55]:162) TRAP,
SNMP v1, community REGR-SYS
        FORTINET-FORTIGATE-MIB::fgt2601F Enterprise Specific Trap (106) Uptime: 0:20:47.07
        FORTINET-CORE-MIB::fnSysSerial.0 = STRING: F2K61FTK22901112 SNMPv2-MIB::sysName.0 =
STRING: FGT_G FORTINET-CORE-MIB::fnGenTrapMsg = STRING: PS1 Status: restore
2023-11-06 11:28:50 <UNKNOWN> [UDP: [172.16.200.12]:162->[172.16.200.55]:162]:
DISMAN-EXPRESSION-MIB::sysUpTimeInstance = Timeticks: (124707) 0:20:47.07 SNMPv2-
MIB::snmpTrapOID.0 = OID: FORTINET-CORE-MIB::fnTrapPowerSupplyFORTINET-CORE-
MIB::fnSysSerial.0 = STRING: F2K61FTK22901112 SNMPv2-MIB::sysName.0 = STRING: FGT_G
FORTINET-CORE-MIB::fnGenTrapMsg = STRING: PS1 Status: restore
```

FortiGuard

This section includes information about FortiGuard related new features:

FortiGuard DLP service

This information is also available in the FortiOS 7.4 Administration Guide:
- FortiGuard DLP pattern service

The FortiGuard DLP service offers a database of predefined DLP patterns such as data types, dictionaries, and sensors. Example include:

- Drivers licenses for various countries, various states in the USA, and various provinces in Canada
- Tax numbers for various countries
- Credit card numbers
- Bank statements

When enabled, the DLP database (DLDB) is downloaded to the FortiGate and its predefined patterns can be configured in DLP profiles.

To configure DLP database updates:

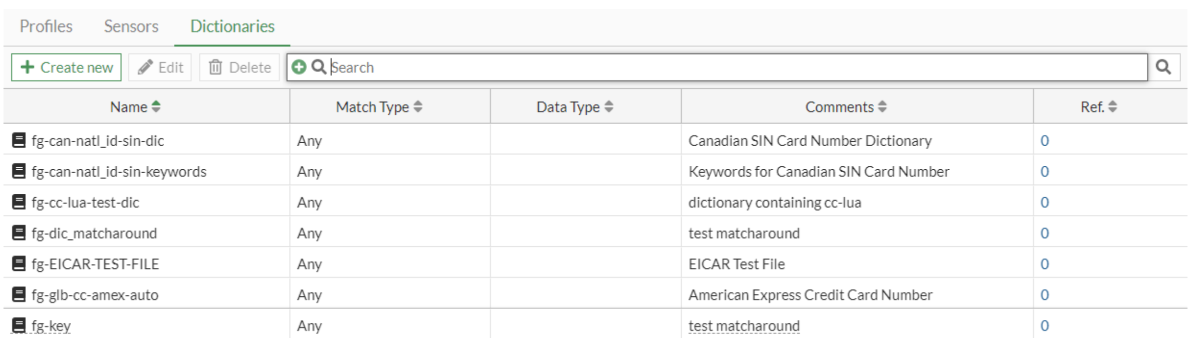
```
config system fortiguard
    set update-dldb {enable | disable}
end
```

To verify the database signature status:

```
# diagnose autoupdate versions
...
DLP Signature
---------
Version: 1.00010 signed
Contract Expiry Date: n/a
Last Updated using manual update on Fri Jan 27 15:25:00 2023
Last Update Attempt: Mon Jan 30 15:18:39 2023
Result: No Updates
```

Example

In this example, the administrator wants to look for data leakage of Canadian social insurance number (SIN) information and block this traffic. A DLP profile is created that uses the predefined dictionary, fg-can-natl_id-sin-dic, to check for Canadian Social Insurance Numbers (SINs).



Name	Match Type	Data Type	Comments	Ref.
fg-can-natl_id-sin-dic	Any		Canadian SIN Card Number Dictionary	0
fg-can-natl_id-sin-keywords	Any		Keywords for Canadian SIN Card Number	0
fg-cc-lua-test-dic	Any		dictionary containing cc-lua	0
fg-dic_matcharound	Any		test matcharound	0
fg-EICAR-TEST-FILE	Any		EICAR Test File	0
fg-glb-cc-amex-auto	Any		American Express Credit Card Number	0
fg-key	Any		test matcharound	0

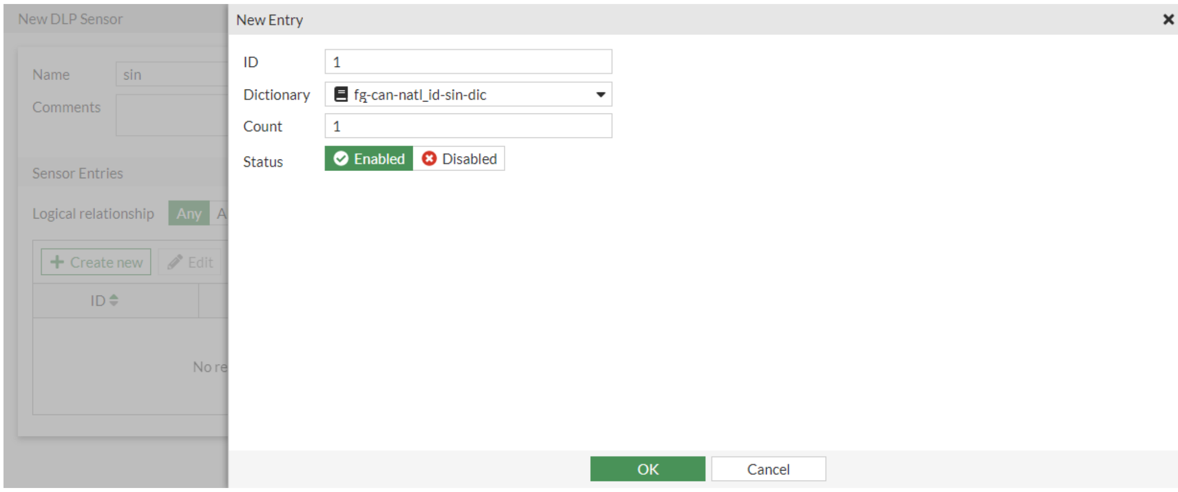
To verify that the Canadian SIN data type is added to the list of predefined data types:

```
show dlp data-type
config dlp data-type
    ...
    edit "fg-can-natl_id-proximity"
        set pattern "fortiguard dlp signature"
    next
end
```

To configure the DLP profile in the GUI:

1. Configure the DLP sensor using the predefined dictionary from FortiGuard:
   a. Go to *Security Profiles > Data Leak Prevention*, select the *Sensors* tab, and click *Create New*.
   b. Enter a name (*sin*).
   c. In the *Sensor Entries* section, click *Create New*.

d. Set the *Dictionary* to *fg-can-natl_id-sin-dic* and click *OK*.



e. Click *OK* to save the sensor.

2. Configure the DLP profile:

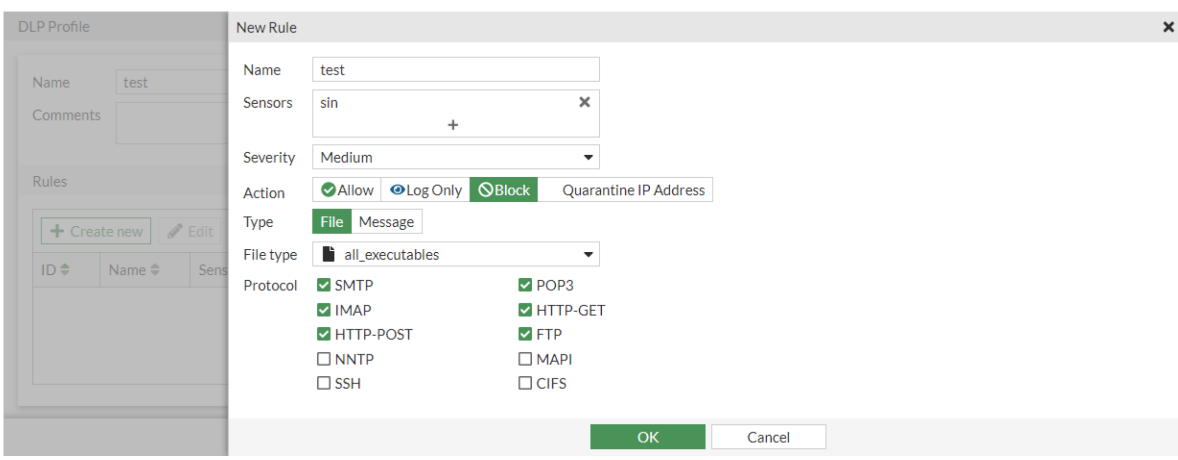
a. Go to *Security Profiles > Data Leak Prevention*, select the *Profiles* tab, and click *Create New*.

b. Enter a name (*test*).

c. In the *Rules* section, click *Create New*.

d. Configure the following settings:

Name	*test*
Sensors	*sin*
Severity	*Medium*
Action	*Block*
Type	*File*
File type	*all_executables*
Protocol	*SMTP*, *POP3*, *IMAP*, *HTTP-GET*, *HTTP-POST*, *FTP*



e. Click *OK*.

f. Click *OK* to save the profile.

To configure the DLP profile in the CLI:

1. Configure the DLP sensor using the predefined dictionary from FortiGuard:

```
config dlp sensor
    edit "sin"
        config entries
            edit 1
                set dictionary "fg-can-natl_id-sin-dic"
            next
        end
    next
end
```

2. Configure the DLP profile:

```
config dlp profile
    edit "test"
        set feature-set proxy
        config rule
            edit 1
                set name "test"
                set proto smtp pop3 imap http-get http-post ftp
                set filter-by sensor
                set file-type 2
                set sensor "sin"
                set action block
            next
        end
    next
end
```

Attack Surface Security Rating service - 7.4.1

The following table provides an overview of changes to the Security Rating service entitlement starting in 7.4.1:

7.4.0 and earlier	7.4.1 and later
Security Rating entitlement Includes: • PSIRT/Outbreak Package Definitions • Checking all the PSIRT/Outbreak rules in Security Rating • Running all the built-in free and paid security rating rules	**Attack Surface Security Rating entitlement** Includes: • Running all the built-in free and paid security rating rules • Checking all the Outbreak rules in Security Rating • Displaying CIS compliance information • IoT Detection Definitions • IoT Query
Firmware entitlement Includes: • Application Control Signatures • Device & OS Identification • Internet Service Database Definitions	**Firmware entitlement*** Includes: • Application Control Signatures • Device & OS Identification • Internet Service Database Definitions • PSIRT Package Definitions

7.4.0 and earlier	7.4.1 and later
	• Checking all PSIRT rules in Security Rating
IoT Detection service Includes: • IoT Detection Definitions • IoT Query	n/a

[*] The list is not exhaustive and does not include services such as FortiGate Virtual Patch Signatures, Inline-CASB, and SaaS Application Definitions.

Re-position the PSIRT packages into the Firmware entitlement

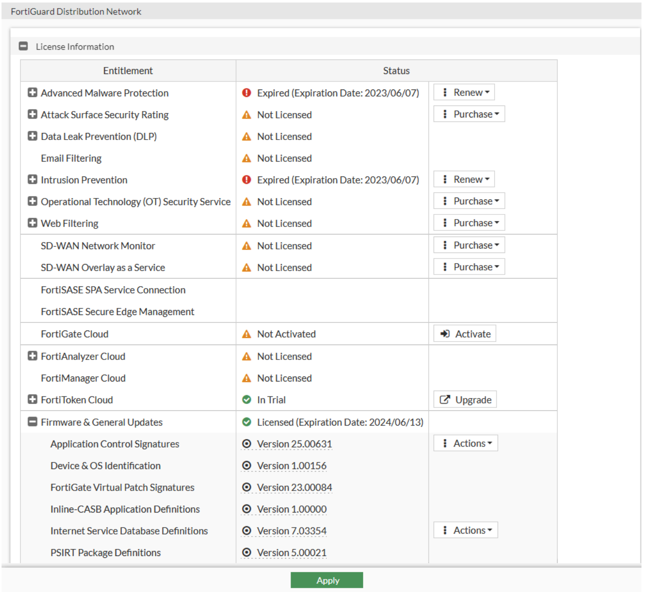
Starting in 7.4.1, PSIRT related packages and functionalities are re-positioned from the Security Rating entitlement into the Firmware entitlement. This allows more customers with the basic Firmware entitlement to have access to the latest PSIRT package updates, which can be executed under *Security Fabric > Security Rating > Security Posture* checks.

Devices with different entitlements can expect the following behaviors:

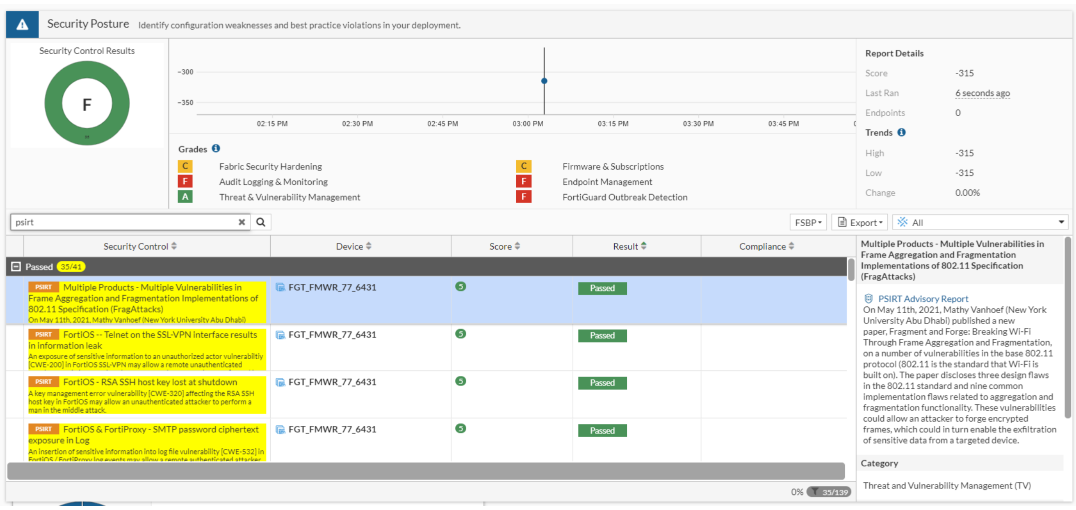
Entitlement		Action			
Firmware (FMWR)	**Attack Surface Security Rating (FGSA)**	**Download PSIRT package from FortiGuard**	**Run PSIRT security rating checks**	**Run built-in paid security rating checks**	**Run built-in free security rating checks**
Yes	No	Yes	Yes	No	Yes
Yes	Yes	Yes	Yes	Yes	Yes
No	No	No	No	No	Yes
No	Yes	No	No	Yes	Yes

Example 1: device with Firmware entitlement, but no Attack Surface Security Rating entitlement

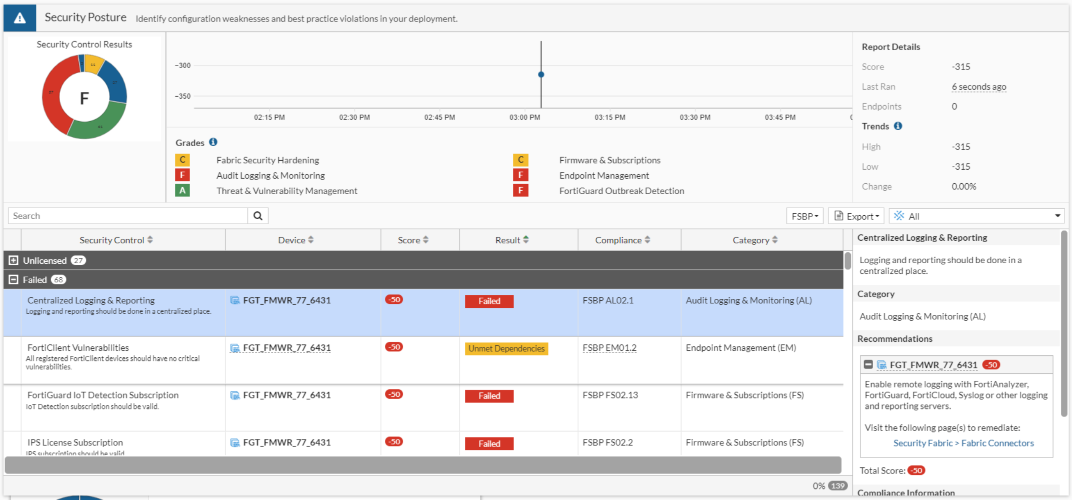
On the *System > FortiGuard* page, note that *Firmware & General Updates* is licensed, but *Attack Surface Security Rating* is not.



PSIRT-related rules can be executed from the *Security Fabric > Security Rating > Security Posture* page.

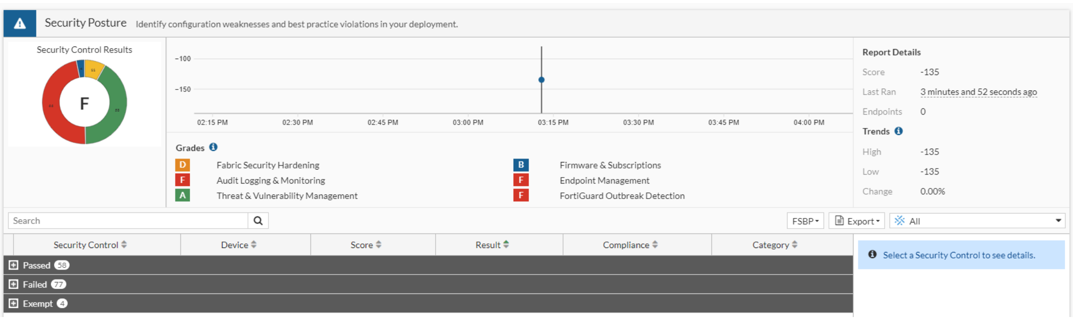


Free built-in security rating rules can be run. Other paid rules cannot be run, which fall under the *Unlicensed* category.



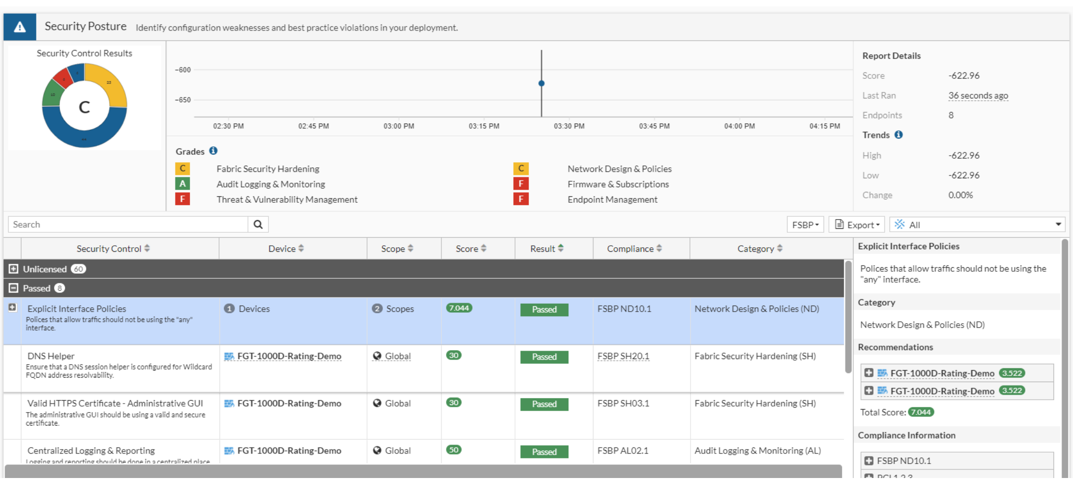
Example 2: device with both Firmware and Attack Surface Security Rating entitlements

In this scenario, all PSIRT, Outbreak, paid, and free rules can be run. There is no *Unlicensed* rule category.



Example 3: device with no Firmware or Attack Surface Security Rating entitlement

In this scenario, only free built-in rules can be run. Other rules are grouped under the *Unlicensed* category.



Merge the IoT Detection service into the Attack Surface Security Rating service

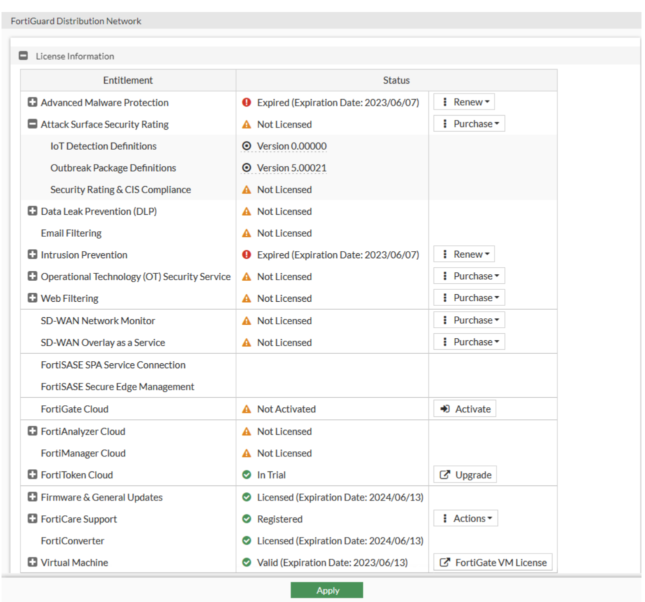
Starting in 7.4.1, the IoT Detection service, which includes IoT Detection Definitions (APDB) and the IoT Query service (IOTH), is merged into the Attack Surface Security Rating service (FGSA).

The following table provides a breakdown of the entitlements before and after upgrading:

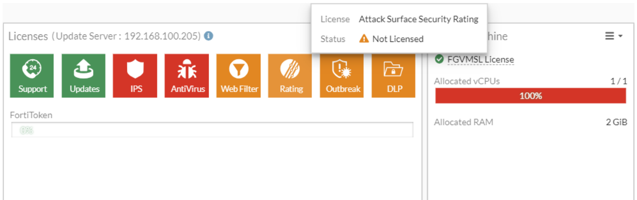
Before upgrading		After upgrading	
Entitlement	**Licensed**	**Entitlement**	**Licensed**
Security Rating	Yes	Attack Surface Security Rating	Yes
IoT Detection	Yes		Yes, for IoT Detection subcategory
Security Rating	Yes	Attack Surface Security Rating	Yes
IoT Detection	No		Yes, for IoT Detection subcategory
Security Rating	No	Attack Surface Security Rating	No
IoT Detection	Yes		Yes, for IoT Detection subcategory
Security Rating	No	Attack Surface Security Rating	No
IoT Detection	No		No, for IoT Detection subcategory

Example 1: device does not have an Attack Surface Security Rating entitlement

On the *System > FortiGuard* page, note that *Attack Surface Security Rating* is not licensed, and *IoT Detection Definitions* was not downloaded.

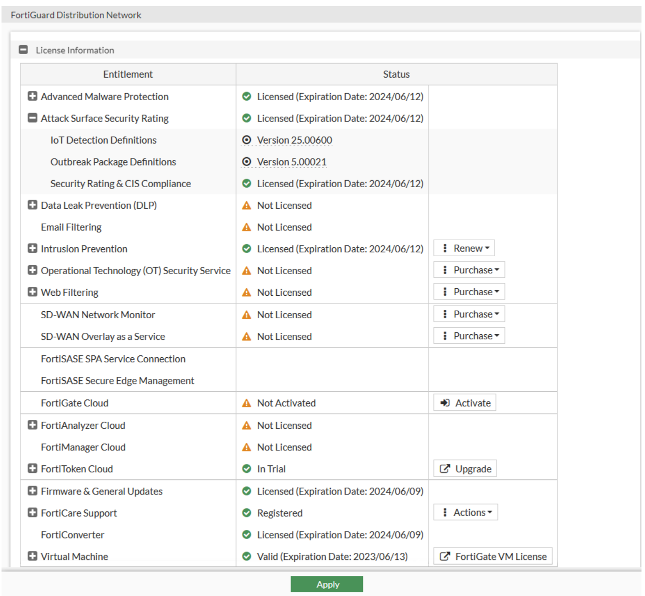


In the *Dashboard > Status > Licenses* widget, hovering over the *Rating* icon displays a tooltip that the status of *Attack Surface Security Rating* is *Not Licensed*.



Example 2: device has an Attack Surface Security Rating entitlement

On the *System > FortiGuard* page, note that *Attack Surface Security Rating* is licensed, and *IoT Detection Definitions* is downloaded.



FortiGuard Distribution Network

License Information

Entitlement	Status	
Advanced Malware Protection	Licensed (Expiration Date: 2024/06/12)	
Attack Surface Security Rating	Licensed (Expiration Date: 2024/06/12)	
IoT Detection Definitions	Version 25.00600	
Outbreak Package Definitions	Version 5.00021	
Security Rating & CIS Compliance	Licensed (Expiration Date: 2024/06/12)	
Data Leak Prevention (DLP)	Not Licensed	
Email Filtering	Not Licensed	
Intrusion Prevention	Licensed (Expiration Date: 2024/06/12)	Renew
Operational Technology (OT) Security Service	Not Licensed	Purchase
Web Filtering	Not Licensed	Purchase
SD-WAN Network Monitor	Not Licensed	Purchase
SD-WAN Overlay as a Service	Not Licensed	Purchase
FortiSASE SPA Service Connection		
FortiSASE Secure Edge Management		
FortiGate Cloud	Not Activated	Activate
FortiAnalyzer Cloud	Not Licensed	
FortiManager Cloud	Not Licensed	
FortiToken Cloud	In Trial	Upgrade
Firmware & General Updates	Licensed (Expiration Date: 2024/06/09)	
FortiCare Support	Registered	Actions
FortiConverter	Licensed (Expiration Date: 2024/06/09)	
Virtual Machine	Valid (Expiration Date: 2023/06/13)	FortiGate VM License

Apply

To view the definitions and license information in the CLI:

1. Verify the IoT definition version and update status:

```
# diagnose autoupdate versions | grep IoT -A 6
IoT Detect Definitions
---------
Version: 25.00600 signed
Contract Expiry Date: n/a
Last Updated using manual update on Fri Jul 14 11:12:19 2023
Last Update Attempt: Fri Jul 14 11:12:19 2023
Result: Updates Installed
```

2. Verify the Attack Surface Security Rating (FGSA) license and IoT detection service object:

```
# diagnose test update info
…
System contracts:
…
FGSA,Thu Jun 13 17:00:00 2024
```

```
…
Object versions:
…
07004000IOTD00105-00025.00600-2307121926
…
```

Operational Technology Security Service - 7.4.1

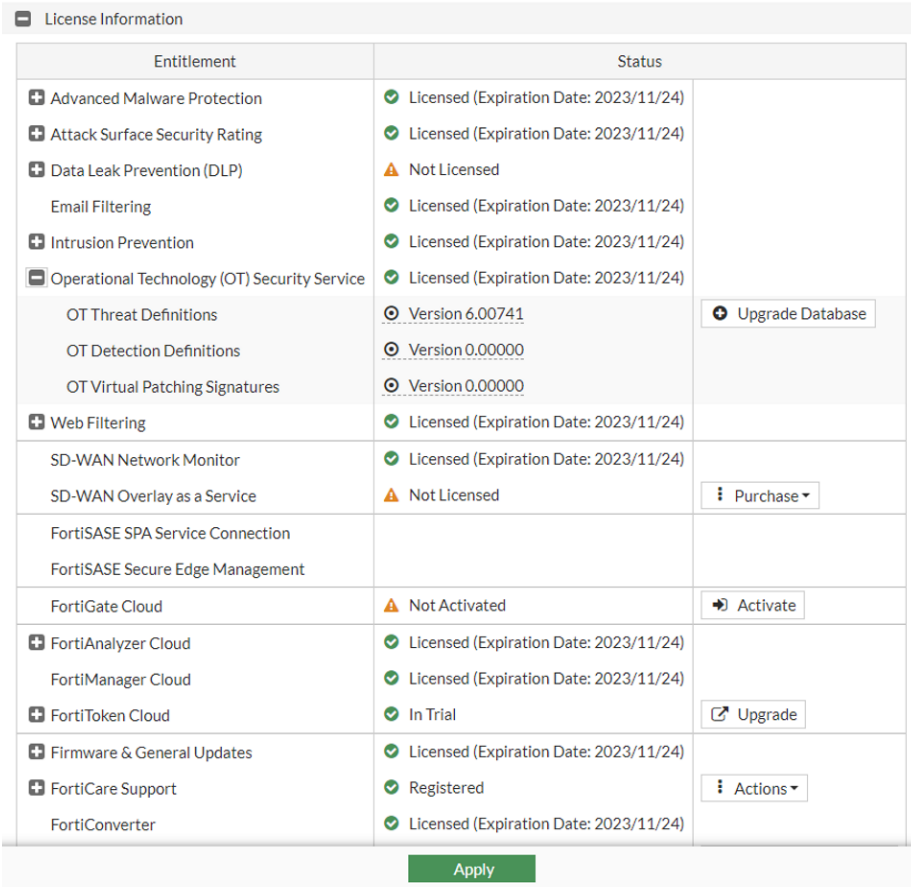
The Operational Technology (OT) Security Service is introduced to help consolidate OT services under one license and to decouple the underlying definitions and packages from IoT ones. New OT-related services such as OT Detection Definitions and OT Virtual Patching Signatures used in the virtual patching profile are now licensed under the OT Security Service.

The following table provides an overview of the new Operational Technology (OT) Security Service entitlement:

7.4.0 and earlier	7.4.1 and later
Industrial Security Service entitlement Includes: • Industrial Attack Definitions	**Operational Technology (OT) Security Service entitlement** Includes: • OT Threat Definitions (renamed) • OT Detection Definitions (new) • OT Virtual Patching Signatures (new)

To view the entitlement information in the GUI:

1. Go to *System > FortiGuard*.
2. Expand the *Operational Technology (OT) Security Service* entry in the *License Information* table.



To view the entitlement information in the CLI:

```
# diagnose autoupdate versions | grep OT -A7
OT Threat Definitions
---------
Version: 6.00741 signed
Contract Expiry Date: Sat Sep 16 2023
Last Updated using manual update on Tue Dec  1 02:30:00 2015
Last Update Attempt: n/a
Result: Updates Installed

--
OT Detect Definitions
---------
Version: 0.00000
Contract Expiry Date: Sat Sep 16 2023
Last Updated using manual update on Mon Jan  1 00:00:00 2001
Last Update Attempt: Mon Aug 14 15:42:43 2023
Result: No Updates

OT Patch Definitions
---------
Version: 0.00000
Contract Expiry Date: Sat Sep 16 2023
Last Updated using manual update on Mon Jan  1 00:00:00 2001
```

```
Last Update Attempt: Mon Aug 14 15:42:43 2023
Result: No Updates
```

OT Threat Definitions

Users upgrading to 7.4.1 from previous FortiOS versions with an Industrial Security Service entitlement will continue to receive the OT Security Service entitlement. The existing Industrial Attack Definitions have been renamed OT Threat Definitions. These definitions include both application control and IPS signatures for OT applications and protocols.

To include or exclude the use of OT signatures in IPS and application control:

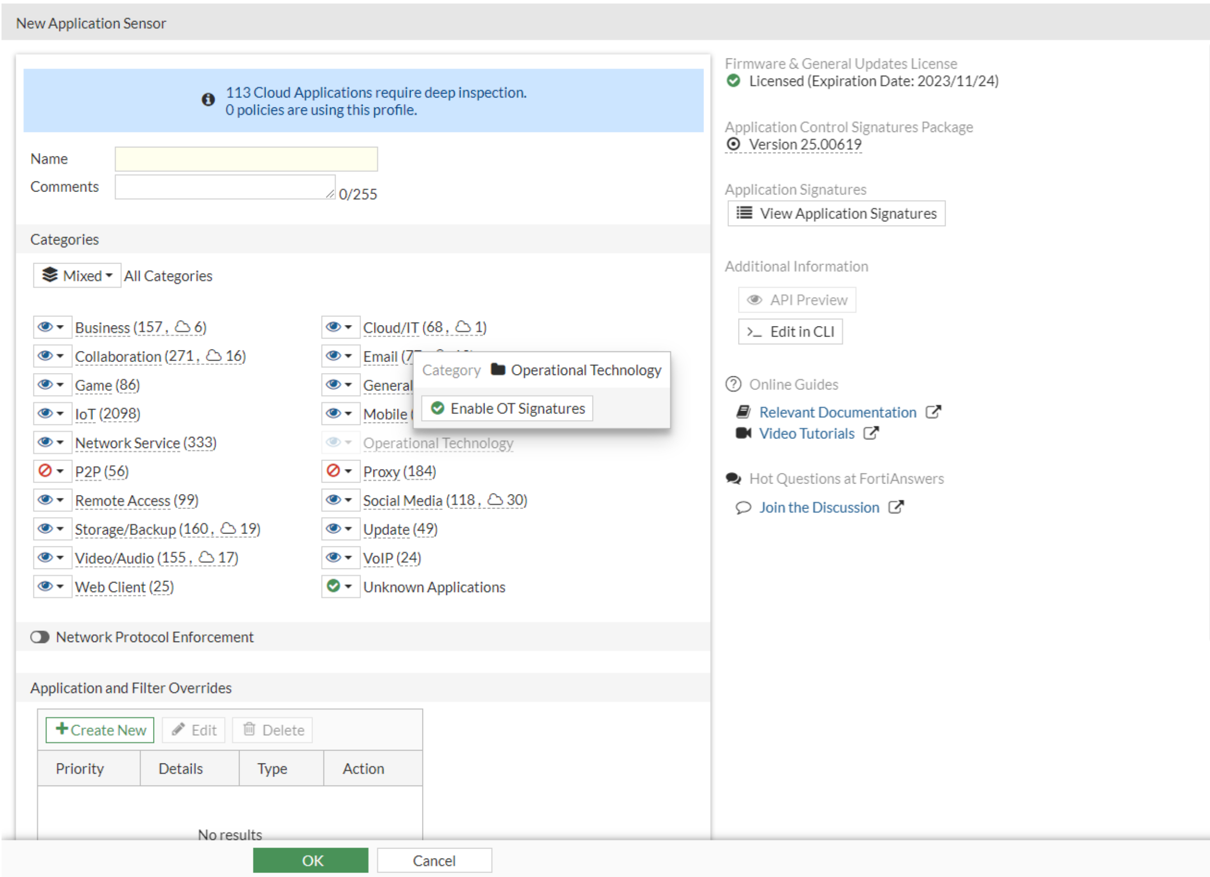
```
config ips global
    set exclude-signatures {none | ot}
end
```

The `exclude-signatures` setting's `industrial` option was renamed to `ot` in 7.4.1. Previously, the command options were:

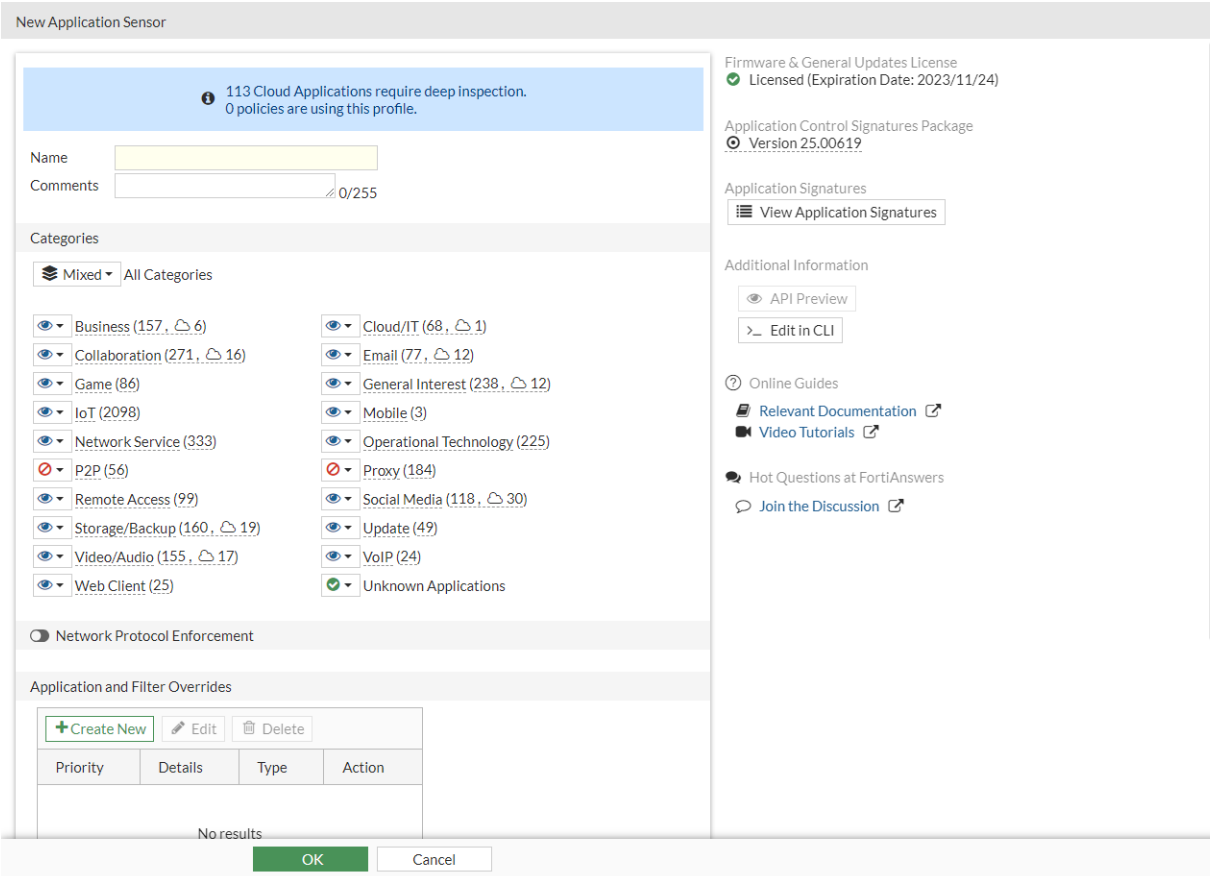
```
config ips global
    set exclude-signatures {none | industrial}
end
```

To apply the OT category to an application control sensor:

1. Go to *Security Profiles > Application Control*.
2. Click *Create New* or edit an existing profile.
3. If the OT category has not been enabled yet, hover over *Operational Technology* and click *Enable OT Signatures*.



4. The *Confirm* dialog opens, noting that *This will enable operational technology signatures globally. Are you sure you wish to proceed?* Click *OK*.
5. Select the action from the dropdown for the *Operational Technology* category.

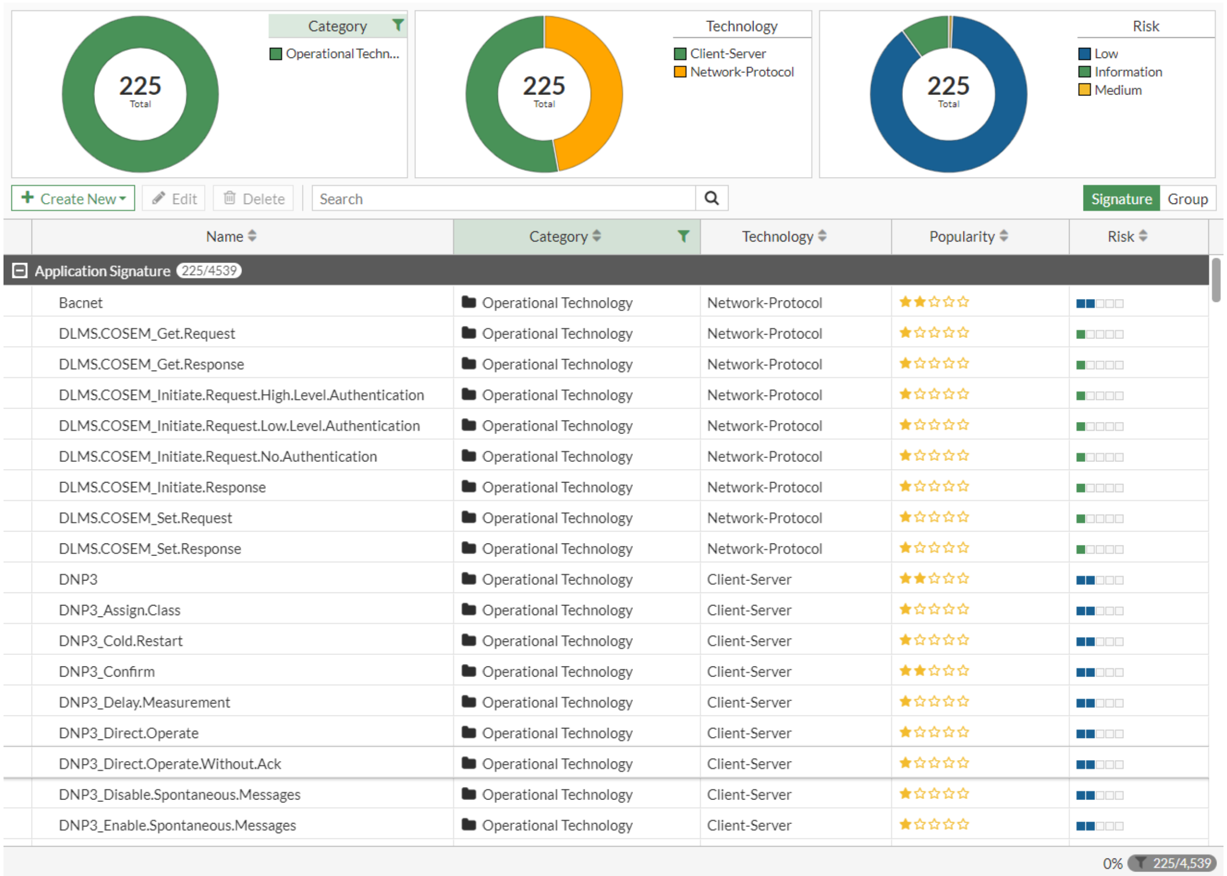


In FortiOS 7.4.1 and later, the *Industrial* category is renamed to *Operational Technology*.

6. Configure the other application sensor settings as needed.
7. Click *OK*.

To view OT application signatures:

1. Go to *Security Profiles > Application Signatures*.
2. In the *Category* column, click the funnel icon and filter by *Operational Technology*, then click *Apply*.



Support automatic federated firmware updates of managed FortiAPs and FortiSwitches - 7.4.1

When the automatic firmware updates setting is enabled, in addition to an automatic federated upgrade being performed on the FortiGate, automatic federated upgrades are now performed on any managed FortiAPs and FortiSwitches. The federated upgrades of these LAN edge devices adhere to the FortiOS-FortiAP and FortiOS-FortiSwitch compatibility matrix information maintained on the FortiGuard Distribution Network (FDN).

Example 1: FortiAP

In this example, automatic firmware updates are enabled on a FortiGate that is running 7.4.0. The FortiGate and two FortiAPs with older firmware are upgraded after the federated update.

To configure automatic federated firmware updates:

```
config system fortiguard
    set auto-firmware-upgrade enable
    set auto-firmware-upgrade-day sunday monday tuesday wednesday thursday friday saturday
    set auto-firmware-upgrade-delay 0
    set auto-firmware-upgrade-start-hour 17
    set auto-firmware-upgrade-end-hour 19
end
```

The auto-upgrade time is scheduled daily, between 5:00 p.m. and 7:00 p.m.

To verify that the federated update occurs:

1. Verify that the update is scheduled:

```
FortiGate-401F (global) # diagnose test application forticldd 13
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled.
    Next upgrade check scheduled at (local time) Tue Sep 12 17:25:03 2023
```

2. Verify the current firmware versions of the devices.
   a. For the FortiGate:

   ```
   FortiGate-401F # get system status | grep Version
   Version: FortiGate-401F v7.4.0,build2360,230509 (GA.F)
   ```

   b. For the FortiAPs:

   ```
   FortiGate-401F (root) # get wireless wtp-status connection-state
   Managed-devices in current vdom root:
       wtp-id                    : FP223E5519001619
       software-version : FP223E-v7.2-build0317
       connection-state : Connected
       wtp-id                    : FP231FTF23046483
       software-version : FP231F-v7.2-build0318
       connection-state : Connected
   ```

3. Verify the compatibility matrix:

```
FortiGate-401F (global) # diagnose test application forticldd 15
Last update: 1573 secs ago

FP223E:    7.4.0 b529 07004000FIMG0504204000 (FGT Version 7.4.1 b0)
FP231F:    7.4.0 b540 07004000FIMG0505804000 (FGT Version 7.4.1 b0)
```

4. Verify the installation schedule after the patch update is detected:

```
FortiGate-401F (global) # diagnose test application forticldd 13
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled.
    Next upgrade check scheduled at (local time) Wed Sep 13 17:11:50 2023
    New image 7.4.1b2463(07004000FIMG0030404001) installation is scheduled to
        start at Wed Sep 13 17:04:47 2023
        end by Wed Sep 13 19:00:00 2023
```

5. Verify which devices will be included in the federated update:

```
FortiGate-401F (global) # show system federated-upgrade
config system federated-upgrade
    set status initialized
    set upgrade-id 1
    config node-list
        edit "FG4H1FT922901903"
            set timing immediate
```

```
            set maximum-minutes 115
            set setup-time 00:04 2023/09/14 UTC
            set upgrade-path 7-4-1
        next
        edit "FP223E5519001619"
            set timing immediate
            set maximum-minutes 115
            set setup-time 00:04 2023/09/14 UTC
            set upgrade-path 7-4-1
            set device-type fortiap
            set coordinating-fortigate "FG4H1FT922901903"
        next
        edit "FP231FTF23046483"
            set timing immediate
            set maximum-minutes 115
            set setup-time 00:04 2023/09/14 UTC
            set upgrade-path 7-4-1
            set device-type fortiap
            set coordinating-fortigate "FG4H1FT922901903"
        next
    end
end
```

6. Wait for the FortiGate to perform the federated update.
7. After the federated update is complete, verify that the devices were upgraded to the latest version.
   a. For the FortiGate:

   ```
   FortiGate-401F # get system status | grep Version
   Version: FortiGate-401F v7.4.1,build2463,230830 (GA.F)
   ```

   b. For the FortiAPs:

   ```
   FortiGate-401F (root) # get wireless wtp-status connection-state
       wtp-id                    : FP223E5519001619
       software-version : FP223E-v7.4-build0529
       connection-state : Connected
       wtp-id                    : FP231FTF23046483
       software-version : FP231F-v7.4-build0540
       connection-state : Connected
   ```

Example 2: FortiSwitch

In this example, automatic firmware updates are enabled on a FortiGate that is running 7.4.1. Two FortiSwitches with older firmware are upgraded after the federated update.

To configure automatic federated firmware updates:

```
config system fortiguard
    set auto-firmware-upgrade enable
    set auto-firmware-upgrade-day tuesday
    set auto-firmware-upgrade-delay 0
    set auto-firmware-upgrade-start-hour 11
    set auto-firmware-upgrade-end-hour 12
end
```

The auto-upgrade time is scheduled on Tuesday, between 11:00 a.m. and 12:00 p.m.

To verify that the federated update occurs:

1. Verify that the update is scheduled:

```
FGT_A (global) # diagnose test application forticldd 13
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled.
        Next upgrade check scheduled at (local time) Tue Sep  5 11:06:58 2023
```

2. Verify if there are managed FortiSwitches that can be upgraded:

```
FGT_A (vdom1) # execute switch-controller get-conn-status
Managed-devices in current vdom vdom1:

FortiLink interface : flink
SWITCH-ID         VERSION           STATUS         FLAG   ADDRESS              JOIN-TIME
           SERIAL
FS1D243Z17000032  v7.2.5 (453)      Authorized/Up   2   169.254.1.4     Tue Sep  5
10:16:26 2023    FS1D243Z17000032
S548DF4K16000730  v7.0.7 (096)      Authorized/Up   2   169.254.1.5     Tue Sep  5
10:16:51 2023    S548DF4K16000730

         Flags: C=config sync, U=upgrading, S=staged, D=delayed reboot pending, E=config
sync error, 3=L3, V=VXLAN
         Managed-Switches: 2 (UP: 2 DOWN: 0 MAX: 72)
```

3. Verify the compatibility matrix:

```
FGT_A (global) # diagnose test application forticldd 16
Last update: 3 secs ago

FS1D24: 7.4.0 b767 07004000FIMG0900304000 (FGT Version 7.4.1 b0)
```

4. Wait for the FortiGate to perform the federated update.
5. After the federated update is complete, verify that the managed FortiSwitches were upgraded to the latest version:

```
FGT_A (vdom1) # execute switch-controller  get-conn-status
Managed-devices in current vdom vdom1:

FortiLink interface : flink
SWITCH-ID         VERSION           STATUS         FLAG   ADDRESS              JOIN-TIME
           SERIAL
FS1D243Z17000032  v7.4.0 (767)      Authorized/Up   2   169.254.1.2     Tue Sep  5
11:22:44 2023    FS1D243Z17000032
S548DF4K16000730  v7.4.0 (767)      Authorized/Up   2   169.254.1.5     Tue Sep  5
11:23:37 2023    S548DF4K16000730

         Flags: C=config sync, U=upgrading, S=staged, D=delayed reboot pending, E=config
sync error, 3=L3, V=VXLAN
         Managed-Switches: 2 (UP: 2 DOWN: 0 MAX: 72)
```

Certificates

This section includes information about certificate system related new features:

- Support Enrollment over Secure Transport for automatic certificate management 7.4.1 on page 665

Support Enrollment over Secure Transport for automatic certificate management - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:
- Enrollment over Secure Transport for automatic certificate management

The FortiGate supports Enrollment over Secure Transport (EST) and the RFC 7030 standards when generating a new CSR request, performing automatic renewals, or manually regenerating a certificate. EST provides more security for automatic certificate management than Simple Certificate Enrollment Protocol (SCEP), which is commonly used for certificate enrollment.

Background

SCEP helps automate and simplify the process for obtaining a digital certificate from a certificate authority (CA). However, SCEP does not natively support secure connections, and instead relies on the underlying transport protocol to provide security. EST was developed, which uses TLS to establish a secure communication channel over which subsequent certificate management protocol messages like initial certificate enroll and certificate renewal messages are exchanged.

On the FortiGate, when generating a certificate signing request (CSR), you can use the SCEP method to send the request to an SCEP server, or use EST to send the request to an EST server to be signed by a CA.

To configure the enrollment protocol settings for a local certificate:

```
config vpn certificate local
    edit <name>
        set enroll-protocol est
        set est-server <string>
        set est-ca-id <string>
        set est-http-username <string>
        set est-http-password <string>
        set est-client-cert <certificate>
        set est-server-cert <certificate>
        set est-srp-username <string>
        set est-srp-password <string>
    next
end
```

`est-server <string>`	Enter the address and port for EST server (such as https://example.com:1234).
`est-ca-id <string>`	Enter the CA identifier of the CA server for signing with EST.
`est-http-username <string>`	Enter the HTTP Authentication username for signing with EST.
`est-http-password <string>`	Enter the HTTP Authentication password for signing with EST.

`est-client-cert <certificate>`	Enter the certificate used to authenticate this FortiGate to the EST server.
`est-server-cert <certificate>`	Enter the EST server's certificate that has to be verifiable by the specified certificate on the FortiGate.
`est-srp-username <string>`	Enter the EST SRP authentication username.
`est-srp-password <string>`	Enter the EST SRP authentication password.

To manually generate a CSR for the EST server to be signed by a CA:

```
# execute vpn certificate local generate est {reqired_1} {reqired_2} {reqired_3} [options]
```

option 1 (required)	Name of the local server certificate.
option 2 (required)	Cryptography algorithm: `rsa-1024`, `rsa-1536`, `rsa-2048`, `rsa-4096`, `ec-secp256r1`, `ec-secp384r1`, or `ec-secp521r1`.
option 3 (required)	URL and listening port of the remote EST responder.
option 4 (optional)	Server certificate subject in the certificate enroll request. Separate fields by a comma (,).
option 5 (optional)	Subject Alternative Name (SAN). This can be an FQDN and/or IP. Use DNS:<FQDN>,IP:<IP_address> for example. If the issuing CA does not support SAN, this option will be ignored. Separate fields by a comma (,).
option 6 (optional)	HTTP authentication username.
option 7 (optional)	HTTP authentication password.
option 8 (optional)	CA identifier.
option 9 (optional)	CA certificate used to verify the remote EST responder server certificate and certificates issued by a remote PKI.
option 10 (optional)	Password for the private key.
option 11 (optional)	Client certificate.
option 12 (optional)	Source IP for communications to the CA server.
option 13 (optional)	TLS-SRP username.
option 14 (optional)	TLS-SRP password.

Example 1: enrolling for a new FortiGate server certificate with EST

To enroll for a new FortiGate server certificate with EST:

1. Verify that the FortiGate can communicate with remote EST responder (testrfc7030.com):

```
# execute ping testrfc7030.com
PING testrfc7030.com (54.70.32.33): 56 data bytes
64 bytes from 54.70.32.33: icmp_seq=0 ttl=31 time=13.6 ms
64 bytes from 54.70.32.33: icmp_seq=1 ttl=31 time=19.1 ms
```

```
64 bytes from 54.70.32.33: icmp_seq=2 ttl=31 time=16.5 ms
^C
--- testrfc7030.com ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 13.6/16.4/19.1 ms
```

2. Start running debugs to track the progress of the enrollment:

```
# diagnose debug application est -1
# diagnose debug enable
```

3. Create a new server CSR file locally and send it to the remote EST responder:

```
# execute vpn certificate local generate est est-test101 ec-secp256r1
https://testrfc7030.com:8443 CN=firewall-portal1,DC=local,DC=COM DNS:firewall-
portal1.local.ca,IP:172.18.60.184 estuser estpwd G_CA_Cert_1
```

The CA certificate (G_CA_Cert_1) is used to verify the remote EST responder server certificate and certificates issued by a remote PKI.

testrfc7030.com is a self-signed CA, which by default is not in the local trusted root store and must be imported prior to enrollment.

If the CA that issues the server certificate is not in the local root store, an error would appear in the debug messages:

```
# diagnose debug application est -1
# diagnose debug enable
...
[1795] est_curl_req: Error buf: SSL certificate problem: self-signed
certificate in certificate chain,
[2402] est_simple_enroll: Failed to get ca certs: -1.
...
```

4. If the enrollment was successful, in a few seconds, a `Done` message appears. Verify the debugs to view the enrollment process.

   a. The remote CA's certificate is retrieved and stored locally in the EST configuration after being verified with the CA in the trusted root store:

```
[1962] __est_curl_set_auth: trace
[2046] __est_curl_set_auth: HTTP Authentication username is set
[2050] __est_curl_set_auth: HTTP Authentication password is set
[2075] __est_get_ca_certs: ===========STARTED===========
[1728] est_curl_req: URL: https://testrfc7030.com:8443/.well-known/est/cacerts
[1776] est_curl_req: HTTP GET
[143] __curl_ssl_ctx_finalizer: global CAs are loaded.
[165] __curl_ssl_ctx_finalizer: SSL_CTX ex data is set.
[1651] curl_header_debug_func: Header received:HTTP/1.1 200 OK
```

   b. The debug displays the CA used by the remote EST responder:

```
[1191] save_pkcs7_certs: Saving pkcs7 response
[505] est_print_pkcs7: Certs: (1 in total)
[507] est_print_pkcs7: Cert 1:
[427] est_print_x509:         Version: 3 (0x2)
        Serial Number:
            ab:e8:32:e1:f6:6a:6b:43
```

```
Issuer: CN=estExampleCA
Subject: CN=estExampleCA
X509v3 extensions:
    X509v3 Basic Constraints:
        CA:TRUE
    X509v3 Subject Key Identifier:
        1A:DF:39:84:C2:56:E6:6C:CF:2A:B4:26:A5:FD:0C:D2:43:F5:3D:3E

[1220] save_pkcs7_certs: Received 1 certs
[1228] save_pkcs7_certs: Saving cert(s):
        is_global:1
        est_url:https://testrfc7030.com:8443
        source_ip:NULL
        ca_identifier:NULL
```

c. The CA certificate is imported. FortiOS sends a query to learn about the attributes supported by the CA in the certificate request and will then create the CSR accordingly:

```
[1288] save_pkcs7_certs: CA certs imported!
[2101] __est_get_csr_attrs: ============STARTED============
[1728] est_curl_req: URL: https://testrfc7030.com:8443/.well-known/est/csrattrs
[1776] est_curl_req: HTTP GET
[1651] curl_header_debug_func: Header received:HTTP/1.1 200 OK
[1651] curl_header_debug_func: Header received:Status: 200 OK
[1651] curl_header_debug_func: Header received:Content-Type: application/csrattrs
[1651] curl_header_debug_func: Header received:Content-Transfer-Encoding: base64
[1651] curl_header_debug_func: Header received:Content-Length: 57
[1651] curl_header_debug_func: Header received:
[1787] est_curl_req: Response 200
[1788] est_curl_req: Buffer:MCYGBysGAQEBARYGCSqGSIb3DQEJAQYFK4EEACIGCWCGSAFlAwQCAg==
[1439] decode_csrattrs_callback: Decoding csrattrs, resp->len: 57
[1474] decode_csrattrs_callback: Object: 1.3.6.1.1.1.1.22 undefined
[1474] decode_csrattrs_callback: Object: 1.2.840.113549.1.9.1 emailAddress
[1474] decode_csrattrs_callback: Object: 1.3.132.0.34 secp384r1
[1474] decode_csrattrs_callback: Object: 2.16.840.1.101.3.4.2.2 sha384
```

d. The CSR information is generated, which is sent to the remote EST responder:

```
est_ctx: is_global:1
vfid:0
svr_original_url:https://testrfc7030.com:8443
svr_hostinfo:Exists
ca_identifier:(null)
http_username:estuser
http_password:estpwd
clt_cert:(null)
svr_cert:(null)
srp_username:(null)
srp_password:(null)
source_ip:(null)
need_pop:0
newcert_name:est-test101
passwd:(null)
rsa_keysize:0
ec_curvename:secp256r1
subject:CN=firewall-portal1,DC=local,DC=COM
sub_alt_name:DNS:firewall-portal1.local.ca,IP:172.18.60.184
svr_cert_x509:NULL
```

```
        csr_attrs:Exists
        csr:NULL
        pkey:NULL
        header_ptr:NULL
        tmp_p10:NULL
[2259] __est_simple_enroll: ============STARTED============
```

e. The CSR is sent to the EST responder:

```
[1728] est_curl_req: URL: https://testrfc7030.com:8443/.well-known/est/simpleenroll
[1753] est_curl_req: HTTP POST
[1651] curl_header_debug_func: Header received:HTTP/1.1 200 OK
[1651] curl_header_debug_func: Header received:Status: 200 OK
[1651] curl_header_debug_func: Header received:Content-Type: application/pkcs7-mime;
smime-type=certs-only
[1651] curl_header_debug_func: Header received:Content-Transfer-Encoding: base64
[1651] curl_header_debug_func: Header received:Content-Length: 585
[1651] curl_header_debug_func: Header received:
```

f. The CA issues the certificate and sends it back in a PKCS #7 structure:

```
[1787] est_curl_req: Response 200
[1788] est_curl_req:
Buffer:MIIBqwYJKoZIhvcNAQcCoIIBnDCCAZgCAQExADALBgkqhkiG9w0BBwGgggGAMIIB
fDCCASOgAwIBAgIDB0aXMAoGCCqGSM49BAMCMBcxFTATBgNVBAMTDGVzdEV4YW1w
...
```

g. The FortiGate decodes and displays the attributes of the certificate, then saves the certificate:

```
[1191] save_pkcs7_certs: Saving pkcs7 response
[505] est_print_pkcs7: Certs: (1 in total)
[507] est_print_pkcs7: Cert 1:
[427] est_print_x509:         Version: 3 (0x2)
        Serial Number: 476823 (0x74697)
        Issuer: CN=estExampleCA
        Subject: CN=firewall-portal1
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature
            X509v3 Subject Key Identifier:
                9B:F8:39:D5:21:E6:FF:49:FF:AC:02:57:5B:FC:4C:1A:8B:1E:5D:8F
            X509v3 Authority Key Identifier:
                1A:DF:39:84:C2:56:E6:6C:CF:2A:B4:26:A5:FD:0C:D2:43:F5:3D:3E

[1220] save_pkcs7_certs: Received 1 certs
[1228] save_pkcs7_certs: Saving cert(s):
        is_global:1
        est_url:https://testrfc7030.com:8443
        source_ip:NULL
        ca_identifier:NULL

[1246] save_pkcs7_certs: Received 1 cert(s)
[427] est_print_x509:         Version: 3 (0x2)
        Serial Number: 476823 (0x74697)
        Issuer: CN=estExampleCA
        Subject: CN=firewall-portal1
```

```
            X509v3 extensions:
                X509v3 Basic Constraints:
                    CA:FALSE
                X509v3 Key Usage:
                    Digital Signature
                X509v3 Subject Key Identifier:
                    9B:F8:39:D5:21:E6:FF:49:FF:AC:02:57:5B:FC:4C:1A:8B:1E:5D:8F
                X509v3 Authority Key Identifier:
                    1A:DF:39:84:C2:56:E6:6C:CF:2A:B4:26:A5:FD:0C:D2:43:F5:3D:3E

    [827] est_cmdb_update_cert: Cert est-test101 updated in CMDB
    [1276] save_pkcs7_certs: The cert is saved!
    [592] est_ctx_clear_tmp_data: trace
    [2408] est_simple_enroll: POST ret:0
    [592] est_ctx_clear_tmp_data: trace
    Done.
```

Example 2: automatically renewing a FortiGate server certificate with EST

When the time for certificate renewal is up, the FortiGate will use the existing EST parameters to perform an automatic renewal. This example demonstrates the renewal process through debugs.

To automatically renew a FortiGate server certificate with EST:

1. Verify the current local certificate configuration:

```
config vpn certificate local
(local) # get est-test101
name                    : est-test101
password                : *
comments                :
private-key             : *
certificate             :
        Subject:     CN = firewall-portal1
        Issuer:      CN = estExampleCA
        Valid from:  2023-04-06 22:37:34  GMT
        Valid to:    2024-04-05 22:37:34  GMT
        Fingerprint: AE:67:11:CF:7D:F9:57:A4:09:8B:55:0A:F1:B1:7A:CF
...
state                   : OK
range                   : global
source                  : user
source-ip               : 0.0.0.0
ike-localid-type        : asn1dn
enroll-protocol         : est
est-server              : https://testrfc7030.com:8443
est-ca-id               :
est-http-username       : estuser
est-http-password       : estpwd
est-client-cert         :
est-server-cert         :
est-srp-username        :
est-srp-password        :
auto-regenerate-days: 0
auto-regenerate-days-warning: 0
```

Note that the current `Valid to` date and time is `2024-04-05 22:37:34 GMT`, which is one year from the issue date.

2. Start running debugs to track the progress of the renewal:

```
# diagnose debug application est -1
# diagnose debug enable
```

3. For demonstration purposes, update the `auto-regenerate-days` setting to 364 days to trigger the automatic renewal on the FortiGate:

```
config vpn certificate local
    edit est-test101
        set auto-regenerate-days 364
    next
end
```

4. Verify the debugs to confirm that the certificate was renewed.
   a. The FortiGate uses the content of the current certificate to create a new CSR. User credentials used for the initial enrollment are stored in local certificate configuration, but they are not used for renewal:

```
[1024] reconstruct_est_ctx: Reconstruction succeeded
est_ctx:        is_global:1
        vfid:0
        svr_original_url:https://testrfc7030.com:8443
        svr_hostinfo:NULL
        ca_identifier:
        http_username:estuser
        http_password:estpwd
        clt_cert:
        svr_cert:
        srp_username:
        srp_password:
        source_ip:(null)
        need_pop:0
        newcert_name:est-test101
        passwd:f51da8548af5fef820edfe6267b0c178e76f7c3eae40ee0900318fc77ab6bd
        rsa_keysize:0
        ec_curvename:(null)
        subject:(null)
        sub_alt_name:(null)
        svr_cert_x509:NULL
        csr_attrs:NULL
        csr:NULL
        pkey:NULL
        header_ptr:NULL
        tmp_p10:NULL
```

   b. The FortiGate sends the current server certificate for authentication/authorization and not the username/password used for initial enrollment:

```
[2453] est_simple_reenroll: Try to use est-test101 as client cert to authenticate
[1962] __est_curl_set_auth: trace
[2011] __est_curl_set_auth: Warning: cert est-test101 may not have the correct key
usage for TLS client authentication
[2014] __est_curl_set_auth: Will use cert est-test101 to prove my identity
...
[1651] curl_header_debug_func: Header received:
```

```
[1787] est_curl_req: Response 200
[1788] est_curl_req: Buffer:MCYGBysGAQEBARYGCSqGSIb3DQEJAQYFK4EEACIGCWCGSAFlAwQCAg==
[1439] decode_csrattrs_callback: Decoding csrattrs, resp->len: 57
[1474] decode_csrattrs_callback: Object: 1.3.6.1.1.1.1.22 undefined
[1474] decode_csrattrs_callback: Object: 1.2.840.113549.1.9.1 emailAddress
[1474] decode_csrattrs_callback: Object: 1.3.132.0.34 secp384r1
[1474] decode_csrattrs_callback: Object: 2.16.840.1.101.3.4.2.2 sha384
est_ctx:        is_global:1
        vfid:0
        svr_original_url:https://testrfc7030.com:8443
        svr_hostinfo:Exists
        ca_identifier:
        http_username:estuser
        http_password:estpwd
        clt_cert:est-test101
        svr_cert:
        srp_username:
        srp_password:
        source_ip:(null)
        need_pop:0
        newcert_name:est-test101
        passwd:f51da8548af5fef820edfe6267b0c178e76f7c3eae40ee0900318fc77ab6bd
        rsa_keysize:0
        ec_curvename:(null)
        subject:(null)
        sub_alt_name:(null)
        svr_cert_x509:NULL
        csr_attrs:Exists
        csr:NULL
        pkey:NULL
        header_ptr:NULL
        tmp_p10:NULL
[2274] __est_simple_reenroll: ============STARTED============
```

c. The CSR for renewal is successfully generated:

```
[965] est_generate_csr_from_cert: Successfully generated CSR for est-test101
[2200] __est_simple_post: Data to be posted:
|||MIIBQDCB5gIBAjAbMRkwFwYDVQQDDBBmaXJld2FsbC1wb3J0YWwxMFkwEwYHKoZI
zj0CAQYIKoZIzj0DAQcDQgAEQoJQmPedxPNUcfCyRvpqyt1oiiJX/me+TdButUSu
8hg+9nPF6+xNf+5LmtG/YKHeXyCKG6xB9OmJf255Zmx+5qBpMGcGCSqGSIb3DQEJ
DjFaMFgwCQYDVR0TBAIwADALBgNVHQ8EBAMCB4AwHQYDVR0OBBYEFJv4OdUh5v9J
/6wCV1v8TBqLHl2PMB8GA1UdIwQYMBaAFBrfOYTCVuZszyq0JqX9DNJD9T0+MAoG
CCqGSM49BAMCA0kAMEYCIQCK3Li51F7fXsyKZwtIcYMFvDobY3cKKTTDixtN7QZ2
jwIhAKUkqfWPAzwcxQaNQw6pyYvo18ymB9aEheeIXZfGI+tV
|||

[1728] est_curl_req: URL: https://testrfc7030.com:8443/.well-known/est/simplereenroll
[1753] est_curl_req: HTTP POST
[1651] curl_header_debug_func: Header received:HTTP/1.1 200 OK
[1651] curl_header_debug_func: Header received:Status: 200 OK
[1651] curl_header_debug_func: Header received:Content-Type: application/pkcs7-mime;
smime-type=certs-only
[1651] curl_header_debug_func: Header received:Content-Transfer-Encoding: base64
[1651] curl_header_debug_func: Header received:Content-Length: 590
[1651] curl_header_debug_func: Header received:
[1787] est_curl_req: Response 200
```

d. The new certificate is received in PKCS #7 and is saved:

```
[1788] est_curl_req:
Buffer:MIIBrQYJKoZIhvcNAQcCoIIBnjCCAZoCAQExADALBgkqhkiG9w0BBwGgggGCMIIB
fjCCASOgAwIBAgIDB0aYMAoGCCqGSM49BAMCMBcxFTATBgNVBAMTDGVzdEV4YW1w
...
[1191] save_pkcs7_certs: Saving pkcs7 response
[505] est_print_pkcs7: Certs: (1 in total)
...
[1220] save_pkcs7_certs: Received 1 certs
[1228] save_pkcs7_certs: Saving cert(s):
        is_global:1
        est_url:https://testrfc7030.com:8443
        source_ip:NULL
        ca_identifier:

[1246] save_pkcs7_certs: Received 1 cert(s)
[427] est_print_x509:         Version: 3 (0x2)
        Serial Number: 476824 (0x74698)
        Issuer: CN=estExampleCA
        Subject: CN=firewall-portal1
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature
            X509v3 Subject Key Identifier:
                9B:F8:39:D5:21:E6:FF:49:FF:AC:02:57:5B:FC:4C:1A:8B:1E:5D:8F
            X509v3 Authority Key Identifier:
                1A:DF:39:84:C2:56:E6:6C:CF:2A:B4:26:A5:FD:0C:D2:43:F5:3D:3E

[827] est_cmdb_update_cert: Cert est-test101 updated in CMDB
[1276] save_pkcs7_certs: The cert is saved!
[592] est_ctx_clear_tmp_data: trace
[2477] est_simple_reenroll: POST ret:0
[592] est_ctx_clear_tmp_data: trace
```

5. Verify the renewed local certificate configuration:

```
config vpn certificate local
(local) # get est-test101
name                : est-test101
password            : *
comments            :
private-key         : *
certificate         :
        Subject:     CN = firewall-portal1
        Issuer:      CN = estExampleCA
        Valid from:  2023-04-06 22:55:09  GMT
        Valid to:    2024-04-05 22:55:09  GMT
        Fingerprint: D9:51:6C:EF:04:E9:79:8D:A0:EE:10:23:4A:F4:46:B7
        Root CA:     No
        Version:     3
        Serial Num:
                07:46:a5
        Extensions:
                Name:     X509v3 Basic Constraints
```

```
                Critical: no
                Content:
```

Note that the `Valid to` date and time is now `2024-04-05 22:55:09 GMT`.

Example 3: manually regenerating a local certificate with EST

Note that manually regenerating the certificate will not generate a new server key pair.

To manually regenerate a local certificate with EST:

1. Run the following command:

```
# execute vpn certificate local generate est est-test101
Certificate 'est-test101' already exists, re-generate will ignore all the options you
have provided.
Are you sure to re-generate the certificate?
Do you want to continue? (y/n) y
```

2. Verify the debugs to confirm that the certificate was generated:

```
# diagnose debug application est -1
# diagnose debug enable
...
[1024] reconstruct_est_ctx: Reconstruction succeeded
est_ctx:        is_global:1
        vfid:0
        svr_original_url:https://testrfc7030.com:8443
        svr_hostinfo:NULL
        ca_identifier:
        http_username:estuser
        http_password:estpwd
        clt_cert:
        svr_cert:
        srp_username:
        srp_password:
        source_ip:(null)
        need_pop:0
        newcert_name:est-test101
        passwd:f51da8548af5fef820edfe6267b0c178e76f7c3eae40ee0900318fc77ab6bd
        rsa_keysize:0
        ec_curvename:(null)
        subject:(null)
        sub_alt_name:(null)
        svr_cert_x509:NULL
        csr_attrs:NULL
        csr:NULL
        pkey:NULL
        header_ptr:NULL
        tmp_p10:NULL
[2453] est_simple_reenroll: Try to use est-test101 as client cert to authenticate
[1962] __est_curl_set_auth: trace
...
```

3. Once the certificate is saved, verify the local certificate configuration:

```
config vpn certificate local
(local) # get est-test101
name                : est-test101
password            : *
comments            :
private-key         : *
certificate         :
        Subject:     CN = firewall-portal1
        Issuer:      CN = estExampleCA
        Valid from:  2023-04-13 17:23:40  GMT
        Valid to:    2024-04-12 17:23:40  GMT
        Fingerprint: 4A:96:E1:73:6D:D3:64:FE:A3:A8:28:56:1D:39:05:37
        Root CA:     No
        Version:     3
        Serial Num:
                07:47:02
        Extensions:
                Name:     X509v3 Basic Constraints
                Critical: no
                Content:
                CA:FALSE

                Name:     X509v3 Key Usage
                Critical: no
                Content:
                Digital Signature

                Name:     X509v3 Subject Key Identifier
                Critical: no
                Content:
                9B:F8:39:D5:21:E6:FF:49:FF:AC:02:57:5B:FC:4C:1A:8B:1E:5D:8F

                Name:     X509v3 Authority Key Identifier
                Critical: no
                Content:
                1A:DF:39:84:C2:56:E6:6C:CF:2A:B4:26:A5:FD:0C:D2:43:F5:3D:3E


state               : OK
range               : global
source              : user
source-ip           : 0.0.0.0
ike-localid-type    : asn1dn
enroll-protocol     : est
est-server          : https://testrfc7030.com:8443
est-ca-id           :
est-http-username   : estuser
est-http-password   : estpwd
est-client-cert     :
est-server-cert     :
est-srp-username    :
est-srp-password    :
auto-regenerate-days: 0
auto-regenerate-days-warning: 0
```

The Subject Key Identifier is the same, so no new key pair was generated.

Security

This section includes information about security system related new features:

Enhance BIOS-level signature and file integrity checking

This information is also available in the FortiOS 7.4 Administration Guide:

- BIOS-level signature and file integrity checking

The BIOS-level signature and integrity checking has been enhanced by enforcing each FortiOS GA firmware image, AV engine file, and IPS engine file to be dually-signed by the Fortinet CA and a third-party CA. The BIOS verifies that each file matches their secure hash as indicated by their certificates. Users are warned when there is a failed integrity check, and the system may be prevented from booting depending on the severity and the BIOS security level.

Signature checking occurs when the FortiOS firmware, AV, and IPS engine files are uploaded. This allows the FortiGate to warn users of potential risks involved with uploading an unauthenticated file.

The outcome of the signature and integrity check depends on the security level configured in BIOS and the certificate authority that signed the file.

The following table summarizes the use cases and the potential outcome based on the security level.

Use case	Certificate signed by		Outcome based on security level		
	Fortinet CA	Third-party CA	Level 2	Level 1	Level 0
GA-Certified (GA firmware, Beta firmware, Top3 final builds)	Yes	Yes	Accept	Accept	Accept
Non-GA certified (Special builds: Top3 and NPI quick builds)	Yes	No	Warning	Accept	Accept
Interim and Dev builds, or unknown build	No	Yes or No	Reject	Warning	Accept

The security levels on the BIOS are:

- Level 2: in order to operate normally, FortiOS requires all file signatures to match their secure checksums as indicated on both Fortinet and third-party CA signed certificates.
  - If a file has a Fortinet CA signed certificate but no third-party signed certificates, then FortiOS can still run but displays a warning in the GUI and CLI.

  - If a file has no valid certificate signed by the Fortinet CA, then FortiOS is not allowed to run.
- Level 1: in order to operate normally, FortiOS only requires all file signatures to match their secure checksums as indicated on the Fortinet CA signed certificate.
  - If a file has no valid certificate signed by the Fortinet CA, then FortiOS can still run but displays a warning in the GUI and CLI.
- Level 0 (not recommended): FortiOS does not perform code verification.

On FortiGates without supported BIOS security levels, the device acts like security level 1. For example, on a FortiGate-VM that does not have BIOS, the security level is defaulted to level 1.

To verify the BIOS security level:

```
# get system status
Version: FortiGate-101F v7.4.0,build2352,230427 (GA.F)
Security Level: 2
Firmware Signature: certified
```

The following examples outline the different use cases when upgrading firmware and AV files on a FortiGate model that supports BIOS security levels, and a FortiGate model that does not support BIOS security levels.

For more information, see the Firmware and Registration section and Manual updates in the FortiOS Administration Guide.

Upgrading on a device with BIOS security levels

The following use cases are applicable when upgrading firmware and AV files on a FortiGate with BIOS security levels. Firmware is upgraded using the *System > Firmware & Registration* page, and AV files are upgraded using the *System > FortiGuard* page. Fictitious build numbers are used to demonstrate the functionality of this feature.

Level 2

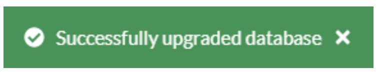
When upgrading from 7.2.4 to 7.4.0 with a dually-signed firmware image, FortiOS verifies the certificates and accepts the image. The following CLI output shows the messages displayed when a FortiGate is upgraded.

```
FortiGate_101F (global) # get system status
Version: FortiGate-101F v7.2.4,build1396,230131 (GA.F)
Firmware Signature: certified
Virus-DB: 1.00000(2018-04-09 18:07)
…
FortiGate_101F (global) # Image verification OK!
Firmware upgrade in progress ...

Done.

The system is going down NOW !!

Please stand by while rebooting the system.
Restarting system.
…

System is starting...

The config file may contain errors.
```

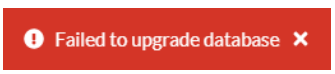
```
Please see details by the command 'diagnose debug config-error-log read'.

FortiGate_101F login: admin
Password:
Welcome!

FortiGate_101F (global) # get system status
Version: FortiGate-101F v7.4.0,build2352,230427 (GA.F)
Security Level: 2
Firmware Signature: certified
```

When upgrading from 7.2.4 to 7.4.0 with an unsigned firmware image in the GUI, FortiOS is unable to verify the certificates and rejects the image. A notification is displayed that *This firmware image didn't pass the signature verification*.

When running 7.4.0 and uploading a dually-signed AV engine file on the *System* > *FortiGuard* page, FortiOS verifies the certificates and accepts the file. A notification is displayed (*Successfully upgraded database*).



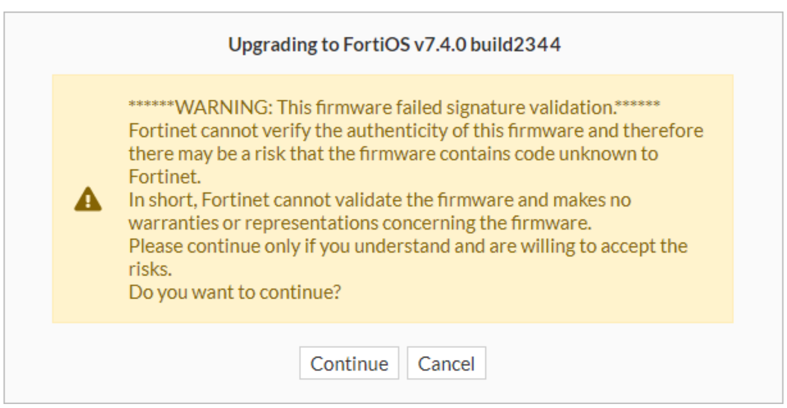
When running 7.4.0 and uploading an unsigned AV engine file on the *System* > *FortiGuard* page, FortiOS is unable to verify the certificates and rejects the file. A notification is displayed that the device *Failed to upgrade database*.



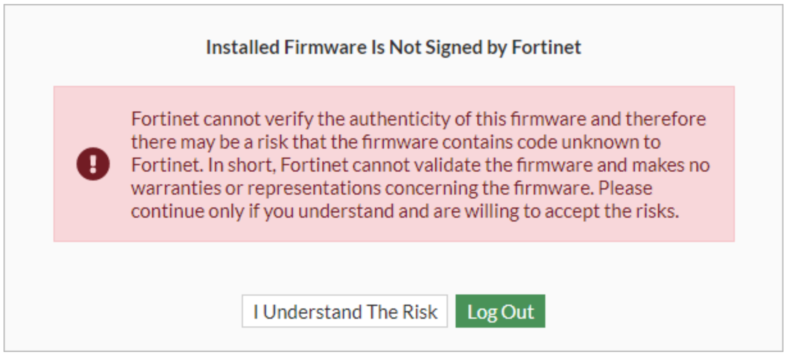
Level 1

When upgrading from 7.2.4 to 7.4.0 with a dually-signed firmware image, FortiOS verifies the certificates and accepts the image. No warning is displayed during the upgrade, or while the system is running in 7.4.0.

When upgrading from 7.2.4 to 7.4.0 with an unsigned firmware image in the GUI, FortiOS is unable to verify the certificates and the image fails verification. The upgrade will still occur. However, during the upgrade process, a warning dialog is displayed indicating that *This firmware failed signature validation*. The user can click *Continue* to upgrade the firmware.

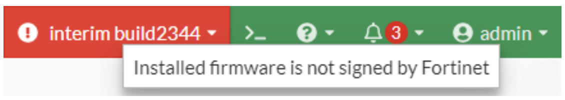


When the user logs in to the FortiGate running 7.4.0, a warning dialog is displayed indicating that the *Installed Firmware is Not Signed by Fortinet*. The user can click *I Understand The Risk* to log in.

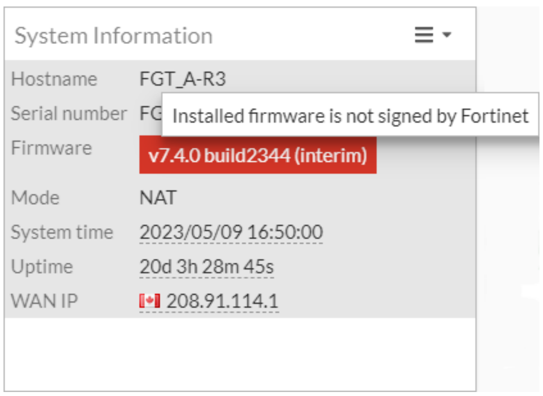


When the FortiGate is running unsigned firmware, warnings appear in the GUI and CLI.

- Top banner: the unsigned firmware version is highlighted in red. Hovering over the unsigned firmware version displays a tooltip that the *Installed firmware is not signed by Fortinet*.



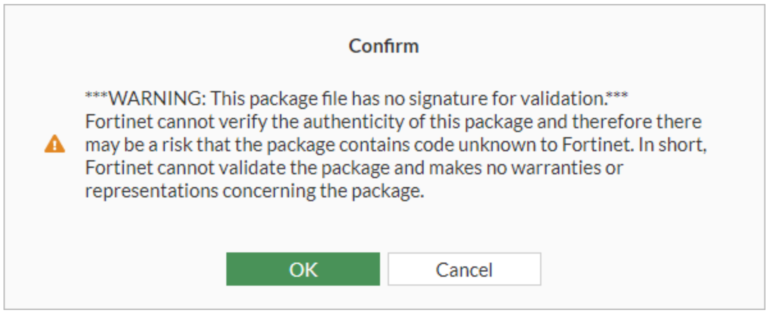
- *Dashboard* > *Status* > *System Information* widget: the unsigned firmware version is highlighted in red. Hovering over the unsigned firmware version displays a tooltip that the *Installed firmware is not signed by Fortinet*.



- Enter the following in the CLI to verify the firmware status:

```
# get system status
Version: FortiGate-VM64 v7.4.0,build2344,230418 (interim)
Security Level: 1
Firmware Signature: un-certified
Virus-DB: 91.03113(2023-05-09 15:26)
```

When running 7.4.0 and uploading an unsigned AV engine file on the *System* > *FortiGuard* page, FortiOS is unable to verify the certificates and the file fails verification. A warning dialog is displayed indicating that *This package file has no signature for validation*, but the user can click *OK* to use the file.



Level 0

When upgrading from 7.2.4 to 7.4.0 with a dually-signed firmware image, FortiOS verifies the certificates and accepts the image. No verification is performed.

When upgrading from 7.2.4 to 7.4.0 with an unsigned firmware image in the GUI, FortiOS does not verify the certificates. No warnings are displayed that the firmware is unverified.

When running 7.4.0 and uploading an unsigned AV engine file on the *System > FortiGuard* page, FortiOS does not verify the certificates. No warnings are displayed that the file is unverified.

Upgrading on a device without BIOS security levels

The following use cases are applicable when upgrading firmware and AV files on a FortiGate without BIOS security levels. Firmware is upgraded using the *System > Firmware & Registration* page, and AV files are upgraded using the *System > FortiGuard* page. A FortiGate 60E is used in these examples and acts like it has security level 1.

When upgrading from 7.2.4 to 7.4.0 with a dually-signed firmware image, FortiOS verifies the certificates and accepts the image.

When upgrading from 7.2.4 to 7.4.0 with an unsigned firmware image in the GUI, FortiOS is unable to verify the certificates and the image fails verification. A warning dialog is displayed indicating that *This firmware failed signature validation*, but the user can click *Continue* to use the firmware.

When running 7.4.0 and uploading an unsigned AV engine file on the *System > FortiGuard* page, FortiOS is unable to verify the certificates and the file fails verification. A warning dialog is displayed indicating that *This package file has no signature for validation*, but the user can click *OK* to use the file.

Real-time file system integrity checking

This information is also available in the FortiOS 7.4 Administration Guide:

- Real-time file system integrity checking

Real-time file system integrity checking has two main purposes:

- Prevent unauthorized modification of important binaries.
- Detect unauthorized binaries and prevent them from running.

How it works

When the FortiGate boots, the system performs a BIOS level integrity check on important internal files, the AV engine file, and the IPS engine file. These files are signed by the process described in Enhance BIOS-level signature and file integrity checking on page 676, and the BIOS verifies their signature against their certificates.

Once these files are verified to be authentic, the BIOS can boot the root filesystem and other executables and libraries. Once loaded, real-time protection begins. The important executables and binaries are protected from write access and any modifications. It also blocks the kernel from loading any modules. Any unauthorized loading of modules is blocked. If violations are found, logs are triggered.

A hash of all executable binaries and libraries is taken and stored in memory. If there is a hash mismatch when attempting to run a binary, that binary is blocked from running, and the system is rebooted. A log will be generated with ID 20234.

If there is a missing hash when attempting to run a binary, then the system is rebooted. A log will be generated with ID 20223.

The system also runs a periodic check to verify the integrity of important binaries and AV and IPS engines.

Log summary

The following logs are recorded when specific actions take place.

Log	Description
20230 - LOG_ID_SYS_SECURITY_WRITE_VIOLATION 432	The root filesystem is read only. Any modification triggers this log.
20231 - LOG_ID_SYS_SECURITY_HARDLINK_VIOLATION 432	An attacker trying to replace symlink triggers this log.
20232 - LOG_ID_SYS_SECURITY_LOAD_MODULE_VIOLATION 433	Only the kernel can load modules. Any unusual loading of modules triggers this log.
20233 - LOG_ID_SYS_SECURITY_FILE_HASH_MISSING 434	File hashes are generated for legitimate files during bootup. If a hash cannot be found, the file may be suspicious as it could be a new routine inserted by an attacker. The binary is blocked.
20234 - LOG_ID_SYS_SECURITY_FILE_HASH_MISMATCH 434	File hashes are generated for legitimate files during bootup. If a hash does not match when the file is exercised, it is an indication that it could have been modified by an attacker. The system is rebooted.

Detection examples

Example 1: system reboots due to mismatched hash

```
fos_ima: fos_process_appraise 110: Executable File(/bin/node) doesn't match previous hash,
it has been changed
Restarting system.
```

```
…
fos_ima: fos_process_appraise 110: Executable File(/lib/libc.so.6) doesn't match previous
hash, it has been changed
Restarting system.
…
```

Logs similar to the following are captured:

```
date="2023-06-16" time="12:01:44" id=7245222014288399309 bid=471609558 dvid=6533
itime=1686909705 euid=3 epid=3 dsteuid=3 dstepid=3 logver=604132092 logid="0100020234"
type="event" subtype="system" level="alert" msg="Hash of executable file(/bin/init) doesn't
match the previous." logdesc="Integrity check of Run/loading Excutable File failed without
Integrity measure" severity="alert" eventtime=1686909705825483706 tz="+0200"
devid="xxxxxxxxx" vd="root" devname="xxxxxxxxx"

date="2023-06-15" time="09:57:54" id=7244819017507013700 bid=470303007 dvid=1431
itime=1686815875 euid=3 epid=3 dsteuid=3 dstepid=3 logver=604132092 logid="0100020234"
type="event" subtype="system" level="alert" msg="Hash of executable file(/lib/libc.so.6)
doesn't match the previous." logdesc="Integrity check of Run/loading Excutable File failed
without Integrity measure" severity="alert" eventtime=1686815874936267770 tz="+0200" devid="
xxxxxxxxx " vd="root" devname=" xxxxxxxxx”
```

Example 2: suspected compromise due to an observed indicator of compromise (IoC)

```
fos_ima: fos_process_appraise 99: Suspicous Executable File(/data2/libcrashpad.so) is
missing hash
…
fos_ima: fos_process_appraise 99: Suspicous Executable File(/data2/flatkc_info) is missing
hash
…
```

No logs are found.

Corrective action

In the previous examples where a mismatched or missing hash occurs, alert technical support straight away so that they may gather information to start a forensic analysis with our internal PSIRT team. There are two possible outcomes:

1. The firewall is reporting a false positive, in which a bug causes a mismatched or missing hash.

   Once verified by technical support, the corrective action may to be upgrade to a newer build where the bug is fixed
2. An actual compromise has occurred, or is occurring.

   The system could be blocking an offending binary that causes the system to malfunction, or the system could reboot to protect itself from compromise.

In either case, contact technical support for further forensic analysis. If an IoC is detected and it is determined that the persistent threat resides on the FortiGate, a reflash and reload of the firmware may be recommended.

Add built-in entropy source - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Built-in entropy source

FortiOS includes a built-in entropy source, which eliminates the need for a physical USB entropy token when booting up in FIPS mode on any platform. This enhancement continues to meet the requirements of FIPS 140-3 Certification by changing the source of entropy to CPU jitter entropy.

The `entropy-token` parameter under `config system fips-cc` is removed if the FortiGate is a SoC3, SoC4, or CP9 device.

To verify that jitter entropy is used:

1. Enable FIPS-CC mode, which will cause the FortiGate to reboot:

```
config system fips-cc
    set status enable
end

Please enter admin administrator password:********
Please re-enter admin administrator password:********

Warning: most configuration will be lost,
do you want to continue?(y/n) y
The system is going down NOW !!

Please stand by while rebooting the system.
Restarting system.
...
Reading boot image 2919154 bytes.
Initializing firewall...
System is starting...

FIPS-CC mode: Starting self-tests.
Running Configuration/VPN Bypass test...      passed
Running AES test...                           passed
Running SHA1-HMAC test...                     passed
Running SHA256-HMAC test...                   passed
Running SHA384/512-HMAC test...               passed
Running RSA test...                           passed
Running ECDSA test...                         passed
Running TLS1.1-KDF test...                    passed
Running TLS1.2-KDF test...                    passed
Running SSH-KDF test...                       passed
Running IKEv1-KDF test...                     passed
Running IKEv2-KDF test...                     passed
Running Primitive-Z test...                   passed
Running Firmware integrity test...            passed
Running RBG-instantiate test...               passed
Running RBG-reseed test...                    passed
Running RBG-generate test...                  passed
Self-tests passed
```

2. Verify the entropy token user event logs:

```
# execute log filter category event
# execute log filter field logid 0102038012
# execute log display
```

```
3 logs found.
3 logs returned.

1: date=2023-07-18 time=20:27:56 eventtime=1689737275853093806 tz="-0700"
logid="0102038012" type="event" subtype="user" level="notice" vd="root" logdesc="Seeding
from entropy source" user="system" action="reseeding" msg="Reseeding PRNG from JitterEnt
entropy"

2: date=2023-07-18 time=20:26:56 eventtime=1689737146847643497 tz="-0700"
logid="0102038012" type="event" subtype="user" level="notice" vd="root" logdesc="Seeding
from entropy source" user="system" action="seeding" msg="Seeding PRNG from JitterEnt
entropy"

3: date=2023-07-18 time=19:29:25 eventtime=1689733702417108422 tz="-0700"
logid="0102038012" type="event" subtype="user" level="notice" vd="root" logdesc="Seeding
from entropy source" user="system" action="seeding" msg="Seeding PRNG from JitterEnt
entropy"
```

Unauthorized firmware modification attempt reporting - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Real time file system integrity checking

This enhancement improves upon the Real-time file system integrity checking feature by implementing an automatic reporting mechanism in the event of a firmware modification attempt. In the rare event that unauthorized modification is detected in the firmware, the system will immediately log and report the modification attempt to FortiGuard through a secure channel. Payloads are encrypted to ensure the security of the transferred information. Information about the attempted modification of firmware helps Fortinet Inc. proactively investigate the incident and protect future malicious attempts at compromising the system.

After reporting the modification attempt, the FortiGate real-time file system integrity checking feature continues with the required actions based on the assessed threat. This may involve reverting the change and rebooting the firewall to mitigate the threat.

Example

This example demonstrates when an attempt to alter files in the 'bin' directory was made by a threat actor.

Captured log:

```
1: date=2024-02-16 time=18:29:15 eventtime=1708136955710925685 tz="-0800" logid="0100020230"
type="event" subtype="system" level="alert" vd="vd1" logdesc="Write Permission Violation"
msg="[Write Violation: try to write readonly file](/bin/lspci)."
```

The FortiGate sends an encrypted report to FortiGuard with information about the affected platform and the Modification Attempt such as:

- FortiGate serial number
- Model number
- FortiOS firmware

- Type of modification attempt (such as *Write violation*)
- File path (such as */bin/lspci*)
- File size
- Time of access and modification

Security Fabric

This section includes information about Security Fabric related new features:

Fabric settings and connectors

This section includes information about Security Fabric settings and Fabric connector related new features:

MAC address threat feed

This information is also available in the FortiOS 7.4 Administration Guide:

- MAC address threat feed

A MAC address threat feed is a dynamic list that contains MAC addresses, MAC ranges, and MAC OUIs. The list is periodically updated from an external server and stored in text file format on an external server. After the FortiGate imports this list, it can be used as a source in firewall policies, proxy policies, and ZTNA rules. For policies in transparent mode or virtual wire pair policies, the MAC address threat feed can be used as a source or destination address.

Text file example:

```
01:01:01:01:01:01
01:01:01:01:01:01-01:01:02:50:20:ff
8c:aa:b5
```

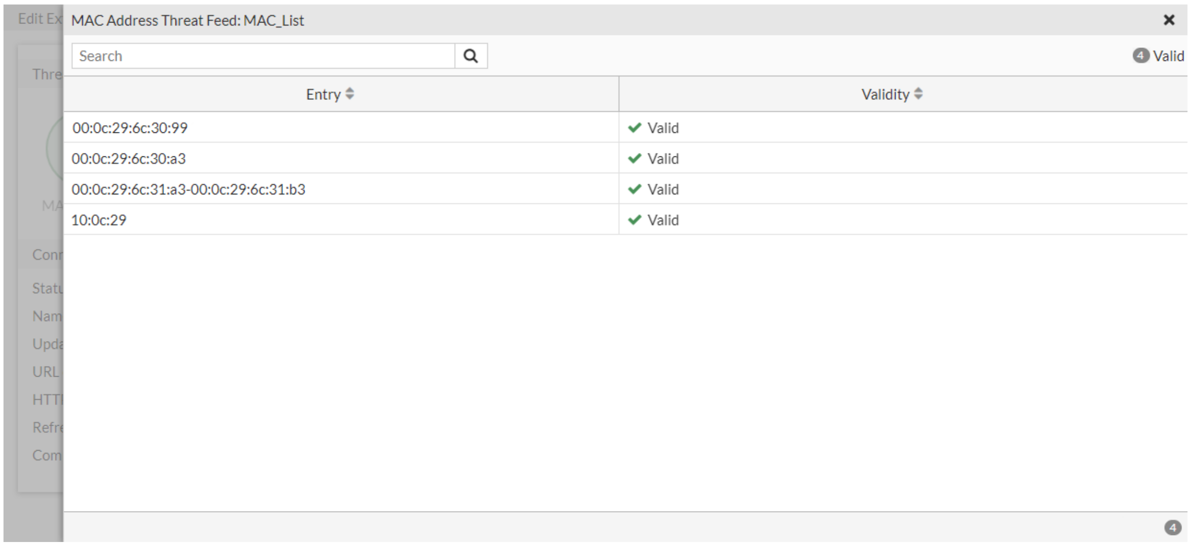
The file contains one MAC address, MAC range, or MAC OUI per line.

Example configuration

In this example, a list of MAC addresses is imported using the MAC address threat feed. The newly created threat feed is then used as a source in a firewall policy with the action set to accept. Any traffic from the client MAC addresses that match the defined firewall policy will be allowed.

To configure a MAC address threat feed in the GUI:

1. Go to *Security Fabric* > *External Connectors* and click *Create New*.
2. In the *Threat Feeds* section, click *MAC Address*.
3. Set the *Name* to *MAC_List*.
4. Set the *Update method* to *External Feed*.
5. Set the *URL of external resource* to *http://172.16.200.55/external-resources/Ext-Resource-Type-as-Address-mac-1.txt*.
6. Configure the remaining settings as required, then click *OK*.
7. Edit the connector, then click *View Entries* to view the MAC addresses in the feed.



To configure a MAC address threat feed in the CLI:

```
config system external-resource
    edit "MAC_List"
        set type mac-address
        set resource "http://172.16.200.55/external-resources/Ext-Resource-Type-as-Address-
mac-1.txt"
        set server-identity-check {none | basic | full}
    next
end
```

To improve the security of the connection, it is recommended to enable server certificate validation (`server-identity-check`) either in basic or full mode. By default, it is set to none.

To apply a MAC address threat feed in a firewall policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and create a new policy, or edit an existing one.
2. Configure the policy fields as required.
3. In the *Source* field, click the + and select *MAC_List* from the list (in the *MAC ADDRESS FEED* section).
4. Set *Action* to *ACCEPT*.
5. Click *OK*.

To apply a MAC address threat feed in a firewall policy in the CLI:

```
config firewall policy
    edit 1
        set name "MAC-traffic"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "MAC_List"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set profile-protocol-options "protocol"
        set nat enable
    next
end
```

To verify the MAC addresses used in the firewall policy:

```
# diagnose sys external-mac-resource list MAC_List
MAC ranges of uuid-idx 574 (num=1)
be:d1:6b:0d:20:61-be:d1:6b:0d:20:61
```

Configuring FortiClient EMS and FortiClient EMS Cloud on a per-VDOM basis

This information is also available in the FortiOS 7.4 Administration Guide:

- Configuring FortiClient EMS and FortiClient EMS Cloud on a per-VDOM basis

FortiClient EMS and FortiClient EMS Cloud can be added on a per-VDOM basis. Enabling override is necessary to add an EMS server for each VDOM.

```
config endpoint-control settings
    set override {enable | disable}
end
```

If override is enabled for a VDOM, the global configuration will not affect the VDOM. Override must be configured for each VDOM that connects to an EMS server.

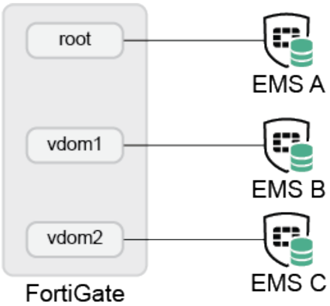
This feature requires FortiClient EMS 7.2.1 and later.

With this override configuration, the FortiGate can connect to multiple on-premise FortiClient EMS instances per VDOM. However, with this same configuration, only one FortiClient EMS Cloud instance can be connected per FortiGate.

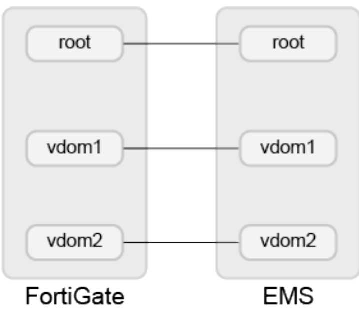
Each VDOM supports up to seven EMS servers, plus an additional seven in the global configuration. With override enabled on all ten VDOMs, a 10-VDOM contract would have up to 77 EMS servers. If override is enabled on only one VDOM, a 10-VDOM contract would have up to 14 EMS servers.

This functionality can be applied to MSSP (managed security service provider) configurations, and each VDOM has its own *FortiClient EMS* card for the EMS server or instance. For example:

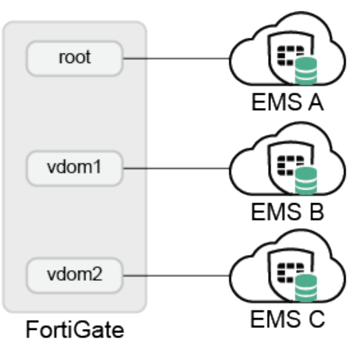
- Separate on-premise FortiClient EMS instances



- Single FortiClient EMS multi-tenant instance based on FQDN type



- Separate FortiClient EMS Cloud instances



To configure a FortiClient EMS server per VDOM in the GUI:

1. Enable override in the FortiOS CLI on the required VDOMs:

```
config endpoint-control settings
    set override enable
end
```

2. Navigate to the desired VDOM, then go to *Security Fabric > Fabric Connectors* and double-click the *FortiClient EMS* card.
3. Configure the EMS server settings as needed (see Configuring FortiClient EMS in the FortiOS Administration Guide for detailed steps).

To configure a FortiClient EMS server per VDOM in the CLI:

1. Enable override on the required VDOMs:

```
config endpoint-control settings
    set override enable
end
```

2. Configure the EMS server on the desired VDOM:

```
(root) config endpoint-control fctems-override
    edit 1
        set status enable
        set name "ems140_root"
        set server "172.16.200.140"
        set serial-number "FCTEMS8821******"
        set tenant-id "00000000000000000000000000000000"
        set capabilities fabric-auth silent-approval websocket websocket-malware push-
ca-certs common-tags-api tenant-id single-vdom-connector
    next
    edit 2
        set name "ems133_root"
        set server "172.16.200.133"
    next
end
```

Update FortiVoice connector features - 7.4.1

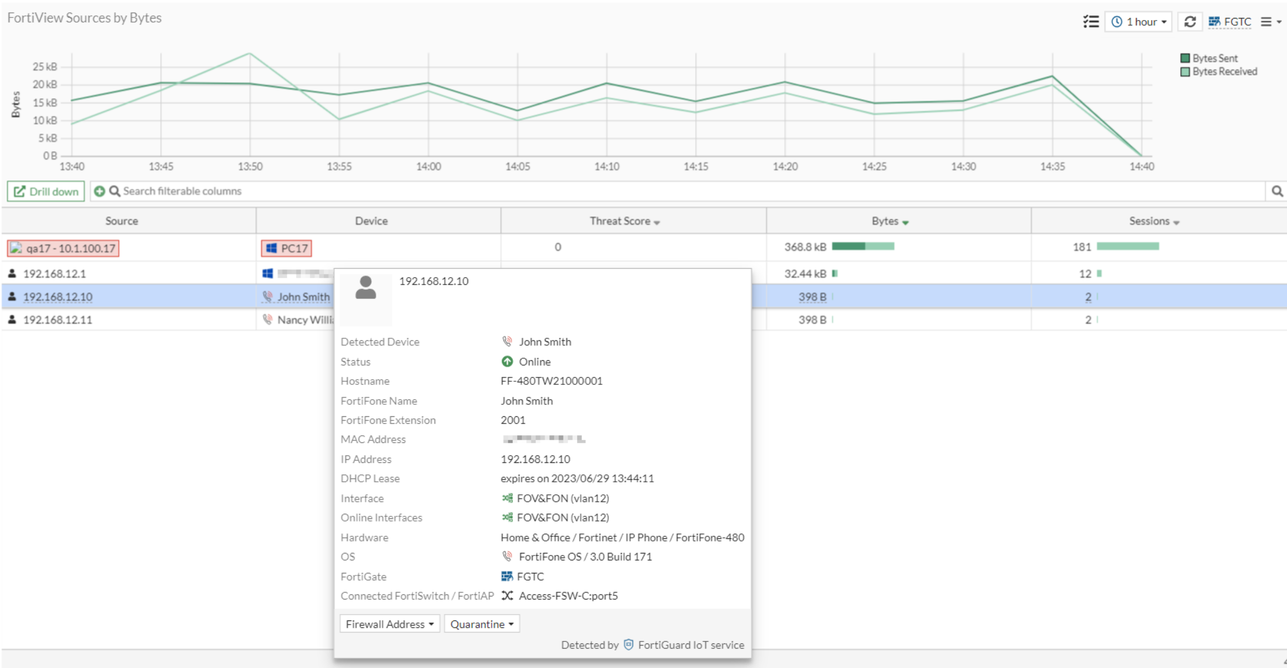
This information is also available in the FortiOS 7.4 Administration Guide:

- FortiVoice tag dynamic address

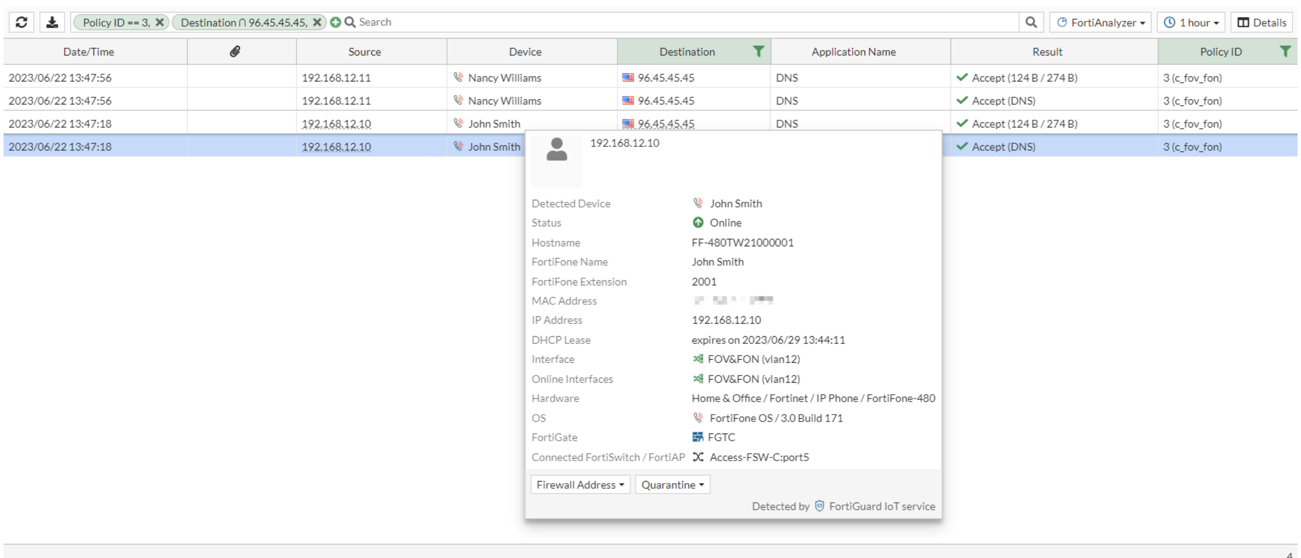
FortiVoice endpoint details are displayed in the device tooltips that can be accessed on the FortiView monitor and log pages. Users can view the display name and extension number of each FortiFone, making it easier to identify and manage endpoint phones.

Sample tooltips

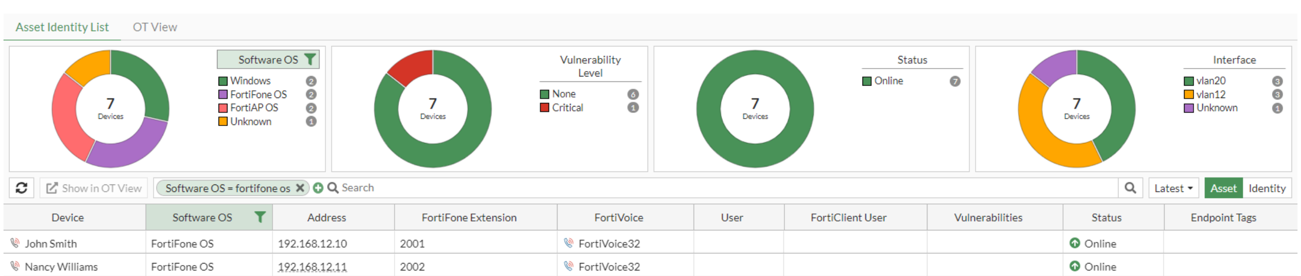
Dashboard > FortiView Sources page:



Log & Report > Forward Traffic page:



Registered FortiFones are visible on the *Security Fabric > Asset Identity Center* page.



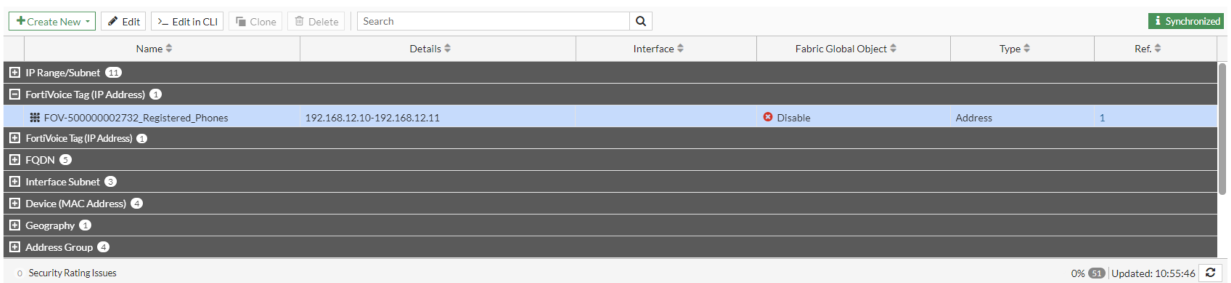
When a FortiVoice-supplied MAC or IP address is used in a firewall policy, a FortiVoice tag (MAC/IP) dynamic address is automatically created on the FortiGate that contains all the provisioned FortiFones registered with FortiVoice. The dynamic address can be used in firewall policies to restrict rules to authorized FortiFones only. This is useful for large voice deployments that require security and efficiency.

Example

In this example, two FortiFones are registered to FortiVoice and are assigned names and extension numbers. A FortiVoice Fabric connector has been authorized to join the Security Fabric. The dynamic FortiVoice tags are applied to a firewall policy.

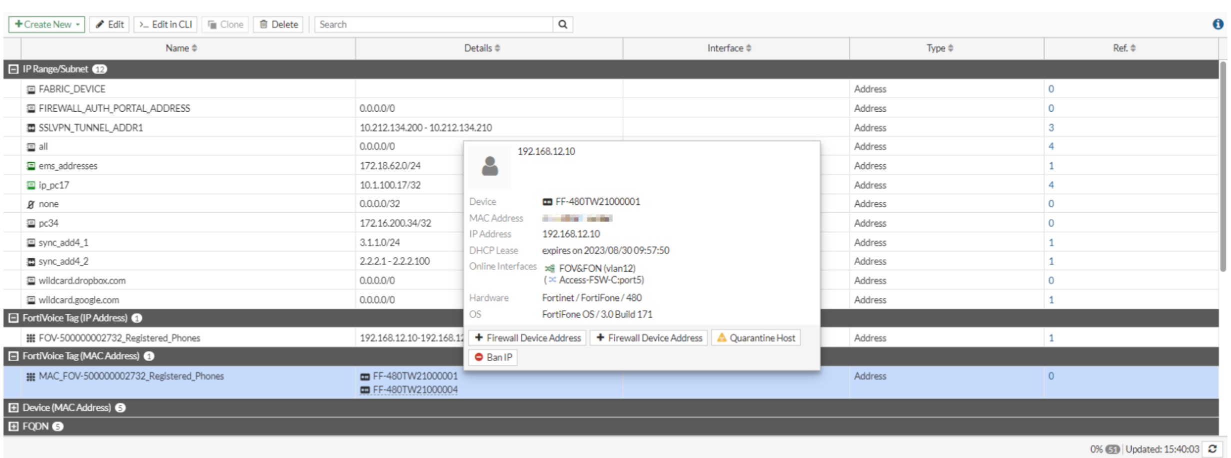
To use a FortiVoice tag dynamic firewall address in a policy:

1. Configure and authorize the FortiVoice Fabric connector (see Configuring FortiVoice for more information).
2. Go to *Policy & Objects > Addresses* to view the newly created dynamic firewall address objects:
   a. Expand the *FortiVoice Tag (IP Address)* section.



   There is one entry, *FOV-500000002732_Registered_Phones*, which matches 192.168.12.10 to 192.168.12.11.

   b. Expand the *FortiVoice Tag (MAC Address)* section. There is one entry, *MAC_FOV-500000002732_Registered_Phones*, which matches two devices. Hover over the device serial number to view the tooltip that contains the MAC address and additional information.



3. Go to *Policy & Objects > Firewall Policy* and click *Create new* or edit an existing policy.
4. In the *Source* field, click the + and add the *FOV-500000002732_Registered_Phones* and *MAC_FOV-500000002732_Registered_Phones* addresses.
5. In the *Destination* field, click the + and add the *FOV-500000002732_Registered_Phones* address.

6. Configure the other settings as needed.
7. Click *OK*.

External SDN connectors

This section includes information about external SDN connector related new features:

- Support IPv6 dynamic addresses retrieved from Cisco ACI SDN connector on page 693

Support IPv6 dynamic addresses retrieved from Cisco ACI SDN connector

IPv6 dynamic addresses can be retrieved from Cisco ACI SDN connectors. IPv6 addresses imported from Cisco ACI to the Fortinet SDN Connector VM can be imported into the FortiGate as IPv6 dynamic addresses. The Fortinet SDN Connector VM must be running version 1.1.10 or later.

```
config firewall address6
    edit <name>
        set type dynamic
        set sdn <ACI_connector>
    next
end
```

For more information about this feature, see Support IPv6 dynamic addresses retrieved from Cisco ACI SDN connector.

Security ratings

This section includes information about security rating related new features:

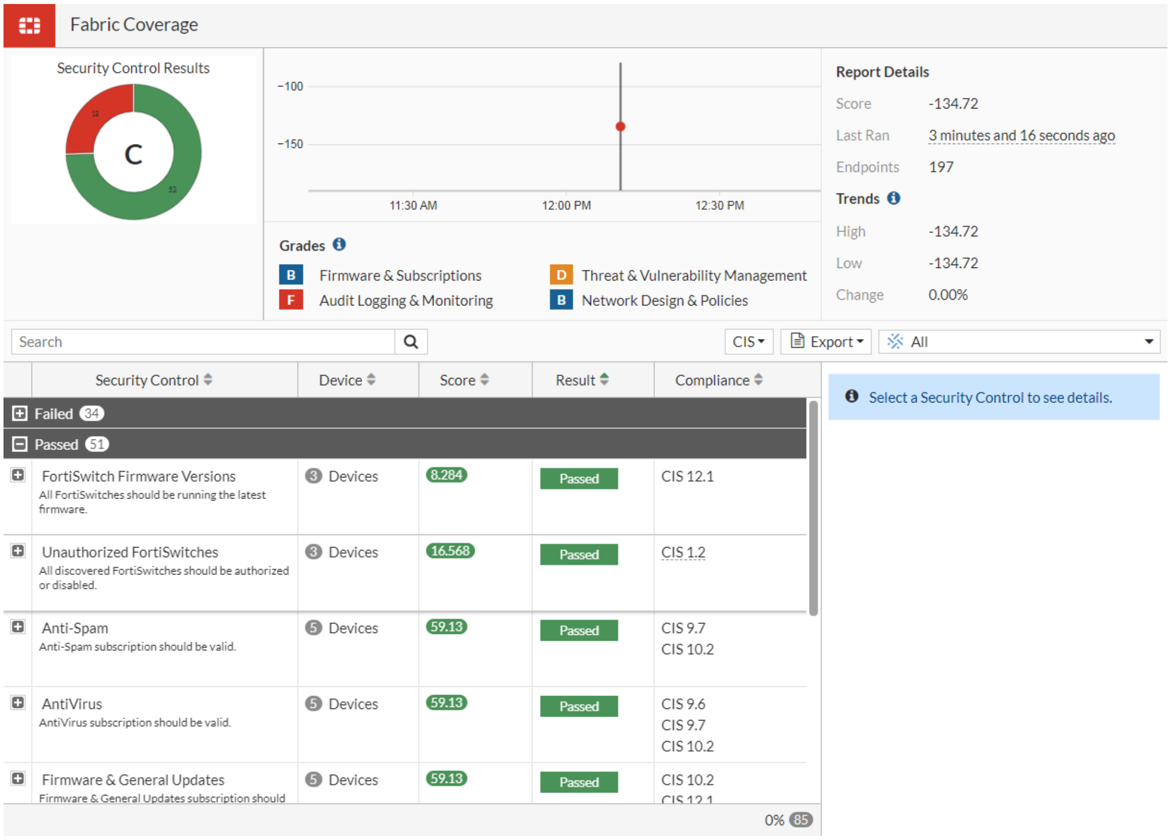
- Support CIS compliance standards within security ratings 7.4.1 on page 693
- Add prompt for one-time upgrade when a critical vulnerability is detected upon login 7.4.1 on page 695

Support CIS compliance standards within security ratings - 7.4.1

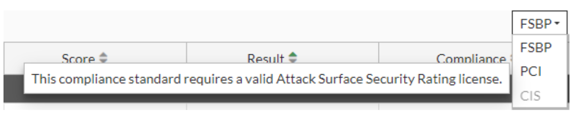
CIS security control mappings have been added to the *Security Rating* page. Users can view ratings by CIS compliance and view the description for each CIS control. The FortiGate must have a valid Attack Surface Security Rating license to view security ratings grouped by CIS.

To view CIS compliance standard security controls:

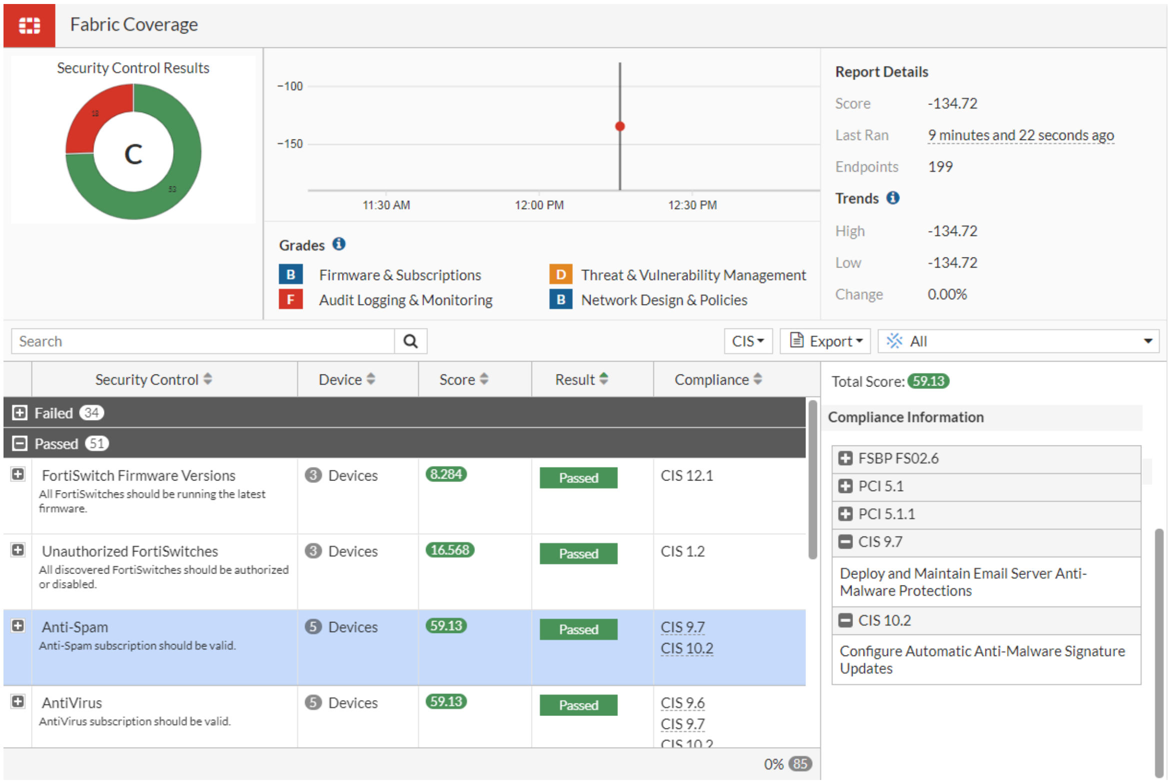
1. Go to *Security Fabric* > *Security Rating* and select a posture card.
2. Select *CIS* from the dropdown.



On FortiGates without valid Attack Surface Security Rating license, the *CIS* option in the dropdown is grayed out.



3. Select a security rule. In the *Compliance Information* section (to the right), click the + to expand and view more details about related CIS compliance for the rule.

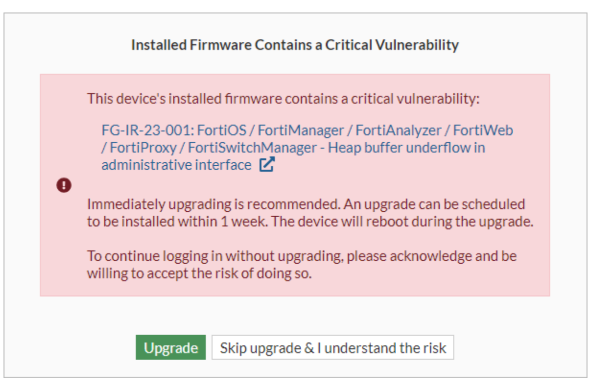


Add prompt for one-time upgrade when a critical vulnerability is detected upon login - 7.4.1

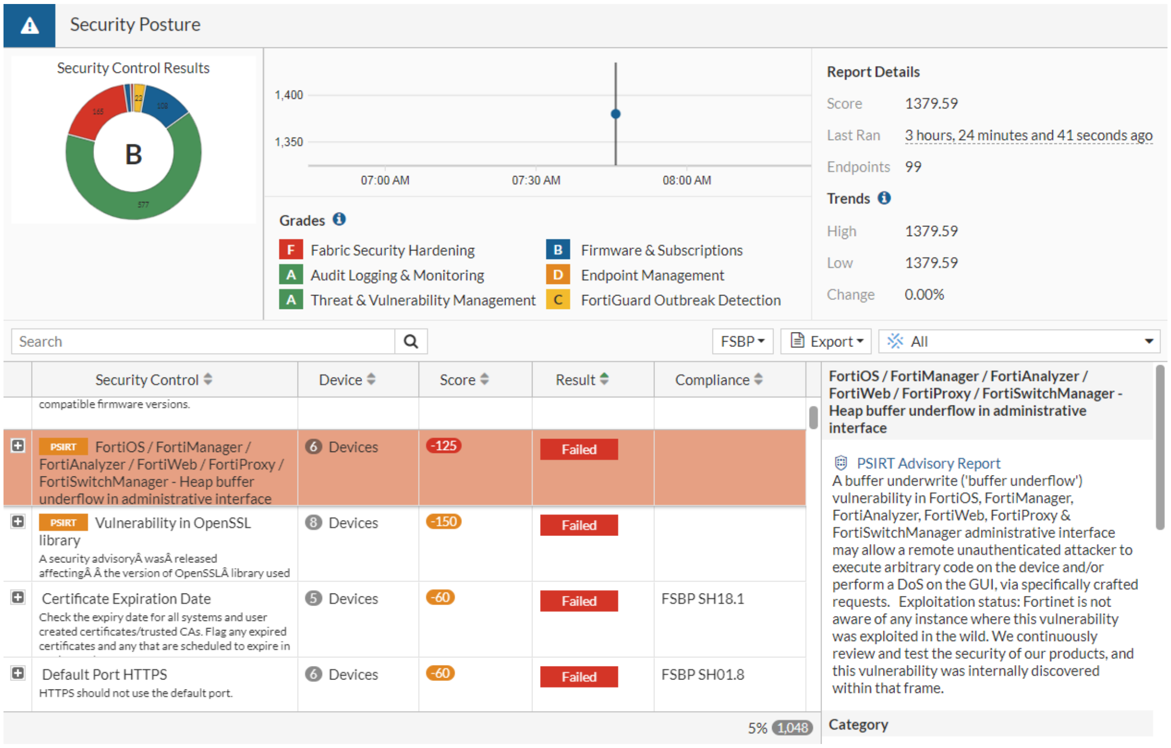
This information is also available in the FortiOS 7.4 Administration Guide:

- One-time upgrade prompt when a critical vulnerability is detected upon login

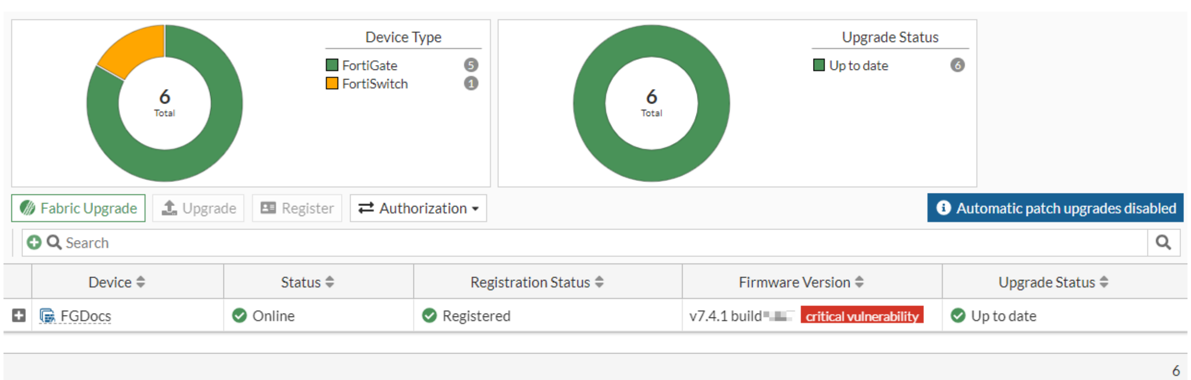
When FortiOS detects a critical vulnerability, a prompt appears for a one-time upgrade after logging into the FortiGate. A warning message is displayed in the GUI about the critical vulnerability and allows the administrator to either upgrade or skip it. This ensures that the administrator is aware of any potential security risks and can take immediate action to address them.



Clicking the hyperlinked vulnerability name opens the *Security Fabric > Security Rating* page, which displays more information about the vulnerability.



Clicking the *Upgrade* button opens the *System > Firmware & Registration* page where the administrator can upgrade the device.



Clicking the *Skip upgrade & I understand the risk* button continues the log in process as usual.

Diagnostics

To view vulnerability results after performing security rating scan:

```
# diagnose report-runner vuln-read
Index: 0
Name: FG-IR-23-001: FortiOS / FortiManager / FortiAnalyzer / FortiWeb / FortiProxy /
FortiSwitchManager - Heap buffer underflow in administrative interface
FortiGate Serial: FGVM02TM23000000
```

To clear the vulnerability result:

```
# diagnose report-runner vuln-clean
Deleted temporary critical vulnerability file
```

Automation

This section includes information about automation related new features:

- Improve automation trigger and action selection on page 697

Improve automation trigger and action selection

This information is also available in the FortiOS 7.4 Administration Guide:
- System Events page shortcut

Automation triggers and actions have been simplified to allow for better management with the following improvements:

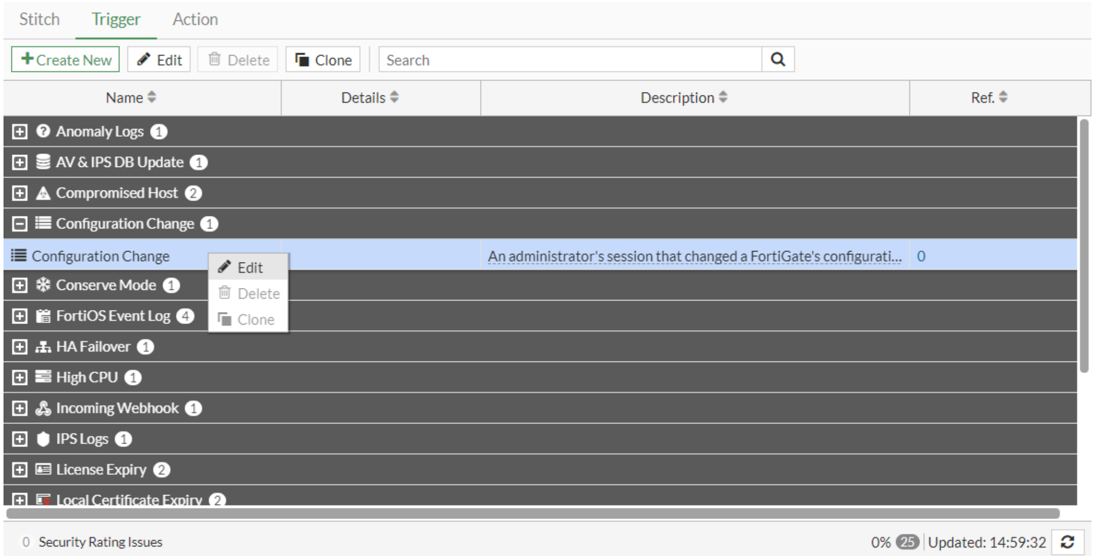
- Hide simple triggers and actions that should be reused from the creation pages.
- Add a shortcut on the *System Events* > *Logs* page to create an automation trigger based on the event log.
- Add FortiCare email option for Email actions.

When upgrading from FortiOS 7.2, all existing automation triggers, actions, and stitches are preserved.

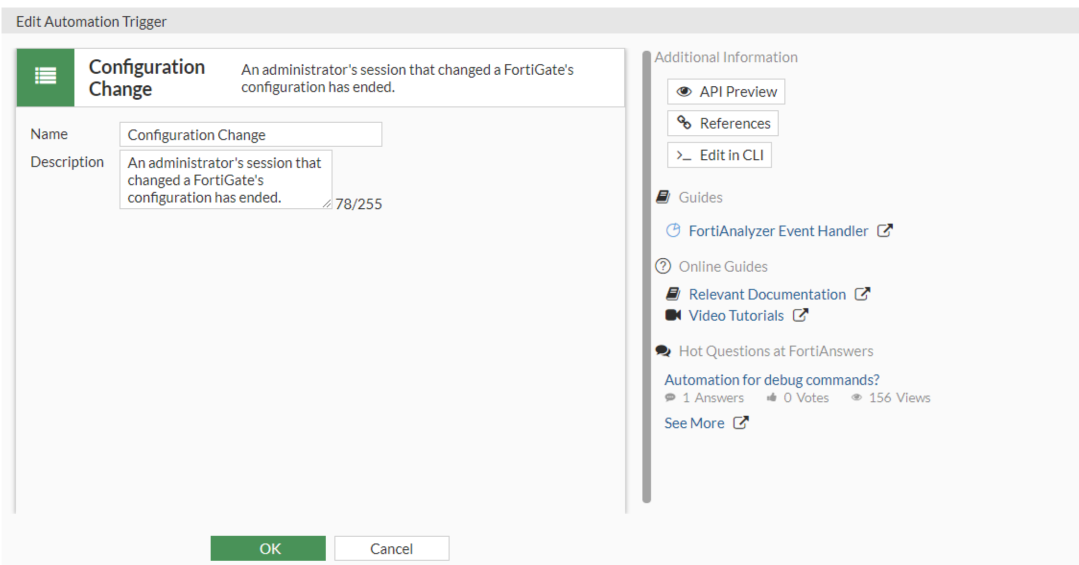
Creating triggers and actions

Static automation triggers and actions that require only a name, description, and one setting are added by default, such as the Configuration Change trigger and IP Ban action. Static triggers and actions can be edited, but they cannot be deleted.

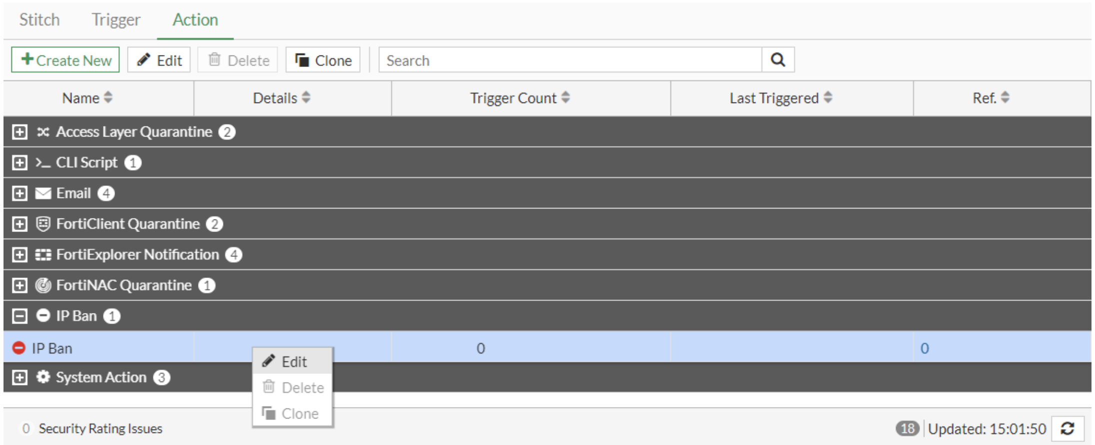
- Configuration Change trigger that appears in the *Trigger* tab:



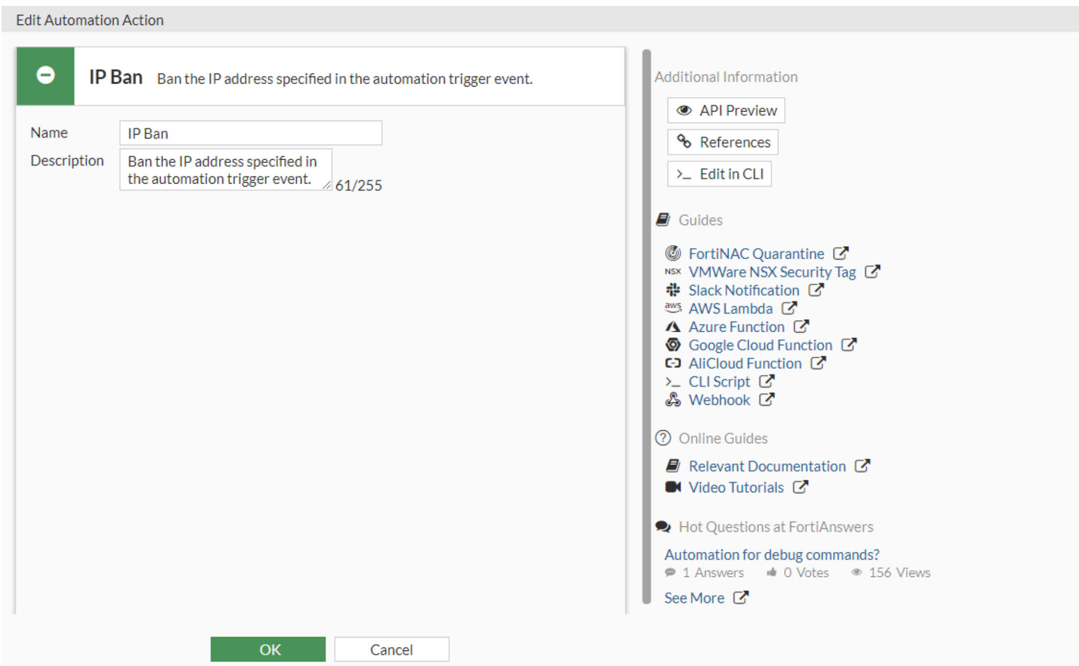
- Editing the Configuration Change trigger:



- IP Ban action that appears in the *Action* tab:

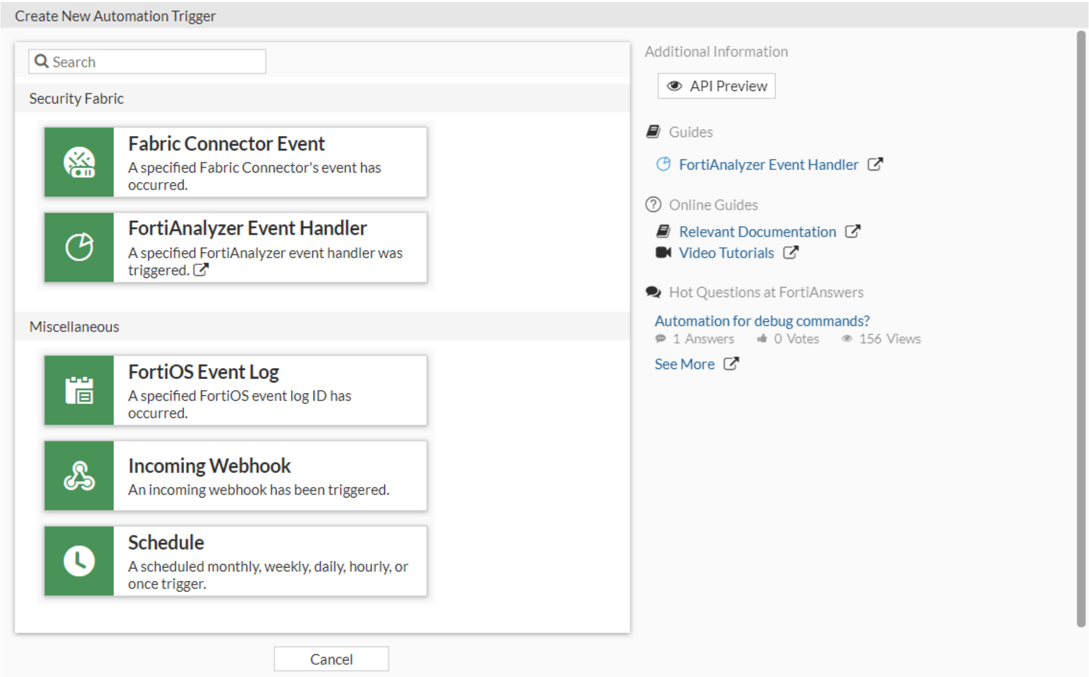


- Editing the IP Ban action:

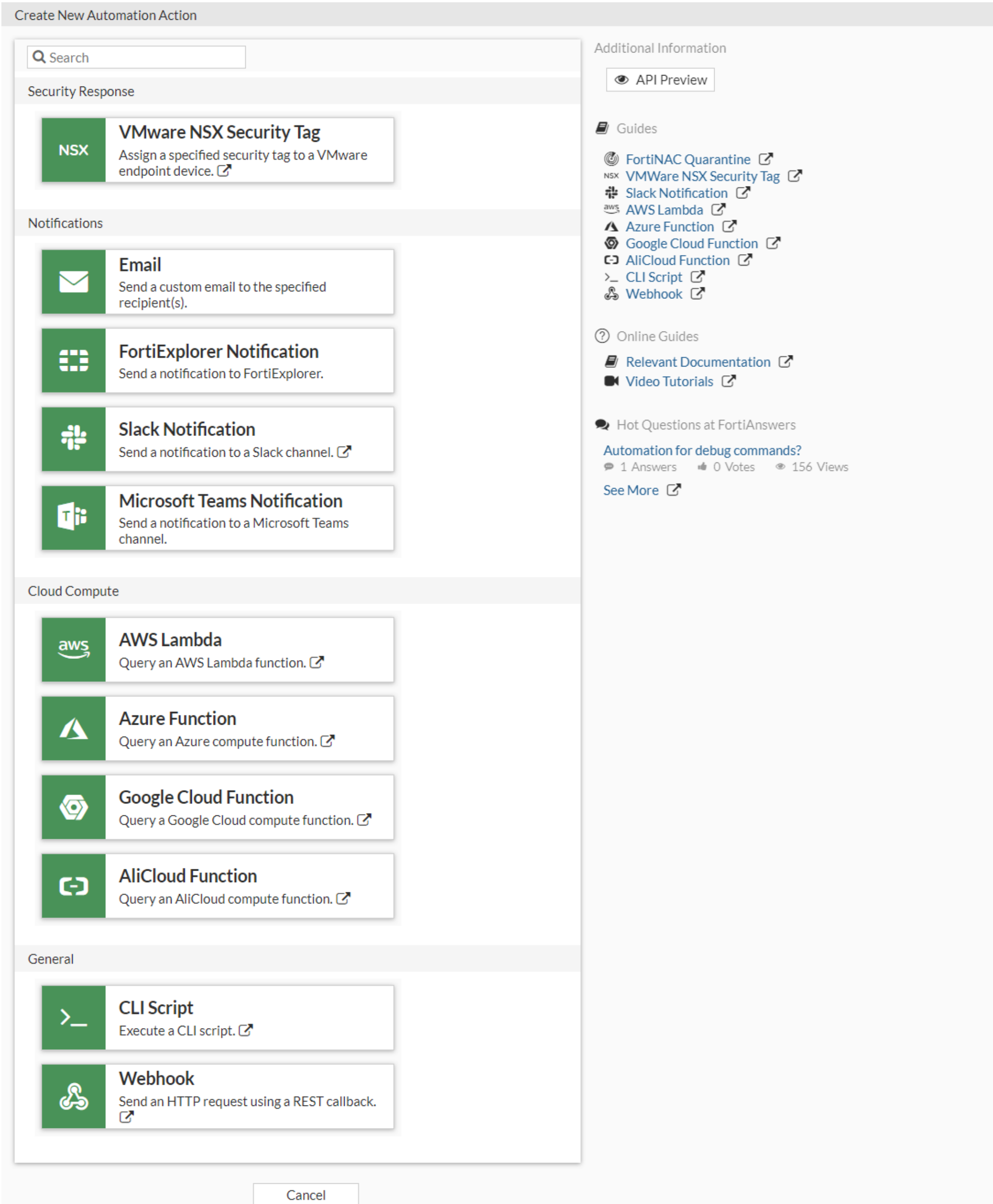


Clicking the *Create New* button on the *Trigger* and *Action* tabs (or clicking *Create* within the *Create Automation Stitch* page) only displays dynamic options where multiple settings need to be configured.

- *Create New Automation Trigger* page:



- *Create New Automation Action* page:

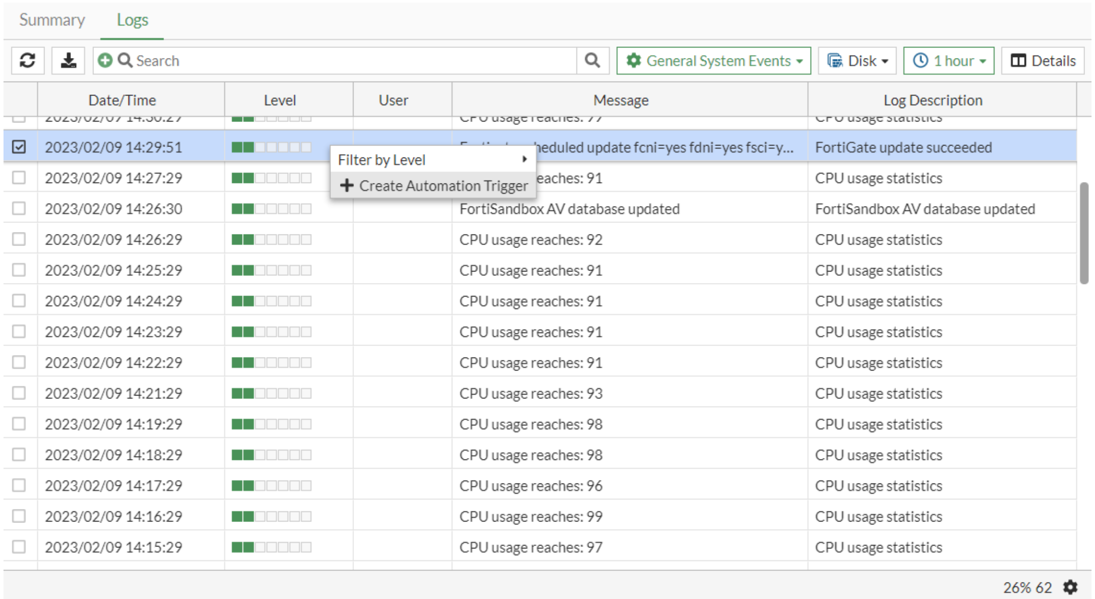


Creating a trigger from the System Events page

A FortiOS Event Log trigger can be created using the shortcut on the *System Events* > *Logs* page. In this example, a trigger is created for a FortiGate update succeeded event log.

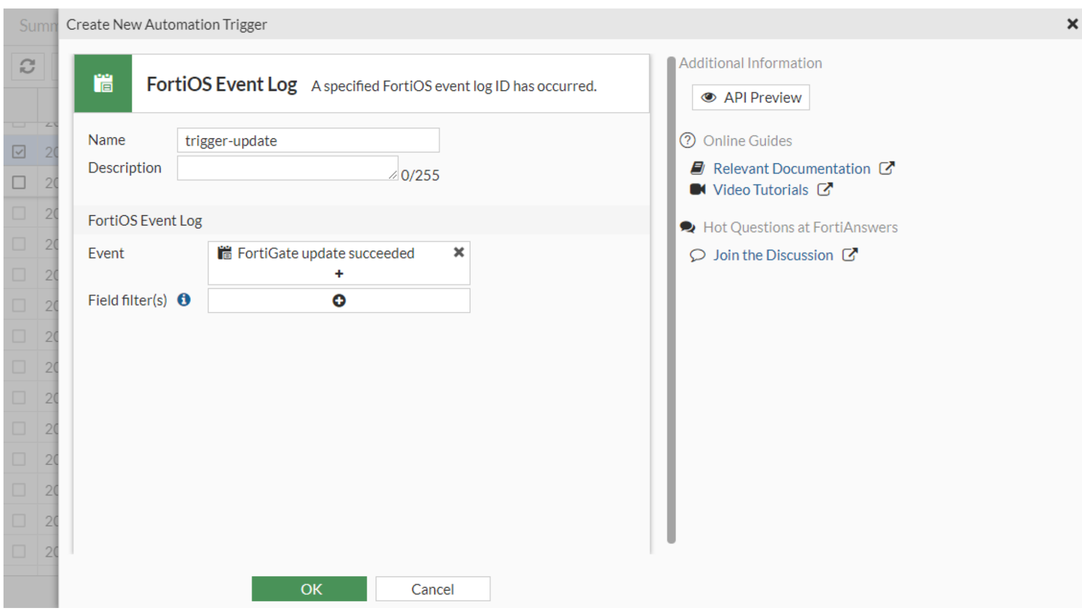
To configure a FortiOS Event Log trigger from the System Events page:

1. Go to *Log & Report* > *System Events* and select the *Logs* tab.
2. Select a log for a successful FortiGate update, then right-click and select *Create Automation Trigger*.

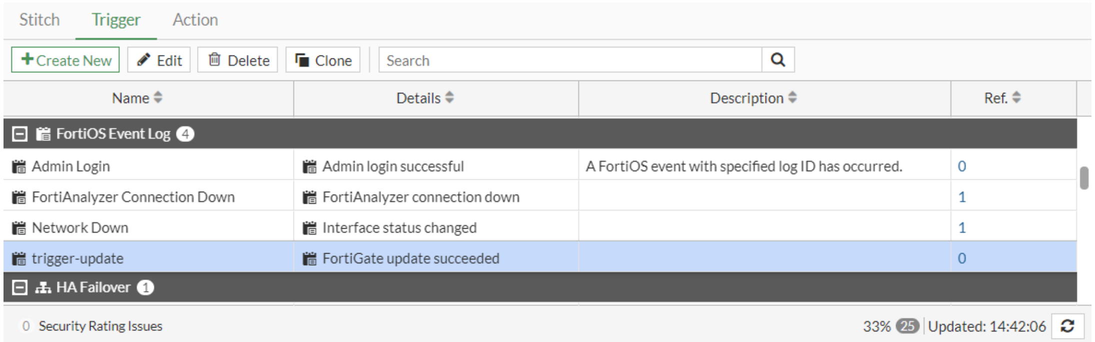


The *Create New Automation Trigger* pane opens to configure the FortiOS Event Log settings.

3. Enter a name (such as *trigger-update*). The *Event* field is already populated with *FortiGate update succeeded*.



4. Optionally in the *Field filter(s)* field, click the + to add multiple field filters. The configured filters must match in order for the stitch to be triggered.
5. Click *OK*. The trigger is now listed on the *Security Fabric > Automation > Trigger* page.



Using the FortiCare email address in Email actions

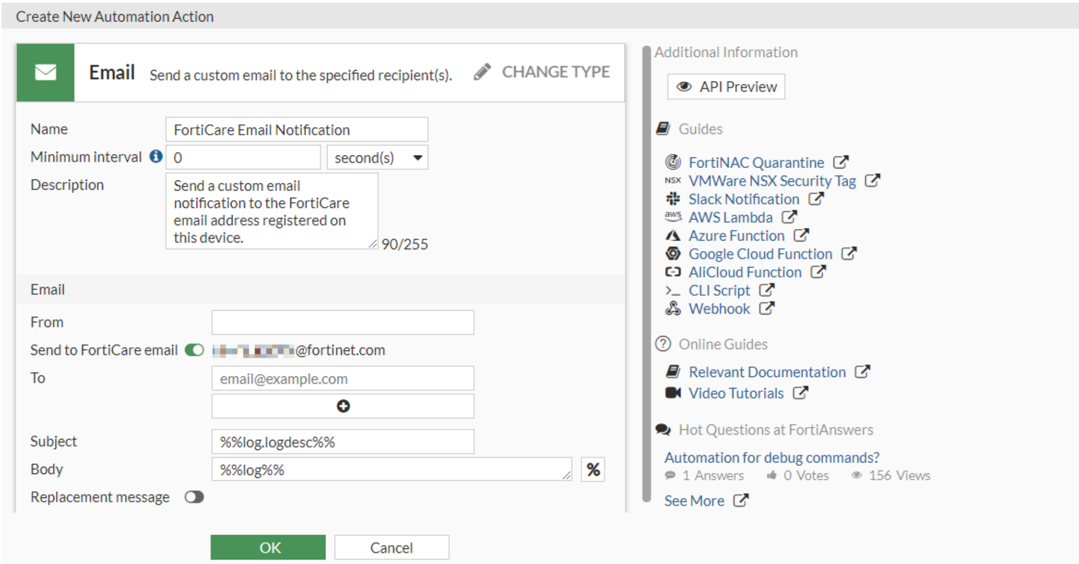
The FortiCare email address can be used in an Email action by enabling the *Send to FortiCare email* field. When enabled, FortiOS will automatically include the email address associated with the FortiCare Support entitlement. This is the FortiCloud email address visible on the *System > FortiGuard* page under the *FortiCare Support* license information.

If *Send to FortiCare email* is enabled, other email addresses can still be included in the action.

To configure an Email action with a FortiCare email address in the GUI:

1. Go to *Security Fabric > Automation* and select the *Action* tab.
2. Click *Create New* and select *Email*.
3. Enter the following:

Name	FortiCare Email Notification
Description	Send a custom email notification to the FortiCare email address registered on this device.
Send to FortiCare email	Enable
Subject	%%log.logdesc%%
Body	%%log%%



4. Click *OK*.

To configure an Email action with a FortiCare email address in the CLI:

```
config system automation-action
    edit "FortiCare Email Notification"
        set description "Send a custom email notification to the FortiCare email address
registered on this device."
        set action-type email
        set forticare-email enable
        set email-subject "%%log.logdesc%%"
    next
end
```

Asset Identity Center

This section includes information about Asset Identity Center related new features:

- Configure Purdue Levels for Fabric devices 7.4.2 on page 704

Configure Purdue Levels for Fabric devices - 7.4.2

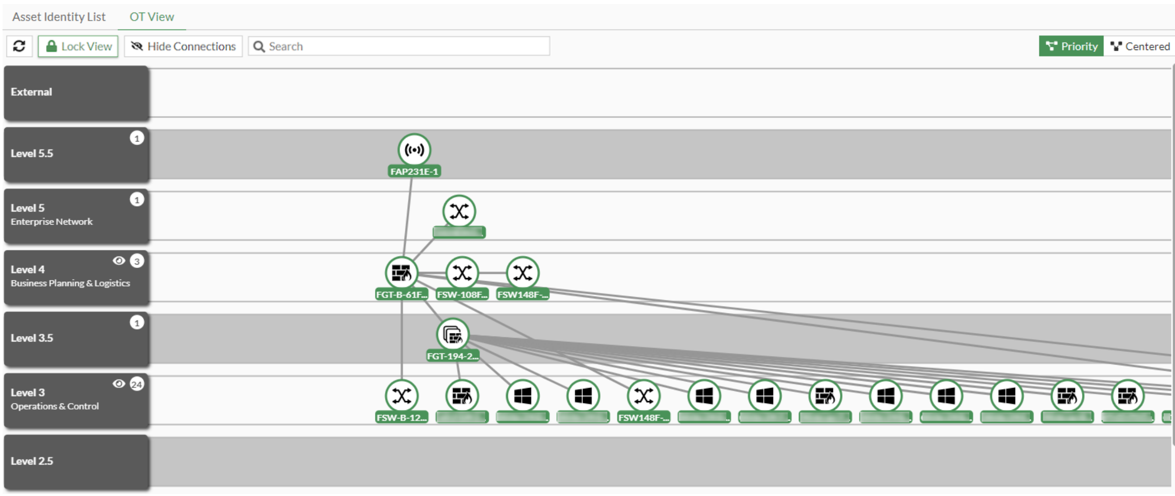
This information is also available in the FortiOS 7.4 Administration Guide:

- OT asset visibility and network topology

FortiOS now supports configurable Purdue levels for Fortinet Inc. Fabric devices, specifically FortiGates, managed FortiSwitches, and FortiAPs. This means that users have the flexibility to adjust the Purdue levels of these devices according to their specific needs and preferences, enhancing the adaptability and functionality of their Fabric devices. The default Purdue Level for these devices is 3.

To configure the Purdue Level in the GUI:

1. Go to *Security Fabric* > *Asset Identity Center*.
2. Select *OT View*.
3. Click *Unlock View*.
4. Drag and drop the FortiGate, managed FortiSwitch, or FortiAP to the desired Purdue Level.



5. Optionally, click *Lock View* to revert to the locked view.

To configure the FortiGate Purdue Level in the CLI:

```
config system global
    set purdue-level <level 1 - 5.5>
end
```

To configure the managed FortiSwitch Purdue Level in the CLI:

```
config switch-controller managed-switch
    edit "<managed FortiSwitch name>"
        set purdue-level <level 1 - 5.5>
    next
end
```

To configure the FortiAP Purdue Level in the CLI:

```
config wireless-controller wtp
    edit "<WTP ID>"
        set purdue-level <level 1 - 5.5>
    next
end
```

Log and report

This section includes information about logging and reporting related new features:

Logging

This section includes information about logging related new features:

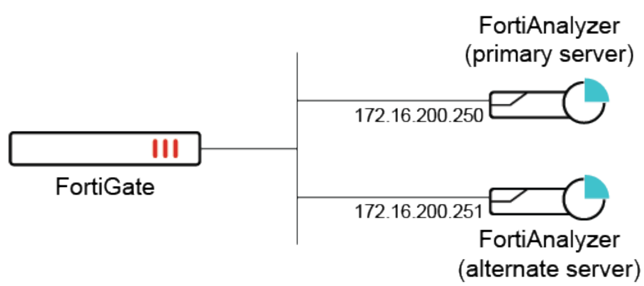
Support switching to an alternate FortiAnalyzer if the main FortiAnalyzer is unavailable - 7.4.1

This information is also available in the FortiOS 7.4 Administration Guide:

- Switching to an alternate FortiAnalyzer if the main FortiAnalyzer is unavailable

FortiOS supports switching to an alternate FortiAnalyzer if the main FortiAnalyzer is unavailable. Once the connectivity is restored, it will automatically fall back to the primary FortiAnalyzer.

This feature can be used in multi VDOM mode when FortiAnalyzer override settings are configured.



To configure switching to an alternate FortiAnalyzer when the main FortiAnalyzer is unavailable:

1. Configure primary and alternate FortiAnalyzer servers:

```
config log fortianalyzer setting
    set status enable
    set server "172.16.200.250"
```

```
    set alt-server "172.16.200.251"
    set fallback-to-primary enable
    set serial "FAZ-VMTM22000000" "FAZ-VMTM23000003"
end
```

2. Verify the primary and alternate FortiAnalyzer server IPs:

```
# diagnose test application fgtlogd 1
vdom-admin=1
mgmt=vdom1

fortilog:
faz: global , enabled
        server=172.16.200.250, alt-server=172.16.200.251, active-server=172.16.200.250,
realtime=3, ssl=1, state=connected
        server_log_status=Log is allowed.,
        src=, mgmt_name=FGh_Log_vdom1_172.16.200.250, reliable=0, sni_prefix_type=none,
        required_entitlement=none, region=ca-west-1,
        logsync_enabled:1, logsync_conn_id:65535, seq_no:0
        disconnect_jiffies:0
                status: ver=6, used_disk=0, total_disk=0, global=0, vfid=0 conn_
verified=Y
                SNs: last sn update:11 seconds ago.
                        Sn list:
                        (FAZ-VMTM22000000,age=11s)      (FAZ-VMTM23000003,age=12s)
                queue: qlen=0.
filter: severity=6, sz_exclude_list=0
         traffic virus webfilter ips emailfilter anomaly voip dlp app-ctrl waf dns ssh
ssl file-filter icap sctp-filter virtual-patch
subcategory:
        traffic: forward local multicast sniffer ztna
        virus:all subcategories are enabled.
        webfilter:all subcategories are enabled.
        ips:all subcategories are enabled.
        emailfilter:all subcategories are enabled.
        anomaly:all subcategories are enabled.
        voip:all subcategories are enabled.
        dlp:all subcategories are enabled.
        app-ctrl:all subcategories are enabled.
        waf:all subcategories are enabled.
        dns:all subcategories are enabled.
        ssh:all subcategories are enabled.
        ssl:all subcategories are enabled.
        file-filter:all subcategories are enabled.
        icap:all subcategories are enabled.
        sctp-filter:all subcategories are enabled.
        virtual-patch:all subcategories are enabled.

        server: global, id=0, ready=1, name=172.16.200.250 addr=172.16.200.250:514
        oftp-state=connected
        primary oftp status:null
        probe oftp status:null, 442
```

The 172.16.200.250 server is currently active and acting as the primary FortiAnalyzer.

3. Make the primary FortiAnalyzer server go down. The FortiGate will automatically connect to the alternate FortiAnalyzer server.
4. Verify the FortiAnalyzer server status information:

```
# diagnose  test application fgtlogd 1
vdom-admin=1
mgmt=vdom1

fortilog:
faz: global , enabled
        server=172.16.200.250, alt-server=172.16.200.251, active-server=172.16.200.251,
realtime=3, ssl=1, state=connected
        server_log_status=Log is allowed.,
        src=, mgmt_name=FGh_Log_vdom1_172.16.200.250, reliable=0, sni_prefix_type=none,
        required_entitlement=none, region=ca-west-1,
        logsync_enabled:1, logsync_conn_id:65535, seq_no:0
        disconnect_jiffies:0
                status: ver=6, used_disk=0, total_disk=0, global=0, vfid=0 conn_
verified=Y
                SNs: last sn update:30 seconds ago.
                        Sn list:
                        (FAZ-VMTM22000000,age=30s)      (FAZ-VMTM23000003,age=31s)
                queue: qlen=0.
filter: severity=6, sz_exclude_list=0
         traffic virus webfilter ips emailfilter anomaly voip dlp app-ctrl waf dns ssh
ssl file-filter icap sctp-filter virtual-patch
subcategory:
        traffic: forward local multicast sniffer ztna
        virus:all subcategories are enabled.
        webfilter:all subcategories are enabled.
        ips:all subcategories are enabled.
        emailfilter:all subcategories are enabled.
        anomaly:all subcategories are enabled.
        voip:all subcategories are enabled.
        dlp:all subcategories are enabled.
        app-ctrl:all subcategories are enabled.
        waf:all subcategories are enabled.
        dns:all subcategories are enabled.
        ssh:all subcategories are enabled.
        ssl:all subcategories are enabled.
        file-filter:all subcategories are enabled.
        icap:all subcategories are enabled.
        sctp-filter:all subcategories are enabled.
        virtual-patch:all subcategories are enabled.

        server: global, id=0, ready=1, name=172.16.200.250 addr=172.16.200.250:514
        oftp-state=connected
        probe oftp status:null, 38
```

The 172.16.200.251 server is currently active and acting as the primary FortiAnalyzer.

5. Restore the connection to the 172.16.200.250 server. The FortiGate will automatically reconnect to this FortiAnalyzer server.
6. Verify the FortiAnalyzer server status information:

```
# diagnose test application fgtlogd 1
vdom-admin=1
mgmt=vdom1

fortilog:
faz: global , enabled
```

```
        server=172.16.200.250, alt-server=172.16.200.251, active-server=172.16.200.250,
realtime=3, ssl=1, state=connected
        server_log_status=Log is allowed.,
        src=, mgmt_name=FGh_Log_vdom1_172.16.200.250, reliable=0, sni_prefix_type=none,
        required_entitlement=none, region=ca-west-1,
        logsync_enabled:1, logsync_conn_id:65535, seq_no:0
        disconnect_jiffies:0
                status: ver=6, used_disk=0, total_disk=0, global=0, vfid=0 conn_
verified=Y
                SNs: last sn update:11 seconds ago.
                        Sn list:
                        (FAZ-VMTM22000000,age=58s)      (FAZ-VMTM23000003,age=59s)
                queue: qlen=0.
filter: severity=6, sz_exclude_list=0
         traffic virus webfilter ips emailfilter anomaly voip dlp app-ctrl waf dns ssh
ssl file-filter icap sctp-filter virtual-patch
subcategory:
        traffic: forward local multicast sniffer ztna
        virus:all subcategories are enabled.
        webfilter:all subcategories are enabled.
        ips:all subcategories are enabled.
        emailfilter:all subcategories are enabled.
        anomaly:all subcategories are enabled.
        voip:all subcategories are enabled.
        dlp:all subcategories are enabled.
        app-ctrl:all subcategories are enabled.
        waf:all subcategories are enabled.
        dns:all subcategories are enabled.
        ssh:all subcategories are enabled.
        ssl:all subcategories are enabled.
        file-filter:all subcategories are enabled.
        icap:all subcategories are enabled.
        sctp-filter:all subcategories are enabled.
        virtual-patch:all subcategories are enabled.

        server: global, id=0, ready=1, name=172.16.200.250 addr=172.16.200.250:514
        oftp-state=connected
        primary oftp status:null
        probe oftp status:null, 530
```

The 172.16.200.250 server is currently active and acting as the primary FortiAnalyzer again.

To manually switch from the primary to alternate FortiAnalyzer (and vice-versa):

```
# execute log {fortianalyzer | fortianalyzer2 | fortianalyzer3} manual-failover
```

If the primary server is still up, the behavior resulting from running this command is based on the `fallback-to-primary` setting configured in the global FortiAnalyzer log settings.

- If `fallback-to-primary` is enabled (default), running `execute log fortianalyzer manual-failover` will switch to the alternate FortiAnalyzer, but it will switch back to the primary since it is not actually down.
- If `fallback-to-primary` is disabled, running `execute log fortianalyzer manual-failover` will switch to the alternate FortiAnalyzer, and it will not switch back to the primary.

Introduce new log fields for long-live sessions - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- Log fields for long-lived sessions

Logging of long-live session statistics can be enabled or disabled in traffic logs.

```
config log setting
     set long-live-session-stat {enable | disable}
end
```

When enabled, traffic logs include the following fields of statistics for long-live sessions:

Duration delta (`durationdelta`)	Displays the time in seconds between the last session log and the current session log.
Sent packet delta (`sentpktdelta`)	Displays the number of sent packets. When the number of packets reported in the `sentpktdelta` field matches the number of bytes reported in the `sentpkt` field, it shows no missing logs.
Received packet delta (`rcvdpktdelta`)	Displays the number of received packets. When the number of packets reported in the `rcvdpktdelta` field matches the number of bytes reported in the `rcvdpkt` field, it shows no missing logs.

The long-live session fields enhance the granularity and accuracy of traffic longs to aid troubleshooting and analysis.

Example

In this example, logging is enabled for long-live session statistics. Log ID 20 includes the new fields for long-live sessions.

To log long-live session statistics:

1. Enable logging of long-live session statistics:

```
config log setting
     set long-live-session-stat enable
end
```

2. View information in the logs:

   In the following example, log fields are filtered for log ID 0000000020 to displays the new fields of data.

   The `sentpkt` field displays `205` bytes, and the `rcvdpkt` field displays `1130` bytes. The new fields (`sentpktdelta=205` and `rcvdpktdelta=1130`) display the same number of packets, which shows no logs have been lost. The `durationdelta` shows `120` seconds between the last session log and the current session log.

```
# execute log filter device Disk

# execute log filter category 0

# execute log filter field subtype forward
```

```
# execute log filter field logid 0000000020

# execute log display

1 logs found.

1 logs returned.

1: date=2023-12-07 time=14:19:59 eventtime=1701987599439429340 tz="-0800"
logid="0000000020" type="traffic" subtype="forward" level="notice" vd="vdom1"
srcip=10.1.100.22 srcport=53540 srcintf="wan2" srcintfrole="undefined"
dstip=172.16.200.55 dstport=80 dstintf="wan1" dstintfrole="undefined"
srccountry="Reserved" dstcountry="Reserved" sessionid=296 proto=6 action="accept"
policyid=1 policytype="policy" poluuid="e538d622-53eb-51ee-8adc-f8fbb0f22fdd"
policyname="B-out" service="HTTP" trandisp="snat" transip=172.16.200.2 transport=53540
duration=120 sentbyte=10855 rcvdbyte=1397640 sentpkt=205 rcvdpkt=1130 appcat="unscanned"
sentdelta=10855 rcvddelta=1397640 durationdelta=120 sentpktdelta=205 rcvdpktdelta=1130
```

Cloud

This section includes information about cloud related new features:

Public and private cloud

This section includes information about public and private cloud related new features:

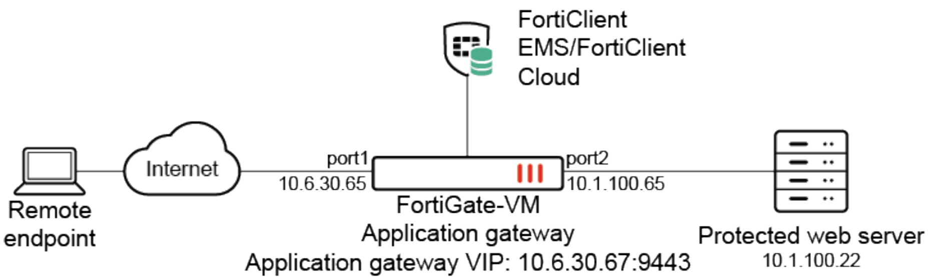
Support the AWS t4g, c6a, and c6in instance families

FortiGate-VM supports the AWS t4g instance family using the FGT-ARM64-AWS image. FortiGate-VM supports the AWS c6a and c6in instance family using the FGT-VM64-AWS image. See Instance type support.

VMware ESXi FortiGate-VM as ZTNA gateway

FortiOS supports deploying a VMware ESXi FortiGate-VM directly as a zero trust application gateway using the OVF template (.vapp). You can configure zero trust network access (ZTNA)-related parameters such as the EMS server, external and internal interface IP addresses, and the application server mapping, during OVF deployment. The deployment also bootstraps ZTNA policy, authentication scheme, rules, and user group configurations.

This enhancement introduces a new FortiGate-VM64-ZTNA-vapp.ovf file. With this file, you can configure all ZTNA-related parameters and the FGT-VM64 instance can act as a ZTNA gateway after bootstrapping. The file supports using FortiClient Cloud or on-premise EMS.

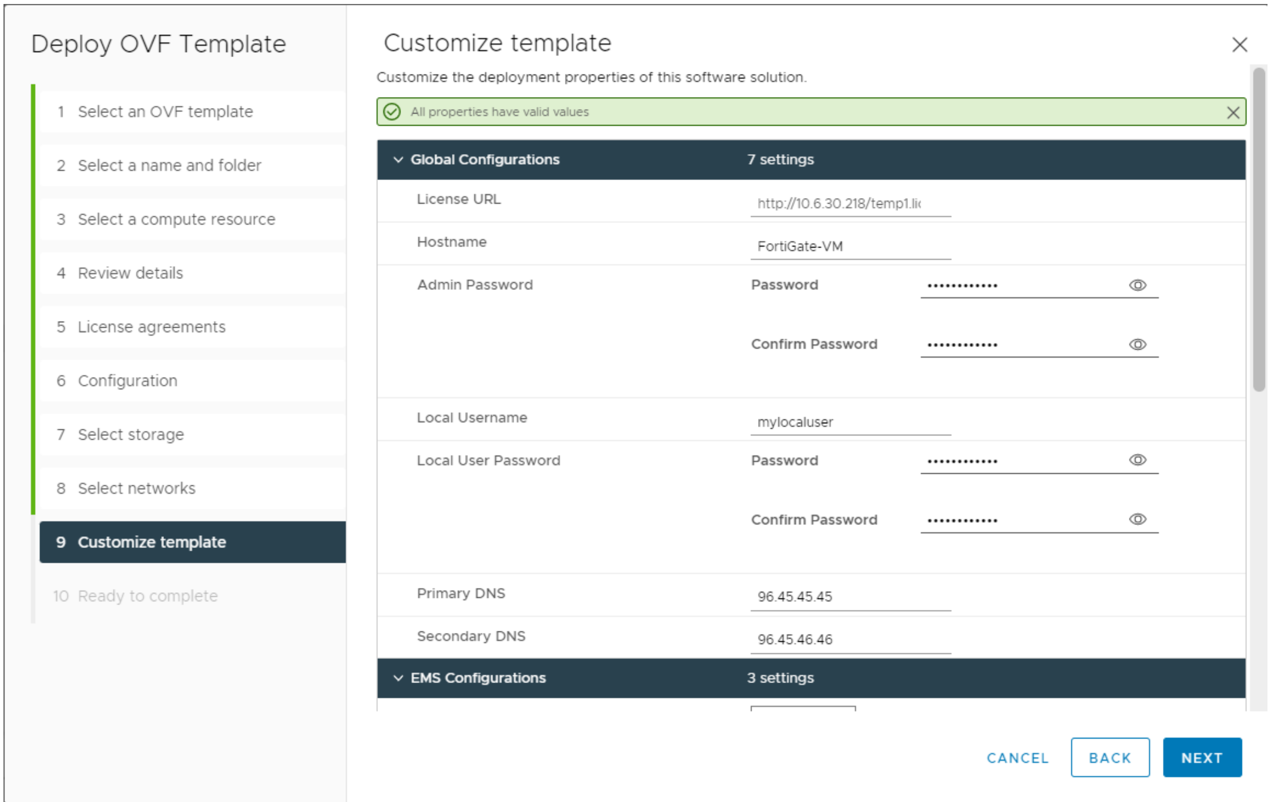


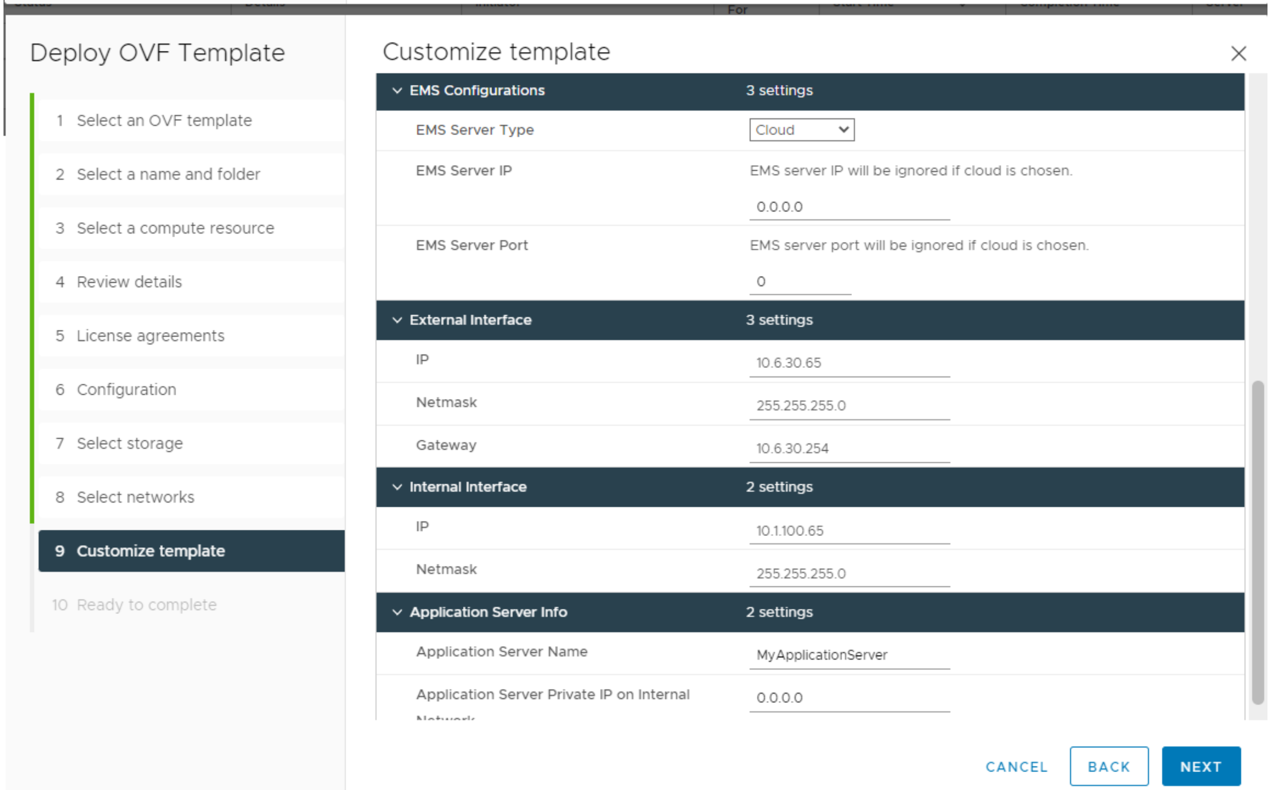
The example deployment is as follows:

- The FortiGate is deployed with the aforementioned addressing scheme.
- FortiClient Cloud is used.
- 10.6.30.67 is used for the HTTPS access proxy external IP address.
- The web sever 10.1.100.22 is configured for server mapping.
- A local user, mylocaluser, is created on the FortiGate and added to ztna_group.
- ztna_group is allowed ZTNA to the protected web server via basic authentication.
- This deployment does not use ZTNA tags for security posture check.

To deploy VMware ESXi FortiGate-VM as ZTNA gateway:

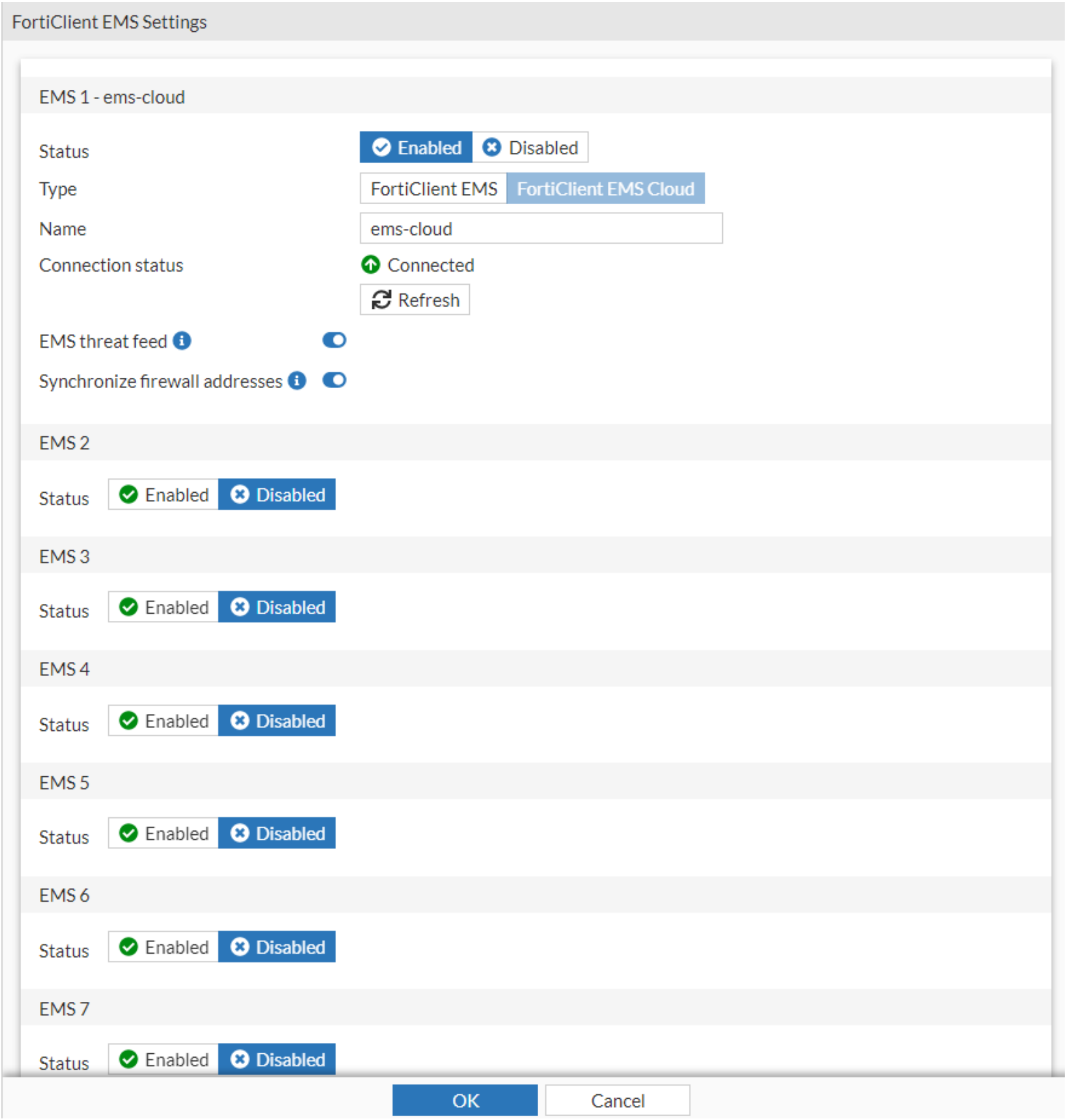
1. Download the OVF package:
   a. In the Fortinet Customer Service & Support site, go to *Support* > *Downloads* > *VM Images*.
   b. From the *Select Platform* dropdown list, select *VMWare ESXi*.
   c. Download the file labeled as *New deployment of FortiGate for VMware FGT_VM64-v7.4.0.F-buildXXXX-FORTINET.out.ovf.zip*.
   d. Extract the zip file and locate the FortiGate-VM64-ZTNA.vapp.ovf file.
2. In vSphere, create a new FGT-VM64 instance using the FortiGate-VM64-ZTNA.vapp.ovf file. You can configure the VM license file and all ZTNA-related parameters.





3. After the FGT-VM64 boots up, go to *Security Fabric > Fabric Connectors*.

4. Verify EMS. EMS authorizes the FortiGate.



You can run `diagnose debug cloudinit show` to view the cloudinit information after the FortiGate boots up:

```
FortiGate-VM # diagnose debug cloudinit show
>> Checking metadata source ovf
>> Cloudinit downloading the license:http://10.6.30.218/temp1.lic
>> Cloudinit download the license successfully
>> Found metadata source: ovf
>> Trying to install vmlicense ...
>> Run config script
>> FortiGate-VM $ config system global
>> FortiGate-VM (global) $ set gui-theme mariner
>> FortiGate-VM (global) $ set admintimeout 60
>> FortiGate-VM (global) $ end
>> FortiGate-VM $ config system admin
>> FortiGate-VM (admin) $ edit admin
>> FortiGate-VM (admin) $ config gui-dashboard
```

```
>> FortiGate-VM (gui-dashboard) $ edit 0
>> FortiGate-VM (0) $ set name "FortiView ZTNA Servers"
>> FortiGate-VM (0) $ set vdom root
>> FortiGate-VM (0) $ set layout-type standalone
>> FortiGate-VM (0) $ set csf disable
>> FortiGate-VM (0) $ config widget
>> FortiGate-VM (widget) $ edit 1
>> FortiGate-VM (1) $ set type fortiview
>> FortiGate-VM (1) $ set width 1
>> FortiGate-VM (1) $ set height 1
>> FortiGate-VM (1) $ set csf-device all
>> FortiGate-VM (1) $ set fortiview-type ztnaServer
>> FortiGate-VM (1) $ set fortiview-sort-by bytes
>> FortiGate-VM (1) $ set fortiview-timeframe 5min
>> FortiGate-VM (1) $ set fortiview-visualization table
>> FortiGate-VM (1) $ end
>> FortiGate-VM (0) $ end
>> FortiGate-VM (admin) $ end
>> FortiGate-VM $ config system settings
>> FortiGate-VM (settings) $ set gui-implicit-policy disable
>> FortiGate-VM (settings) $ set gui-dos-policy disable
>> FortiGate-VM (settings) $ set gui-dynamic-routing disable
>> FortiGate-VM (settings) $ set gui-threat-weight disable
>> FortiGate-VM (settings) $ set gui-file-filter disable
>> FortiGate-VM (settings) $ set gui-application-control disable
>> FortiGate-VM (settings) $ set gui-endpoint-control disable
>> command parse error before 'gui-endpoint-control'
>> Command fail. Return code -61
>> FortiGate-VM (settings) $ set gui-vpn disable
>> FortiGate-VM (settings) $ set gui-wireless-controller disable
>> FortiGate-VM (settings) $ set gui-traffic-shaping disable
>> FortiGate-VM (settings) $ set gui-webfilter disable
>> FortiGate-VM (settings) $ set gui-dnsfilter disable
>> FortiGate-VM (settings) $ set allow-subnet-overlap enable
>> FortiGate-VM (settings) $ end
>> FortiGate-VM $ config user local
>> FortiGate-VM (local) $ edit mylocaluser
>> FortiGate-VM (mylocaluser) $ set type password
>> FortiGate-VM (mylocaluser) $ set passwd <password>
>> FortiGate-VM (mylocaluser) $ next
>> FortiGate-VM (local) $ end
>> FortiGate-VM $ config user group
>> FortiGate-VM (group) $ edit ztna_group
>> FortiGate-VM (ztna_group) $ set member mylocaluser
>> FortiGate-VM (ztna_group) $ next
>> FortiGate-VM (group) $ end
>> FortiGate-VM $ config firewall address
>> FortiGate-VM (address) $ edit webserver1
>> FortiGate-VM (webserver1) $ set subnet 10.1.100.22 255.255.255.255
>> FortiGate-VM (webserver1) $ next
>> FortiGate-VM (address) $ end
>> FortiGate-VM $ config firewall vip
>> FortiGate-VM (vip) $ edit MyApplicationServer
>> FortiGate-VM (MyApplicationServer) $ set type access-proxy
>> FortiGate-VM (MyApplicationServer) $ set extip 10.6.30.67
>> FortiGate-VM (MyApplicationServer) $ set extintf port1
>> FortiGate-VM (MyApplicationServer) $ set server-type https
```

```
>> FortiGate-VM (MyApplicationServer) $ set extport 9443
>> FortiGate-VM (MyApplicationServer) $ set ssl-certificate Fortinet_SSL
>> FortiGate-VM (MyApplicationServer) $ next
>> FortiGate-VM (vip) $ end
>> FortiGate-VM $ config firewall access-proxy
>> FortiGate-VM (access-proxy) $ edit MyApplicationServer
>> FortiGate-VM (MyApplicationServer) $ set vip MyApplicationServer
>> FortiGate-VM (MyApplicationServer) $ config api-gateway
>> FortiGate-VM (api-gateway) $ edit 1
>> FortiGate-VM (1) $ config realservers
>> FortiGate-VM (realservers) $ edit 1
>> FortiGate-VM (1) $ set ip 10.1.100.22
>> FortiGate-VM (1) $ next
>> FortiGate-VM (realservers) $ end
>> FortiGate-VM (1) $ next
>> FortiGate-VM (api-gateway) $ end
>> FortiGate-VM (MyApplicationServer) $ next
>> FortiGate-VM (access-proxy) $ end
>> FortiGate-VM $ config firewall proxy-policy
>> FortiGate-VM (proxy-policy) $ edit 1
>> FortiGate-VM (1) $ set name ZTNA-Web-Server
>> FortiGate-VM (1) $ set proxy access-proxy
>> FortiGate-VM (1) $ set access-proxy MyApplicationServer
>> FortiGate-VM (1) $ set srcintf port1
>> FortiGate-VM (1) $ set srcaddr all
>> FortiGate-VM (1) $ set dstaddr webserver1
>> FortiGate-VM (1) $ set action accept
>> FortiGate-VM (1) $ set schedule always
>> FortiGate-VM (1) $ set logtraffic all
>> FortiGate-VM (1) $ set groups ztna_group
>> FortiGate-VM (1) $ next
>> FortiGate-VM (proxy-policy) $ end
>> FortiGate-VM $ config authentication scheme
>> FortiGate-VM (scheme) $ edit ZTNA
>> FortiGate-VM (ZTNA) $ set method basic
>> FortiGate-VM (ZTNA) $ set user-database local-user-db
>> FortiGate-VM (ZTNA) $ next
>> FortiGate-VM (scheme) $ end
>> FortiGate-VM $ config authentication rule
>> FortiGate-VM (rule) $ edit ZTNA
>> FortiGate-VM (ZTNA) $ set srcintf port1
>> FortiGate-VM (ZTNA) $ set srcaddr all
>> FortiGate-VM (ZTNA) $ set ip-based disable
>> FortiGate-VM (ZTNA) $ set active-auth-method ZTNA
>> FortiGate-VM (ZTNA) $ next
>> FortiGate-VM (rule) $ end
>> FortiGate-VM $ config endpoint-control fctems
>> FortiGate-VM (fctems) $ edit 1
>> FortiGate-VM (1) $ set name ems-cloud
>> FortiGate-VM (1) $ set status enable
>> FortiGate-VM (1) $ set fortinetone-cloud-authentication enable
>> FortiGate-VM (1) $ next
>> The configuration will not be effective unless server certificate is verified.
>> You can get and verify server certificate by the following command:
>> "execute fctems verify 1" (ems table id)
>> FortiGate-VM (fctems) $ end
>> Finish running config script
```

Support the new AWS c7gn instance family

FortiGate-VM supports the new AWS c7gn instance family using the FGT-ARM64-AWS image. See Instance type support.

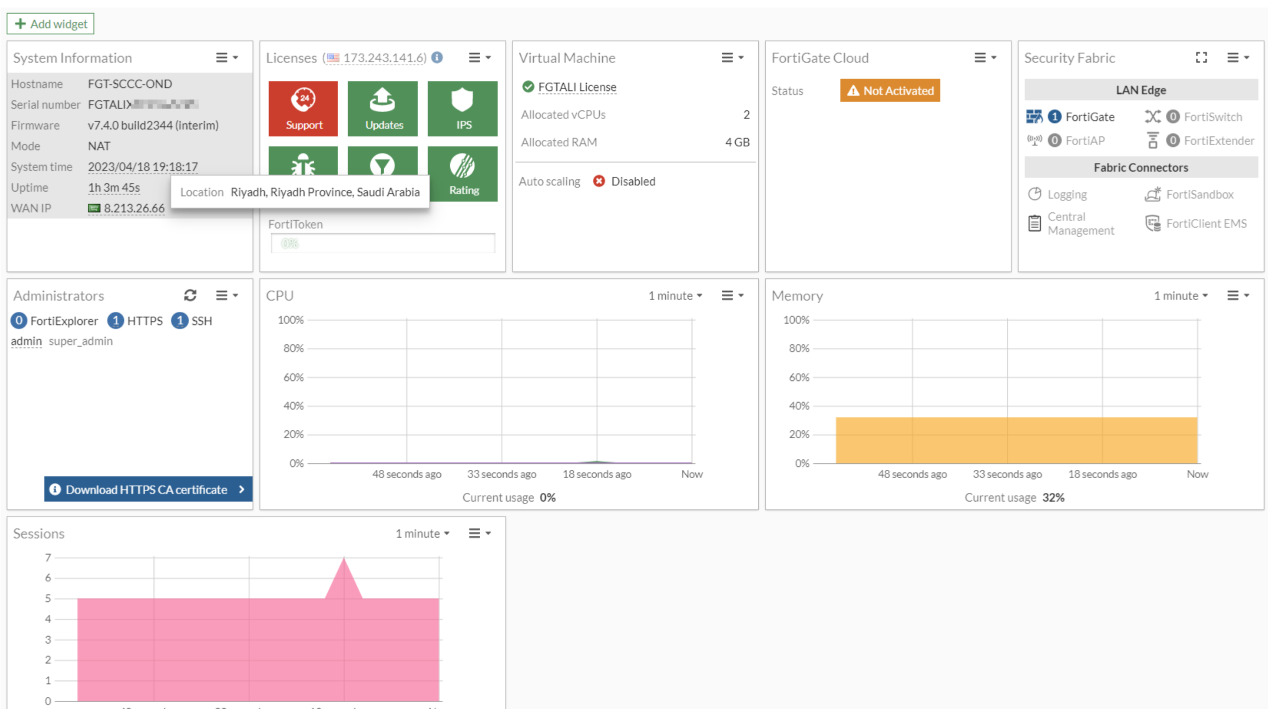
Support SCCC backed by AliCloud

FortiOS 7.4.0 supports Saudi Cloud Computing Company (SCCC) and the domain alibabacloud.sa, a standalone cloud that AliCloud backs. This includes support for the SCCC region, me-central-1. You can create FortiGate-VM custom, standalone, and high availability images on AliCloud SCCC.

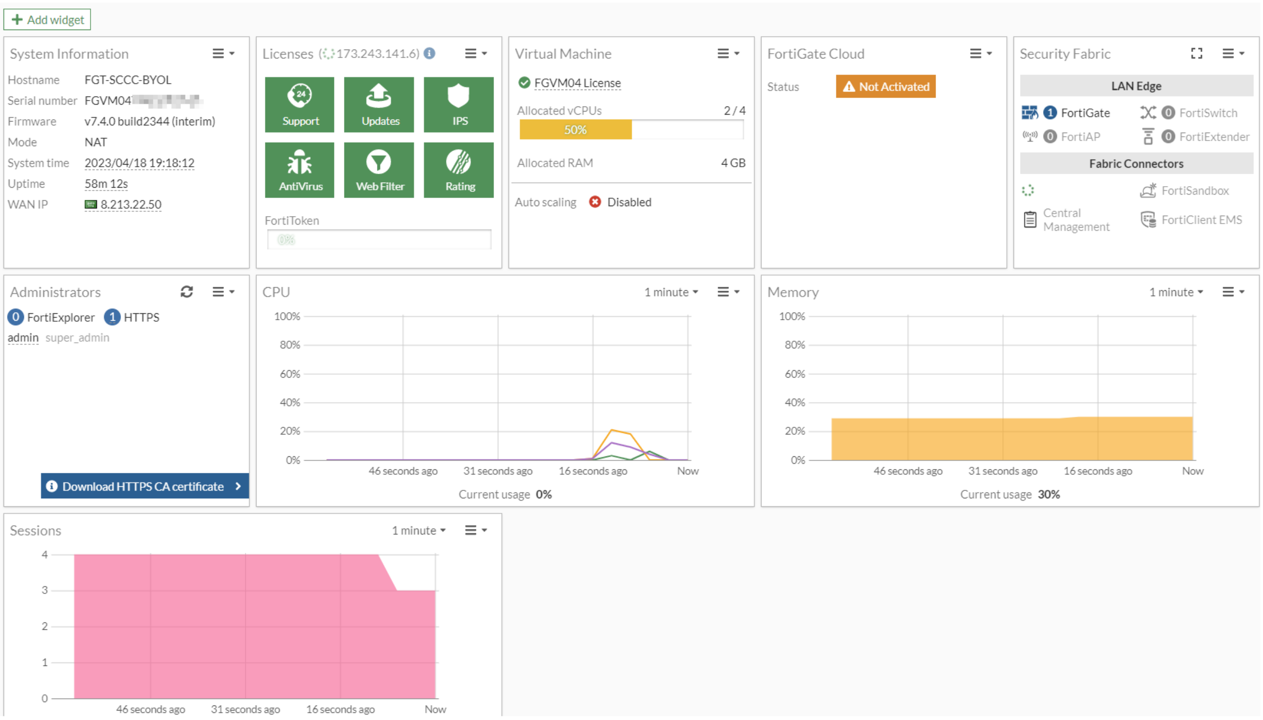
As SCCC is a separate region from other AliCloud regions, it requires a different user account.

Fortinet images are not available on SCCC marketplace. You deploy FortiGate-VMs on SCCC manually by uploading to object storage and creating a custom image.

The following shows the GUI for an on-demand instance deployed on SCCC:



The following shows the GUI for a bring your own license instance deployed on SCCC:



The following shows CLI commands which use the SCCC region me-central-1 to configure a SDN connector to SCCC:

```
config system sdn-connector
   edit "myali"
      set type alicloud
      set access-key "LTAxxxxxxxxxxxxxfQR"
      set secret-key xxxxxxxxxxxx
      set region "me-central-1" <====now FGT-ALI support this new region id "me-central-1"
            for Aliyun SCCC
   next
end
```

Upgrade AWS ENA network interface driver to 2.8.3

FortiOS 7.4.0 upgrades the FortiGate-VM AWS ENA network interface driver from 2.6.1g to 2.8.3. The AWS ENA driver 2.8.3 introduces performance and stability optimizations over the previously used 2.6.1 driver. It also prepares FortiGate-VM for new features that newer instance types include.

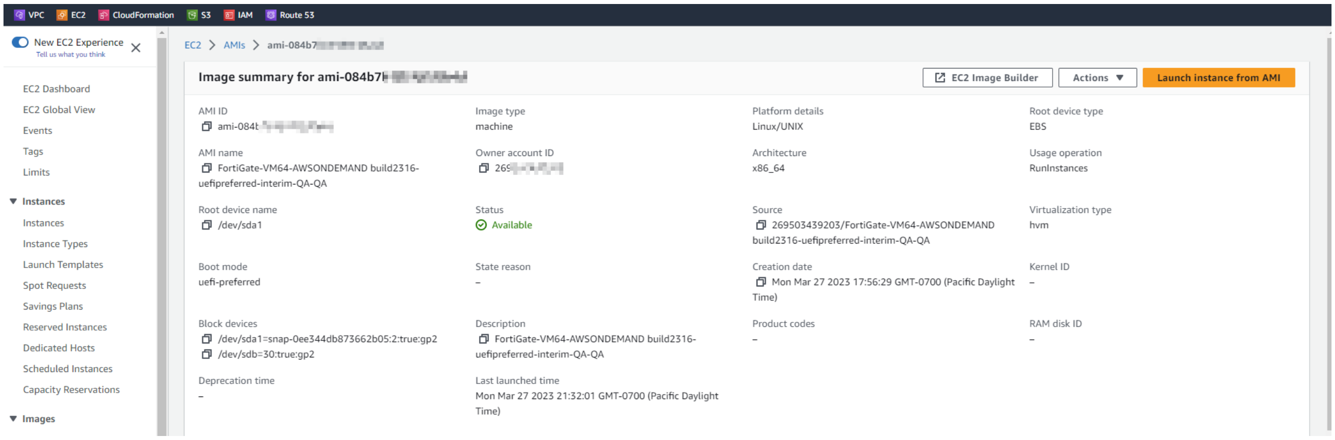
You can confirm the ENA driver version by running the `get hardware nic port1` command:

```
Name: port1
Driver: ena
Version: 2.8.3g
```

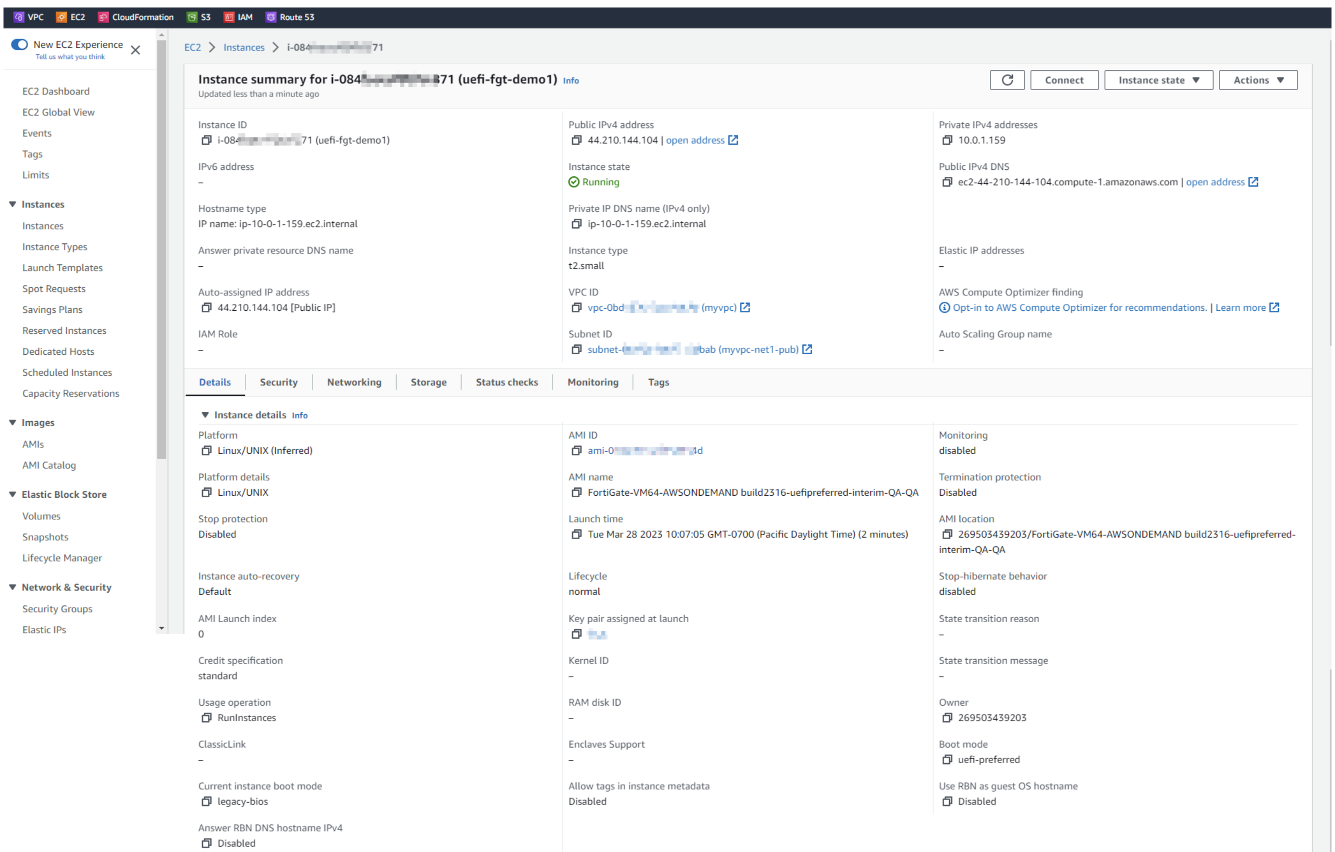
Support UEFI-Preferred boot mode on AWS FortiGate-VM models

When deployed on instance types that support `--boot-mode uefi-preferred`, FortiGate-VM on AWS supports UEFI-Preferred boot mode. You can label AMI images as UEFI-Preferred and boot with UEFI when the instance type supports UEFI.

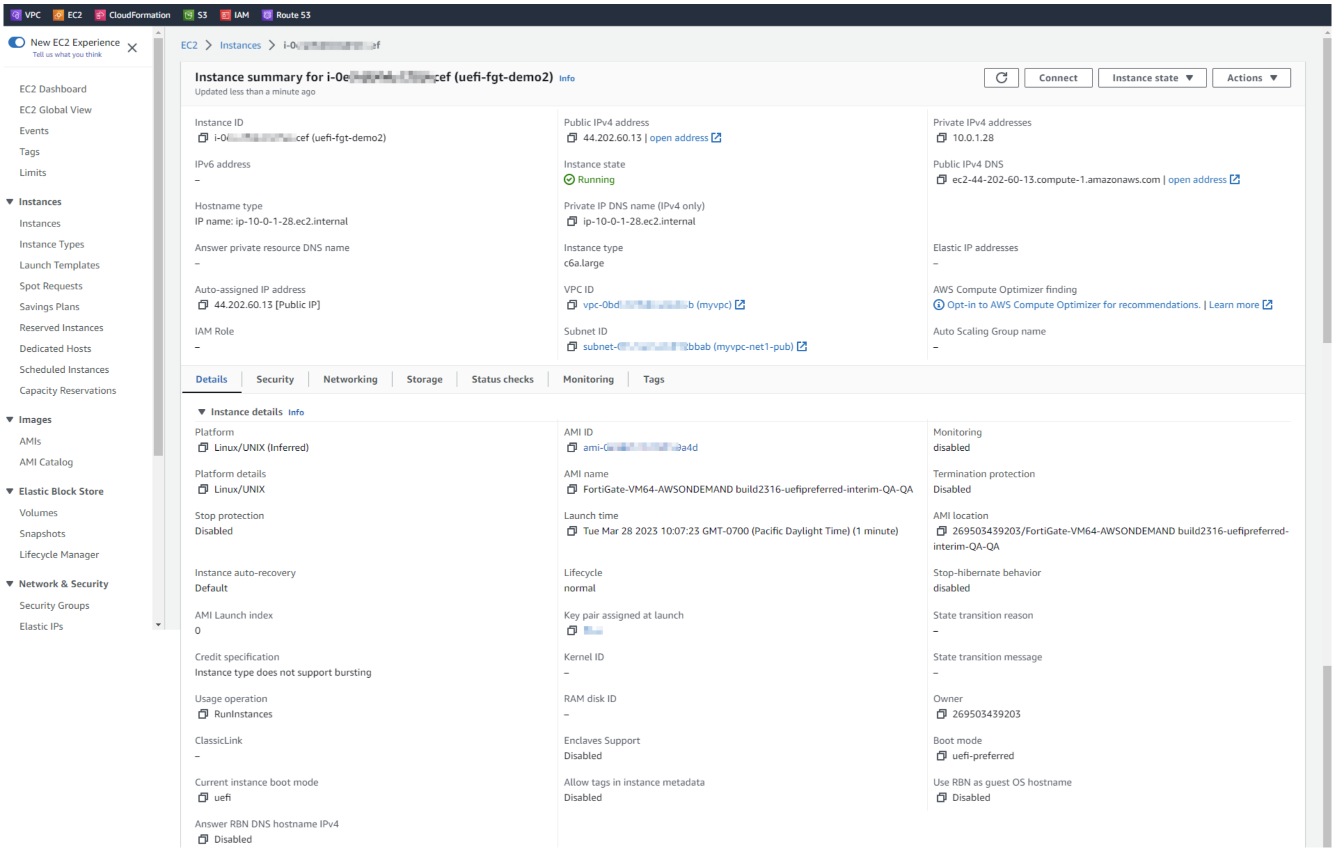
You can register a FortiGate-VM64-AWS custom image with the `--boot-mode uefi-preferred` option.



If the instance type only supports legacy BIOS boot mode, the FortiGate-VM64-AWS boots in BIOS mode even if it is labelled as `--boot-mode uefi-preferred`. For example, the t2.small instance type does not support UEFI-Preferred boot mode.



If the instance type supports legacy BIOS and UEFI boot modes, the FortiGate-VM64-AWS boots in UEFI mode if it is labelled as `--boot-mode uefi-preferred`. For example, the c6a.large instance type supports legacy BIOS and UEFI boot modes.



OCI DRCC support

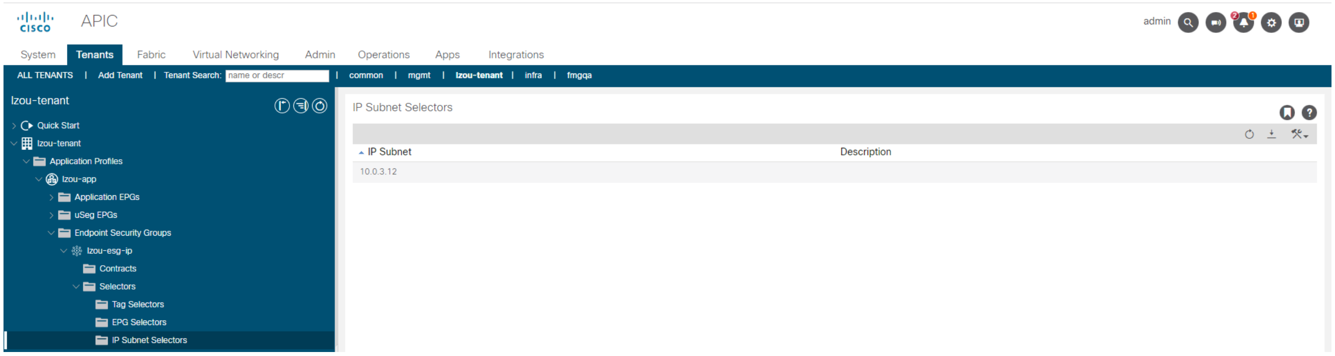
FortiGate-VM is supported in OCI Dedicated Region Cloud@Customer (DRCC). For more information, see Dedicated Region Cloud@Customer.

Support multiple compartments and regions with single OCI SDN connector

FortiOS 7.4.0 introduces the ability to set multiple regions and multiple compartments for a single OCI SDN connector. This reduces the number of SDN connectors needed for any given OCI environment that uses multiple regions and multiple compartments. You can combine a configuration that previously required multiple SDN connectors into a single SDN connector.

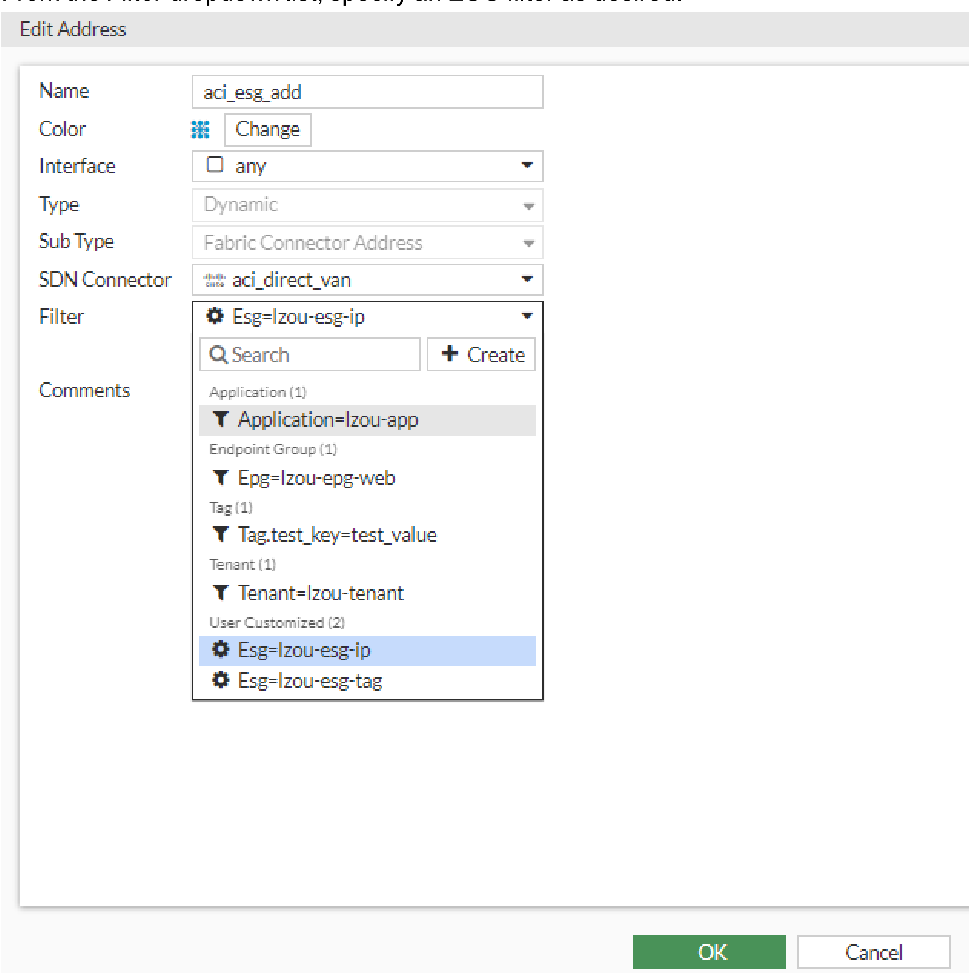
Add Cisco ACI ESG support for direct connector - 7.4.1

When integrating with Cisco ACI using a direct connection SDN connector, you can filter on the endpoint security group (ESG) when defining and resolving a dynamic address. The following shows a Cisco ACI tenant with an ESG in the Cisco ACI-side GUI:

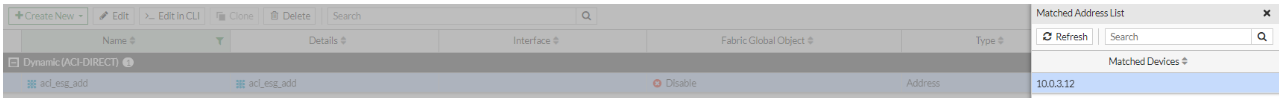


To configure a Cisco ACI SDN connector using the ESG filter using the GUI:

1. In FortiOS, go to *Security Fabric > External Connectors*.
2. Configure a Cisco ACI SDN connector. Ensure that the connector status is up.
3. Go to *Policy & Objects > Addresses*.
4. Create a dynamic firewall address. From the *Sub Type* dropdown list, select *Fabric Connector Address*.
5. From the *SDN Connector* dropdown list, select the Cisco ACI SDN connector.
6. From the *Filter* dropdown list, specify an ESG filter as desired.



7. Save the address. The resolved dynamic address can show up in dynamic firewall address configuration and is the same as the IP address configured on the Cisco ACI side.



To configure a Cisco ACI SDN connector using the ESG filter using the CLI:

1. Configure a Cisco ACI SDN connector:

```
config system sdn-connector
    edit "aci_direct_van"
        set type aci-direct
        set verify-certificate disable
        set server-list "10.59.8.35"
        set username "admin"
        set password xxxxxx
    next
end
```

2. Ensure that the connector status is up.

```
diagnose system sdn status aci_direct_van
SDN Connector                        Type        Status
-------------------------------------------------------------
aci_direct_van                       aci-direct  Up
```

3. Create a dynamic firewall address, specifying an ESG filter as desired:

```
config firewall address
    edit "aci_esg_add"
        set uuid 7b199716-1450-51ee-22bb-12b344f6b1cf
        set type dynamic
        set sdn "aci_direct_van"
        set color 17
        set filter "Esg=lzou-esg-ip"
    next
end
```

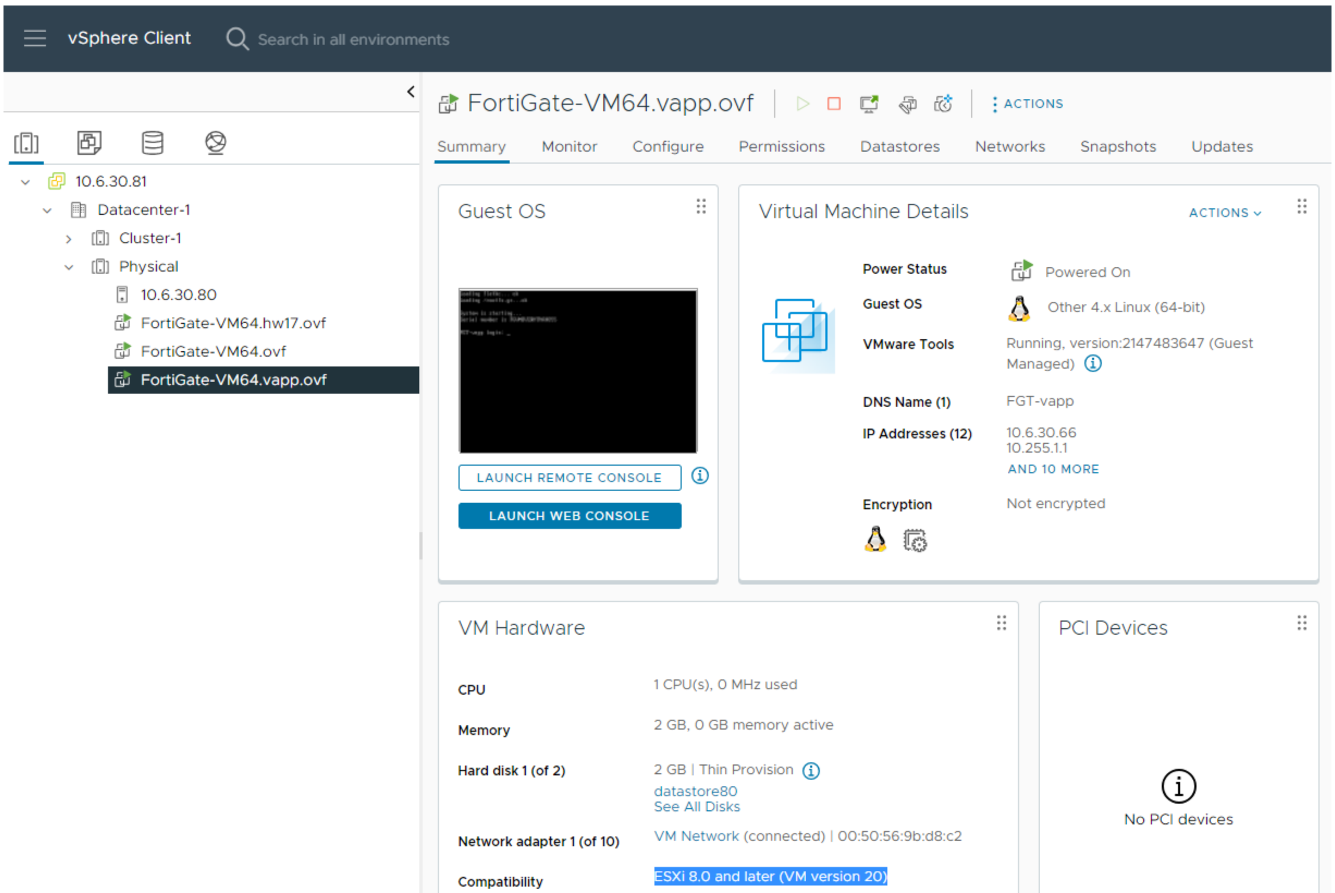
The resolved dynamic address can show up in dynamic firewall address configuration and is the same as the IP address configured on the Cisco ACI side:

```
config firewall address
    edit "aci_esg_add"
        set uuid 7b199716-1450-51ee-22bb-12b344f6b1cf
        set type dynamic
        set sdn "aci_direct_van"
        set color 17
        set filter "Esg=lzou-esg-ip"
        config list
            edit "10.0.3.12"
            next
        end
```

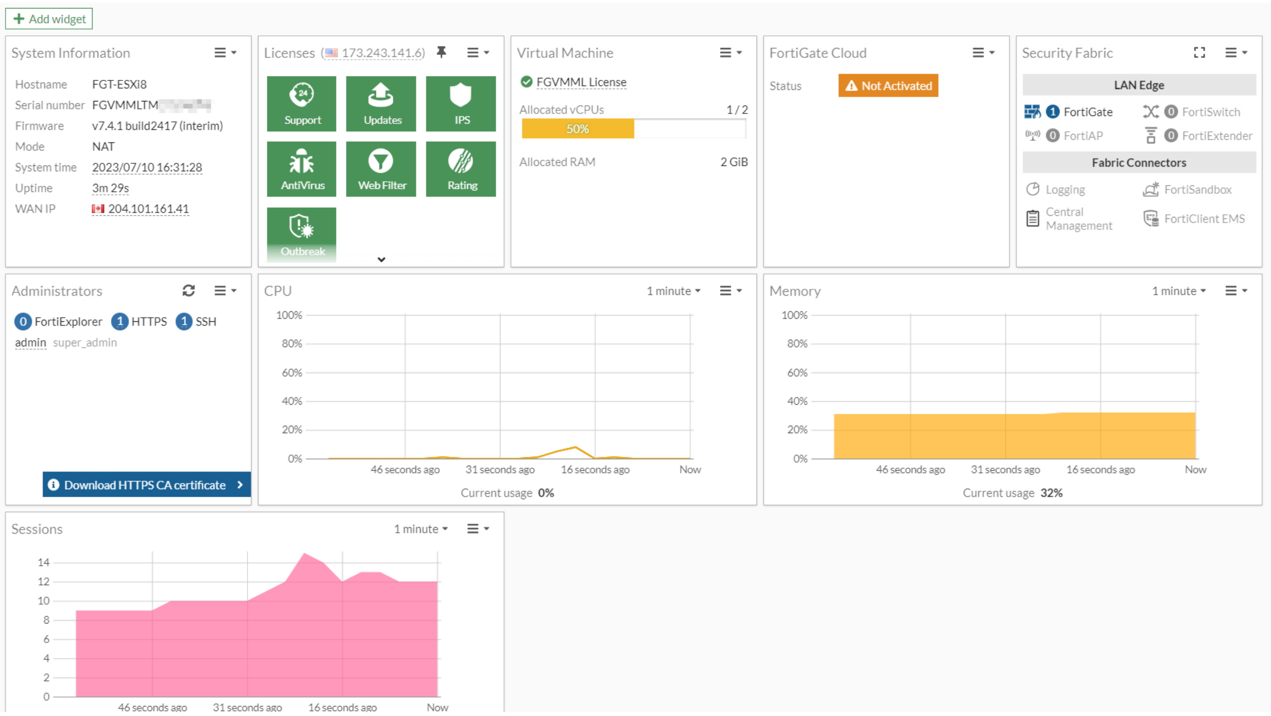
```
        next
    end
```

Add OVF template support for VMware ESXi 8 - 7.4.1

This feature introduces compatibility between the FortiGate-VM64.ovf and FortiGate-VM65.vapp.ovf templates with VMware ESXi 8, virtual hardware version 20. The following shows that you can boot up FortiGate-VM64.vapp.ovf on vSphere 8.0 from both VMware ESXi and VCSA, which is compatible with VMware ESXi 8 virtual hardware version 20.



The following shows the FortiOS GUI:



GCP support for C3 machine type - 7.4.1

FortiGate-VM supports the GCP C3 machine type family. See Machine type support.

AWS support for local zones - 7.4.1

FortiGate-VM supports certain local zones with instance types c5d.2xlarge, c5d.4xlarge, and c5d.12xlarge. See Region support.

AWS SBE support - 7.4.1

FortiOS 7.4.1 supports AWS Snowball Edge (SBE) devices, which are compute and storage resources at the edge that have a limited connection or are entirely air gapped. See Deploying FortiGate-VM on SBE.

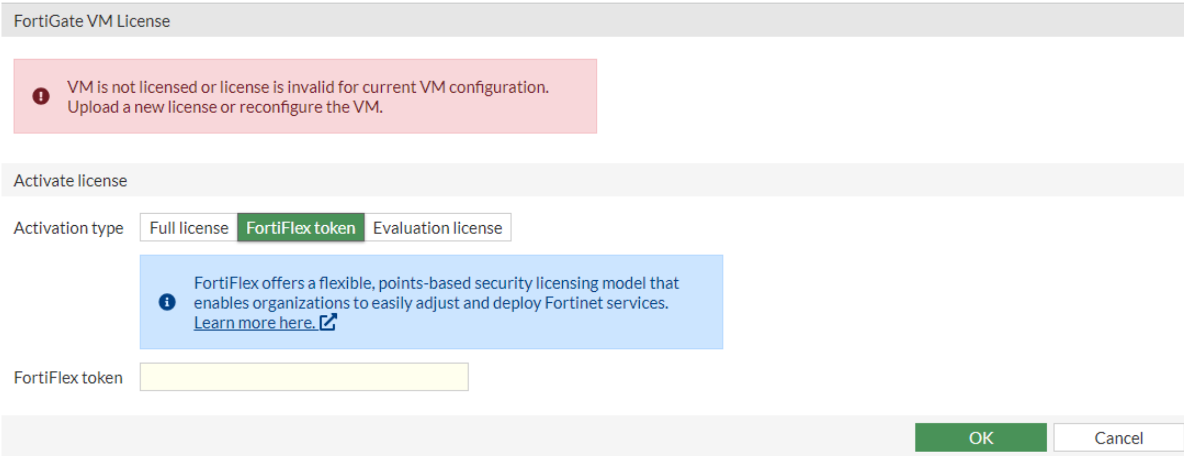
GCP support for C3A and C3D machine type - 7.4.2

FortiGate-VM supports the GCP C3A and C3D machine types. See Machine type support.

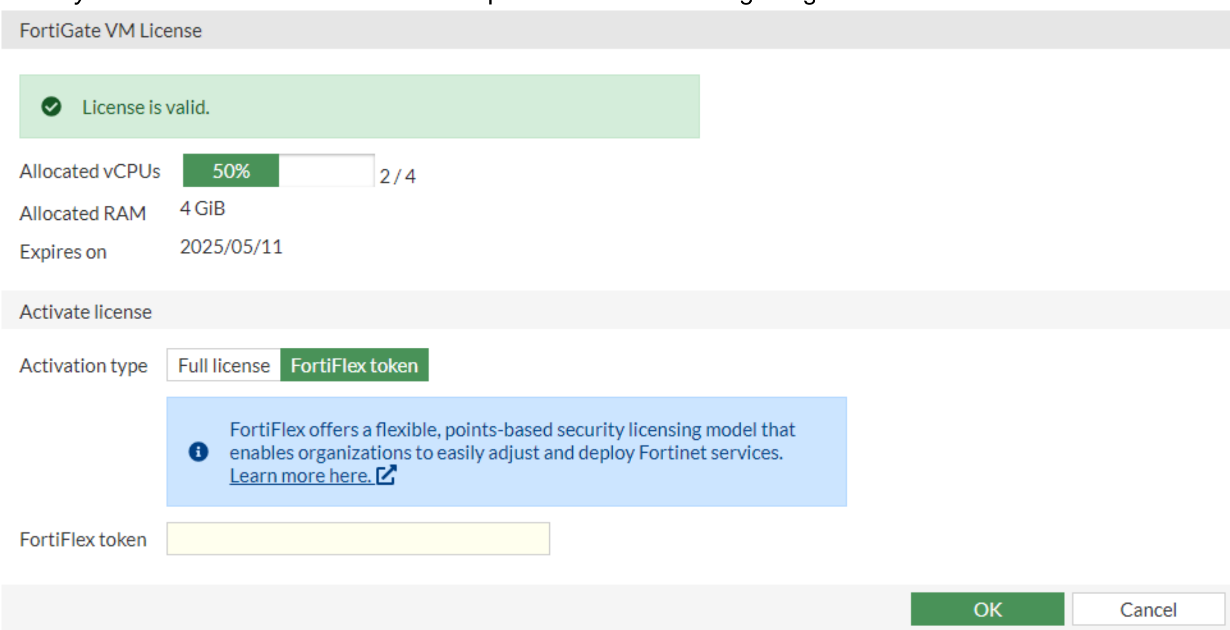
Add FortiFlex GUI option - 7.4.2

7.4.2 adds GUI support for applying a FortiFlex token on the *FortiGate VM License* page for the following VM instance type:

- Newly deployed or expired FortiGate-VM instances. After logging into the FortiOS GUI, a *FortiFlex token* option is available when the license popup appears:



- Already licensed FortiGate-VM instances. You can go to this page from the *Virtual Machine* dashboard widget or from *System > FortiGuard*. *FortiFlex token* option is available for migrating into FortiFlex:



AliCloud support for c7, c7a, and g5ne instance families - 7.4.2

FortiGate-VM supports the following AliCloud instance types that belong to the c7, c7a, and g5ne network-optimized instance families:

- ecs.g5ne.large
- ecs.g5ne.xlarge
- ecs.g5ne.2xlarge
- ecs.g5ne.4xlarge
- ecs.g5ne.8xlarge
- ecs.g5ne.16xlarge

- ecs.g5ne.18xlarge
- ecs.c7.large
- ecs.c7.xlarge
- ecs.c7.2xlarge
- ecs.c7.3xlarge
- ecs.c7.4xlarge
- ecs.c7.6xlarge
- ecs.c7.8xlarge
- ecs.c7.16xlarge
- ecs.c7.32xlarge
- ecs.c7a.large
- ecs.c7a.xlarge
- ecs.c7a.2xlarge
- ecs.c7a.4xlarge
- ecs.c7a.8xlarge
- ecs.c7a.16xlarge
- ecs.c7a-nps1.8xlarge
- ecs.c7a.32xlarge

See Instance type support.

AliCloud support change route table with IPv4 gateway for HA - 7.4.2

FortiGate supports high availability (HA) failover scenarios behind AliCloud IPv4 gateway. For information on how to set up and configure IPv4 gateway on your AliCloud virtual private cloud, see IPv4 gateway overview.

AWS SDN Connector support for alternate resources - 7.4.2

The FortiOS AWS SDN connector supports querying AWS for resource elastic IP addresses based on resource attributes such as the owner ID, resource descriptions, and tags. See SDN connector support for alternate resources.

Integrate FortiGate Azure vWAN solution with Azure Monitor to capture health metrics - 7.4.2

This information is also available in the Azure vWAN SD-WAN NGFW Deployment Guide:

- Integration with Azure Monitor to capture health metrics

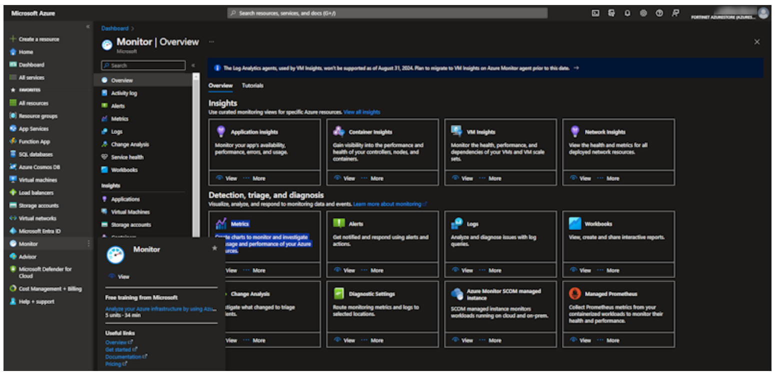
When configuring the FortiGate-VM as a Network Virtual Appliance (NVA) as part of the Azure vWAN solution, FortiGate can make API calls and send health metrics to Azure for integration with Azure Monitor.

Example

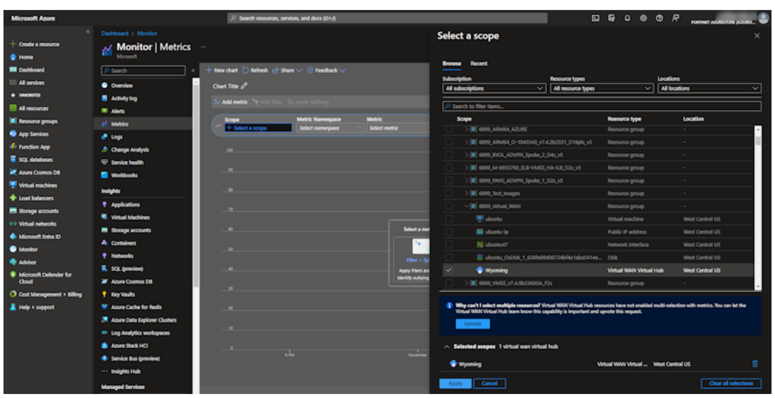
Once Azure Virtual WAN is configured, administrators can add the FortiGate vWAN virtual hub to Azure Monitor within Azure. This allows Azure to receive metrics from the FortiGate and display them in the Monitor console. For information about configuring FortiGate vWAN, see Azure vWAN SD-WAN NGFW Deployment Guide.

To add FortiGate vWAN virtual hub to Azure Monitor:

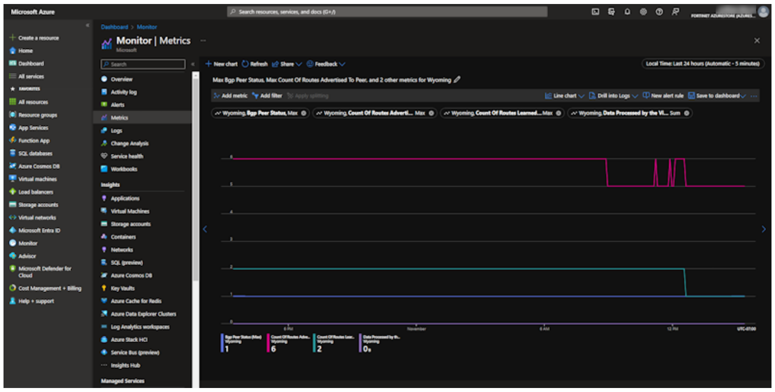
1. Go to the Azure Portal, and log in.
2. On the left navigation bar, click *Monitor*. The *Monitor* | *Overview* page is displayed.



3. On the *Monitor* | *Overview* page, click *Metrics*.
4. Click *Select a scope*, choose your *Virtual WAN Virtual Hub* object, and click *Apply*.
   In this example the Virtual WAN Virtual Hub object named *Wyoming* is selected.
   For more information about the fields, see Analyze metrics with Azure Monitor metrics explorer on the Microsoft site.



5. On the *Monitor | Metrics* page, a line chart displays the metrics for your settings.



Customizing the FortiFlex license token activation retry parameters - 7.4.2

This information is also available in the FortiOS 7.4 Administration Guide:

- VM license

FortiOS supports the customization of the retries for FortiFlex license token activation. The token activation number of retries and the interval between each attempt can be configured using the following commands, respectively:

```
execute vm-license-options count <integer>
execute vm-license-options interval <interval length in seconds>
```

If the `vm-license-options count` is set to zero, the token activation will retry indefinitely until success.

To define the FortiFlex token activation parameters:

1. Set the number of retries allowed:

   ```
   execute vm-license-options count 4
   ```

2. Set the retry interval:

   ```
   execute vm-license-options interval 5
   ```

3. Activate the license. The FortiFlex license token will be requested four times, with an interval of five seconds in between, as set.
   - If the license cannot be verified within the set amount of retries, the download will fail:

     ```
     # execute vm-license F4FC697D65428013FAKE

     This operation will reboot the system !
     Do you want to continue? (y/n)y

     Requesting FortiCare license token: *******, proxy:(null)
     Requesting FortiCare license token: *******, proxy:(null)
     ```

```
Requesting FortiCare license token: *******, proxy:(null)
Requesting FortiCare license token: *******, proxy:(null)
Failed to download VM license.
```

- If the license can be verified within the set number of retries, the VM license will be successfully installed:

```
# execute vm-license 227602862F7E6E9XXXX

This operation will reboot the system !
Do you want to continue? (y/n)y

Requesting FortiCare license token: *******, proxy:(null)
VM license install succeeded. Rebooting firewall.
```

FortiFlex token activation parameters can also be defined in an ISO file using the mime user-data.

To define the parameters in an ISO file:

1. Create a config drive ISO with a MIME file:

```
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="license.txt
"LICENSE-TOKEN: 334ADF7B49F2FEC1XXXX INTERVAL: 5 COUNT: 4
```

   See Cloud-init using config drive for more information.
2. Attach the ISO config drive at boot time. See Cloud-init for more information.
3. Boot up the VM and verify the token activation parameters:

```
# diagnose debug cloudinit show
 >> Found config drive /dev/sr0
 >> Successfully mount config drive
 >> MIME parsed preconfig script
 >> MIME parsed VM token
 >> MIME parsed config script
 >> Found metadata source: config drive
 >> Run preconfig script
 >> FortiGate-VM64  conf sys global
 …
 >> Trying to install vmlicense ...
 >> License-token:334ADF7B49F2FEC1XXXX INTERVAL:5 COUNT:4
 >> Run config script
```

Operational Technology

This section includes information about Operational Technology related new features:

System

This section includes information about system related Operational Technology new features:

Configuring the Purdue Level for discovered assets based on detected interface

This information is also available in the FortiOS 7.4 Administration Guide:

- Configuring the Purdue Level for discovered assets based on detected interface

The default Purdue Level can be set or unset in the CLI (`default-purdue-level`) within the system interface configuration. The default Purdue Level can be applied to discovered assets based on the interface with which they were detected. This feature requires a FortiGuard Industrial Security Service (ISS) license on the FortiGate so the Industrial Database (ISDB) can be used. Device identification must be enabled on interfaces connected to OT devices.

```
config system interface
    edit <name>
        set device-identification enable
        set default-purdue-level {1 | 1.5| 2 | 2.5| 3 | 3.5 | 4 | 5 | 5.5}
    next
end
```

By default, the `default-purdue-level` value is `3`. If the asset's Purdue Level is manually overridden, then it takes precedence over this default value set in the interface.

Example

In this example, the default Purdue Level on port1 is changed to 3.5. Subsequently, the Purdue Level of a detected device on port1 is manually changed to 4 on the *Asset Identity Center* page. After the manual change on the device, the Purdue Level remains at 4.

To configure the default Purdue Level:

1. Configure the interface settings:

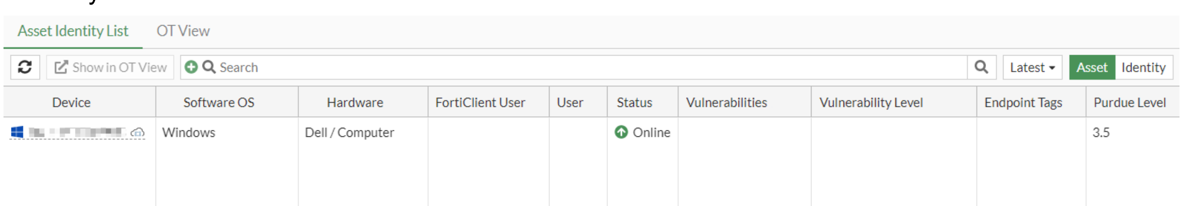
```
config system interface
    edit "port1"
```

```
        set device-identification enable
        set default-purdue-level 3.5
    next
end
```

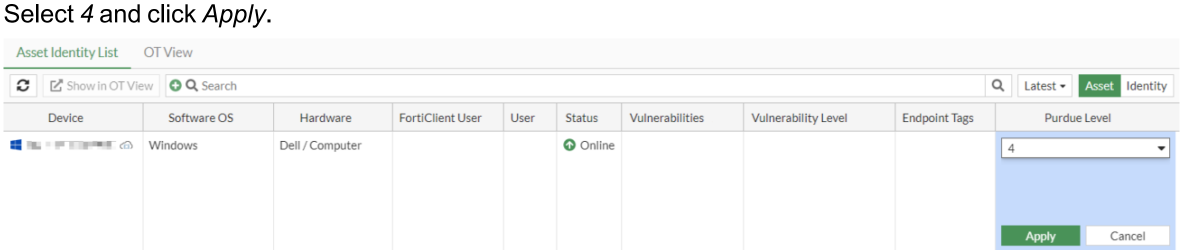
2. Verify that the Purdue Level as been updated in the user device store list:

```
# diagnose user-device-store device memory list

Record #1:

        device_info
                'ipv4_address' = '192.168.1.64'
                'mac' = '**:**:**:**:**:**'
                'hardware_vendor' = 'Dell'
                'hardware_type' = 'Home & Office'
                'hardware_family' = 'Computer'
                'vdom' = 'root'
                'os_name' = 'Windows'
                'os_version' = '10 / 2016'
                'last_seen' = '1680115135'
                'host_src' = 'mwbs'
                'unjoined_forticlient_endpoint' = 'false'
                'is_online' = 'true'
                'active_start_time' = '1680113976'
                'dhcp_lease_status' = 'leased'
                'dhcp_lease_expire' = '1680651757'
                'dhcp_lease_reserved' = 'false'
                'dhcp_server_id' = '2'
                'is_fortiguard_src' = 'true'
                'purdue_level' = '3.5'
                ...
```

3. Go to *Security Fabric* > *Asset Identity Center* and select the *Asset Identity List* tab. The device's *Purdue Level* is currently *3.5*.



4. Manually change the device's Purdue Level:
   a. Select the device and hover over the *Purdue Level* value.
   b. Click the pencil icon to edit the level.
   c. Select *4* and click *Apply*.



5. Verify that the Purdue Level as been updated in the user device store list:

```
# diagnose user-device-store device memory list

Record #1:

        device_info
                'ipv4_address' = '192.168.1.64'
                'mac' = '**:**:**:**:**:**'
                'hardware_vendor' = 'Dell'
                'hardware_type' = 'Home & Office'
                'hardware_family' = 'Computer'
                'vdom' = 'root'
                'os_name' = 'Windows'
                'os_version' = '10 / 2016'
                'last_seen' = '1680115467'
                'host_src' = 'mwbs'
                'unjoined_forticlient_endpoint' = 'false'
                'is_online' = 'true'
                'active_start_time' = '1680113976'
                'dhcp_lease_status' = 'leased'
                'dhcp_lease_expire' = '1680651757'
                'dhcp_lease_reserved' = 'false'
                'dhcp_server_id' = '2'
                'is_fortiguard_src' = 'true'
                'purdue_level' = '4'
                ...
```

Index

The following index provides a list of all new features added to FortiOS 7.4. The index allows you to quickly identify the version where the feature first became available in FortiOS.

Select a version number to navigate in the index to the new features available for that patch:

7.4.0

GUI

Network

SD-WAN

Policy and objects

Zero Trust Network Access

Security Profiles

VPN

User & Authentication

LAN Edge

System

Security Fabric

Cloud

Operational Technology

7.4.1

GUI

Network

SD-WAN

Policy and objects

Zero Trust Network Access

Security Profiles

VPN

User & Authentication

LAN Edge

System

Security Fabric

Log & Report

Cloud

7.4.2

GUI

Network

SD-WAN

Policy and objects

Zero Trust Network Access

Security Profiles

VPN

LAN Edge

System

Security Fabric

Log & Report

Cloud

www.fortinet.com