

WHITE PAPER

ADDRESSING PCI DSS WITH VMWARE NSX-T

A MICRO-AUDIT OF NSX-T MICRO-SEGMENTATION
FOR MICROSERVICES CONTAINERS AND VIRTUAL
MACHINES

COALFIRE CYBER ENGINEERING RESEARCH AND OPINION
VERSION 1.0

JASON MACALLISTER | PRINCIPAL AUTHOR
CHRIS KRUEGER | CISSP, PCI QSA

vmware®



C  A L F I R E .

North America | Europe

877.224.8077 | info@coalfire.com | [Coalfire.com](https://www.coalfire.com)

TABLE OF CONTENTS

Executive Summary	3
Coalfire Opinion	3
Introducing NSX-T, Micro-Segmentation, and PCI DSS 3.2	3
VMware NSX-T	4
NIST Special Publication 800-125B Recommendations	5
Definition of Micro-Segmentation.....	10
Alignment of NSX-T Micro-Segmentation with PCI-DSS Recommendations for Network Segmentation to Support Assessment Scope Reduction.....	11
Coalfire White Paper Objectives.....	11
NSX-T “Micro-Audit” on Effectiveness for PCI DSS Segmentation	12
Lab Characterization	12
Lab Setup.....	12
Setup NSX-T to Enable Security Controls	21
Segmentation Testing Methodology	26
Segmentation Testing Findings and Summary Results	26
Baseline Results	26
With Firewall Rules in Place and Applied to NSGroups	27
NSX-T Alignment with PCI DSS Requirements	28
Conclusion	44
Benchmark Test Summary	44
PCI DSS 3.2 Compliance Applicability Summary	45

EXECUTIVE SUMMARY

As more organizations move their businesses to the cloud, the way in which the cloud is utilized changes. The evolution of business systems from physical x86 servers to x86 virtualization, private cloud, public cloud, containerization, server less, and cloud native applications have required that organization's security practices keep stride to protect sensitive and proprietary information. This rapid evolution has not been without challenges over the years. Fortunately, and out of necessity, the topic of security has been increasingly on the forefront of new technological developments. Companies like VMware are making efforts to develop security solutions to keep up with the evolving landscape of service delivery.

One of the recent trends in service delivery is the increased use of the cloud and containerization for the delivery of applications. Cloud platforms provide immediately available resources to DevOps teams for rapid development and deployment of applications. At the same time, applications built in container platforms allow for the deployment of applications to virtually any cloud platform. Containers allow software to be developed, shipped, and deployed in standardized, isolated units. An application or application component can be packaged with only the necessary system libraries, code, runtime, system tools, and settings necessary to run the application or application component.

As a continuation of the 2016 benchmark on micro-segmentation, titled [VMware NSX Micro-Segmentation Cybersecurity Benchmark – A Micro Audit of NSX Threat Mitigation Effectiveness](#), and a 2017 benchmark on NSX in the DMZ titled [VMware NSX DMZ Anywhere – A Micro Audit of NSX DMZ Anywhere](#), VMware requested that Coalfire perform a benchmark of the efficacy of VMware NSX-T micro-segmentation capabilities as applicable to the Payment Card Industry Data Security Standard 3.2 (PCI DSS 3.2). In this new benchmark Coalfire consulted and performed testing to form an opinion of the effectiveness of NSX-T micro-segmentation for securing both virtual machines (VMs) and “containers” orchestrated by Kubernetes (K8s) on vSphere hypervisors. While PCI DSS does not specifically address the use of containerization in cardholder data environments (CDE), the alignment of requirements and recommendations focuses specifically on capabilities of NSX-T to provide network segmentation for isolation of CDE for the protection of cardholder data (CHD), per PCI DSS recommendations and best practices for assessment scope reduction. Coalfire also evaluated the applicability of the NSX-T Distributed Firewall (DFW) for PCI DSS 3.2 Requirement 1 technical requirements.

COALFIRE OPINION

To conclude, Coalfire found that the NSX-T distributed firewall, combined with the container logical switch and container logical router, was capable of micro-segmentation of pods within the container environment. This micro-segmentation capability, like that found with NSX for vSphere with VMs, is sufficient for providing recommended network segmentation for scope reduction where CDE VMs, pods, and containers along with their associated transport zones can be segmented from non-CDE VMs, pods, and containers. Additionally, the micro-segmentation capabilities of NSX-T were effective for providing more granular security control in support of a Zero Trust network model for assets within the CDE. Moreover, the NSX-T distributed firewall could support PCI DSS 3.2 firewall requirements for CDE.

INTRODUCING NSX-T, MICRO-SEGMENTATION, AND PCI DSS 3.2

VMware introduced micro-segmentation with NSX for vSphere. Coalfire tested the completeness and efficacy of NSX micro-segmentation capabilities in a 2016 benchmark whitepaper. In that benchmark, Coalfire looked at recommendations provided by National Institute of Standards and Technology (NIST) Special Publication 800-125B pertaining to segmentation and firewalls in virtualized environments and evaluated the completeness of the NSX for vSphere solution to align with those recommendations. Along

with this alignment, Coalfire evaluated the granularity of micro-segmentation capable of being achieved by NSX for vSphere.

VMware has expanded capabilities of NSX through the introduction of NSX-T. NSX-T provides similar software defined network constructs as NSX for vSphere to support networking and security capabilities to heterogeneous platforms beyond the vSphere platform. This includes support for mixed platforms that may include a combination of multiple hypervisors, bare-metal, cloud service, VMs, containers, and such. The intent is to provide the same granular control of the network and security as applicable to disparate workloads. This allows for uniformity in management and control of networking and security for a broader application. Included in this capability of control is the benefit of micro-segmentation to provide segmentation down to each VM, container, or pod.

VMWARE NSX-T

VMware NSX-T was designed to provide seamless network virtualization and network security for a multi-cloud and multi-hypervisor environment. The familiar NSX technology can be used for both cloud (private and public) and container environments. NSX-T has integration capabilities for vSphere and Kernel-based Virtual Machine (KVM) hypervisors as well as integration capability for any container management platform. This provides organizations with greater uniformity in support for networking and security to extend to a wider variety of platforms where workloads may reside. Furthermore, it supports greater flexibility for organizations and DevOps decisions for service delivery methods. For more technical details on NSX-T, see [VMware-NSX-T-Technical-White-Paper-20170202-v1.0.pdf](#). Figure 1 displays a high-level overview of the NSX-T architecture. The components include a management plane, a control plane, and a distributed data plane. This architecture is consistent across all platforms where NSX-T is integrated.

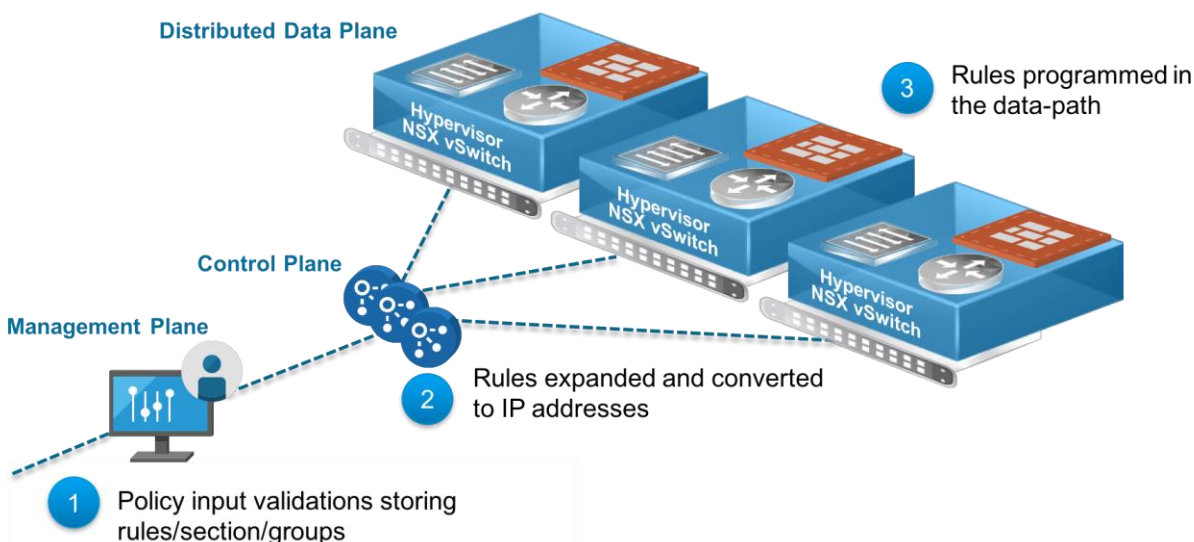


Figure 1: NSX-T Architecture and Components

NSX-T provides rich networking functionality like that found with NSX for vSphere including logical switching, distributed logical routing, distributed firewalling, and network services such as NAT, DHCP relay/server, and metadata proxies.

This paper focuses on the capabilities of NSX to provide segmentation for East/West intra-network as well as North/South inter-network communications. The capabilities to support segmentation through NSX-T distributed logical router and distributed firewall (DFW) in coordination with the VMware NSX-T vSwitch are

the basis for determining the usefulness of NSX-T to meet segmentation recommendations for PCI DSS 3.2 found in this white paper.

NSX-T introduces support for DFW functionality for workloads running on both ESXi and KVM hosts/hypervisors. The NSX DFW provides the capability to enforce firewalling functionality directly at the workload virtual network interface card (vNIC), including the container vNIC. Micro-segmentation policies provided by NSX-T can be based on IP address sets, MAC identifier sets, logical switch residency, logical ports, or advanced security policies based on security groups.

The combination of capabilities to support heterogeneous environments and enable micro-segmentation makes NSX-T an ideal candidate for use in PCI DSS 3.2 regulated environments.

NIST SPECIAL PUBLICATION 800-125B RECOMMENDATIONS

To understand the capabilities of a product or solution, it is useful to align that solution to a commonly adopted and supported standard and determine the capability of that product or solution to meet the standard. As NSX-T is designed to address networking and security capabilities for emerging technologies, it is useful to align the capabilities of NSX-T with NIST Special Publication 800-125B recommendations for virtual infrastructures and to build upon this premise for micro-segmentation capabilities of NSX-T for supporting PCI-DSS segmentation recommendations and providing broader support for PCI DSS 3.2 Requirement 1.

Emerging cybersecurity standards, such as those being developed by the NIST (the US federal technology agency responsible for applied standards for technology and measurement), are contributing to an emerging global consensus on Information Security, particularly about virtualized infrastructures. In NIST Special Publication 800-125B, titled [Secure Virtual Network Configuration for Virtual Machine \(VM\) Protection](#), the Institute makes four recommendations for securing virtualized workloads, found in Section 4.4 of their guidance:

VM-FW-R1: *In virtualized environments with VMs running delay-sensitive applications, virtual firewalls should be deployed for traffic flow control instead of physical firewalls, because in the latter case, there is latency involved in routing the virtual network traffic outside the virtualized host and back into the virtual network.*

VM-FW-R2: *In virtualized environments with VMs running I/O intensive applications, kernel-based virtual firewalls should be deployed instead of subnet-level virtual firewalls, since kernel-based virtual firewalls perform packet processing in the kernel of the hypervisor at native hardware speeds.*

VM-FW-R3: *For both subnet-level and kernel-based virtual firewalls, it is preferable if the firewall is integrated with a virtualization management platform rather than being accessible only through a standalone console. The former will enable easier provisioning of uniform firewall rules to multiple firewall instances, thus reducing the chances of configuration errors.*

VM-FW-R4: *For both subnet-level and kernel-based virtual firewalls, it is preferable that the firewall supports rules using higher-level components or abstractions (e.g., security group) in addition to the basic 5-tuple (source/destination IP address, source/destination ports, protocol).*

With NSX-T virtual networking, where the source and destination are within the virtual environment, network traffic is contained within the environment. Rather than requiring traffic to be directed and hairpinned to firewalls outside of the virtual environment, the NSX-T firewall is virtually distributed, implemented in the kernel, and attached to the vNIC of each workload. This allows the security rules of the distributed firewall to be attached directly to the network source and/or destination device. Data plane forwarding and transformation decisions are made based on local tables populated by the NSX-T control plane.

NSX-T can be deployed using ESXi or KVM based hypervisors. The capability of NSX-T to provide distributed firewall in support of network micro-segmentation for both containers and VMs is similar in either case; only the integration for each type of hypervisor slightly differs.

NSX-T DFW rules are enforced at the kernel level. For both ESXi and KVM deployments, the local control plane (LCP) components are pushed to the hypervisor. The NSX Manager triggers the installation of the LCP components to the hypervisors. In both cases, the DFW configuration is received from the central control plane (CCP). Figure 2 illustrates the flow from the Management Plane (MP) to the CCP and then to the Data Plane. In the dataplane the transport nodes on the left represent ESXi hosts; whereas, the right most transport node represents a KVM host.

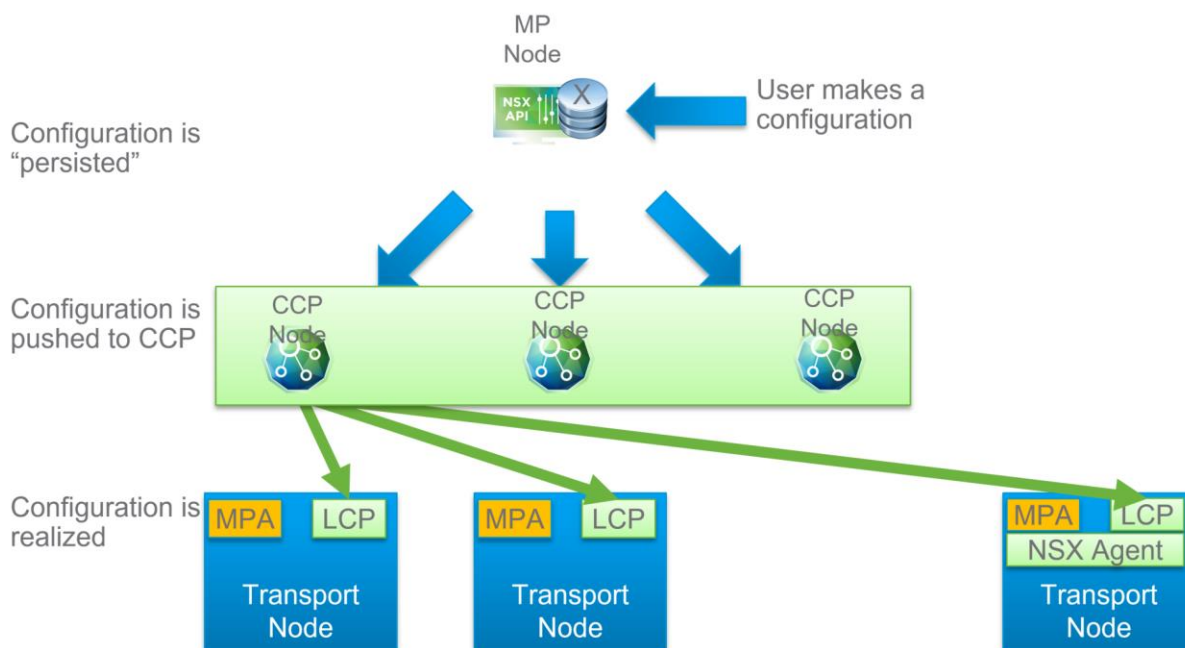


Figure 2: NSX-T DFW - Management Plane, Control Plane, and Data Plane Roles

For KVM deployments, the NSX agent has a DFW wiring module as a component. It's used to generate Openflow flows on the Open Virtual Switch (OVS) based NSX-T vSwitch for firewall rules pushed from the CCP. Stateless filtering is implemented through the ovs-daemon, which is part of OVS distributions. It employs the wiring implementation it received from LCP in the form of Openflows. The Linux conntrack utilities are used to keep track of the state of connections in case they were allowed by a stateful firewall rule. Any new packet is first looked up in conntrack to see if there is an existing connection.

Network statistics are exported through the Management Plane Agent (MPA) directly to the NSX Manager.

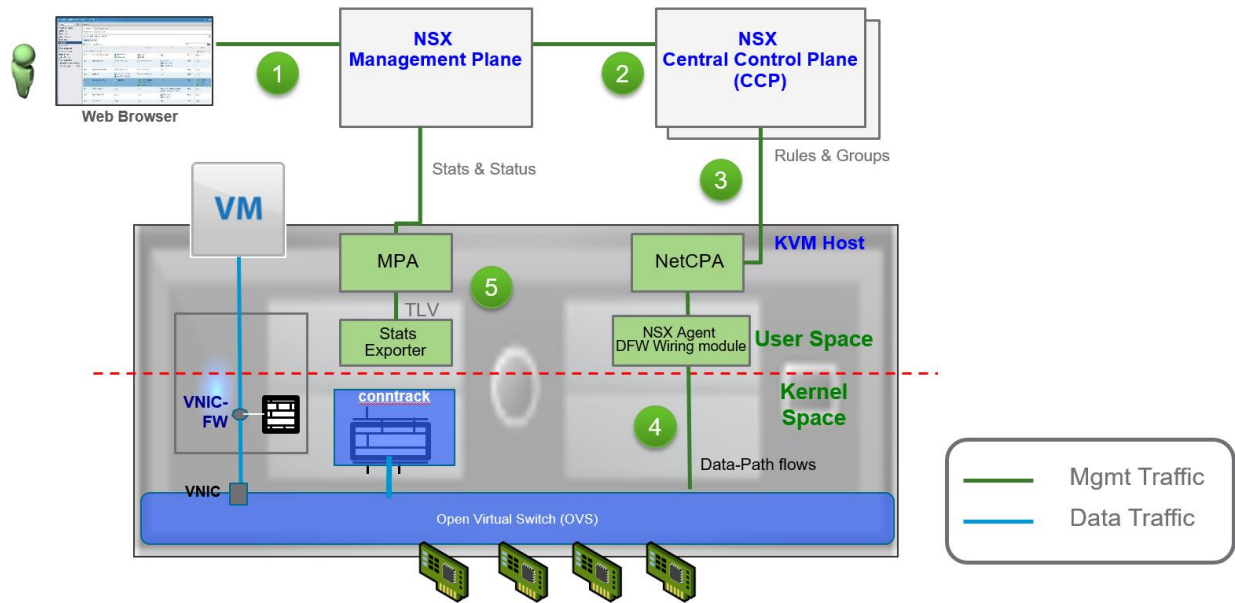


Figure 3: NSX-T DFW: KVM Architecture

For ESXi hypervisor deployments, the integration is directly with the NSX-T vSwitch. The kernel space implementation for ESXi hypervisors is the same as that with NSX-v. However, the distributed virtual switch is not controlled by vCenter in NSX-T deployments.

L3 networking on ESXi uses the distributed logical router similar to that with NSX-v, whereas NSX-T on KVM uses OVS flows to implement routing.

The NSX Manager is decoupled from vCenter and designed to run across heterogeneous platforms. The NSX-T DFW is integrated with the virtualization management platform for both vSphere and KVM deployments. Through the NSX-T management console, firewall rules can be uniformly deployed to the distributed firewall whether deployed to VM instances or to container instances.

In addition to the basic 5-tuple characteristics of firewall rules, NSX-T provides support for application of dynamic rules to security groups. Security groups can be comprised of VMs and containers or pods and defined by various criteria including security tag, machine name, subnet, virtual switch, and so forth. This allows for membership to security groups to be dynamic whereby the application of the distributed firewall can be automated.

These capabilities to enable this level of security control to VMs across heterogeneous platforms is also extended to containerized workloads. The integration of NSX-T with container platforms allows for improvements over native container networking. With NSX-T, a single network fabric can be used to connect VMs, network services, and containers across on-premise and public cloud deployments. The container network integrates with the rest of the data center network with Border Gateway Protocol (BGP), allowing for seamless routing control across the data center. NSX-T supports Layer 3 reachability between load balancers, firewalls, and containers, simplifying the integration of network services.

The NSX-T DFW can be applied with granularity down to individual containers in the integrated container platform. This enables microservices to be secured per container with firewall enforcement and network monitoring. Security administrators can enable policy for securing communication between microservices in the container platform as well as between the microservice and the database. Network policies can be defined per microservice requirement and applied dynamically to microservices based on security tags or

other metadata. Figure 4 illustrates conceptually the granularity of control between microservices as well as to the larger data center including VMs.

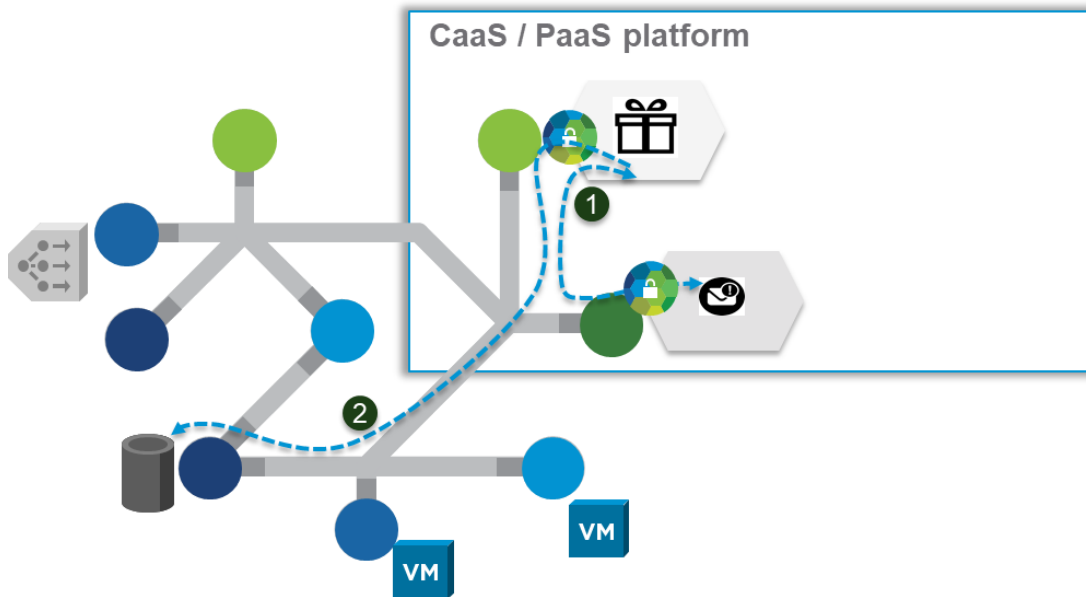


Figure 4: Microsegmentation for Microservices

The integration of NSX-T with container platforms provides improved isolation capabilities in multi-tenant environments. Each tenant can be segmented with its own dedicated routers and network and firewall instances. With NSX-T isolation can occur at the network layer, rather than at the physical layer where the platform provider would build a separate container cluster per tenant. Figure 5 depicts the isolation of tenants in a multi-tenant container platform deployment with NSX-T providing separation of tenants through network controls.

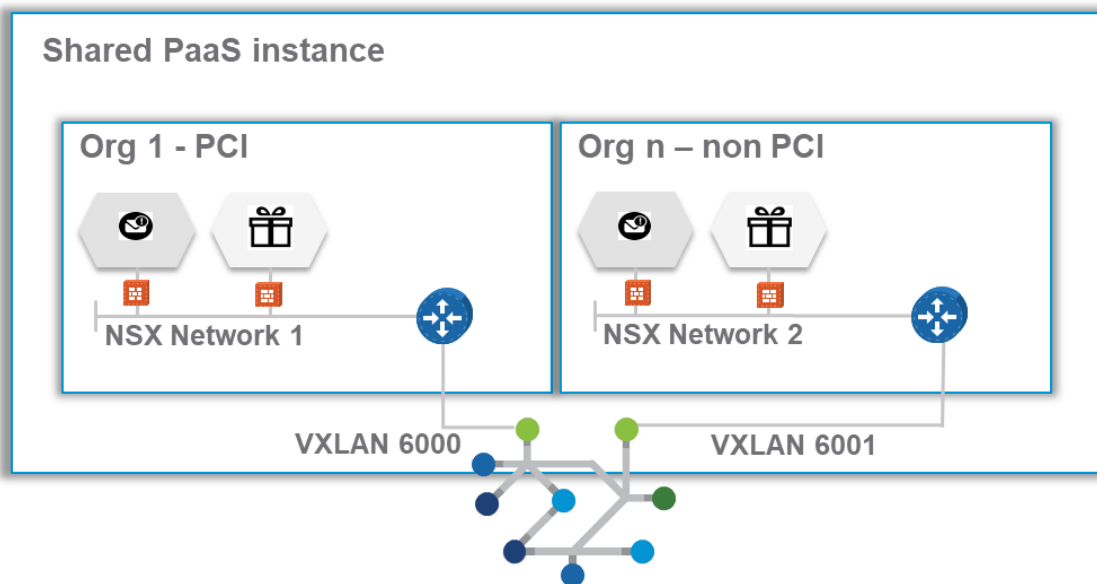


Figure 5: Tenant Isolation with NSX-T for Container Platforms

NSX-T provides the additional benefit of enabling a consistent management experience across platforms by utilizing one operational set of tools for both VMs and containers. This common set of tools includes transmit/receive (TX/RX) counters per container, IP flow information export (IPFIX) for traffic flow records, Switched Port Analyzer (SPAN) for redirection or copying traffic to a monitoring device, and Traceflow for tracing network and host failures.

As part of this assessment, Coalfire examined K8s NSX-T integration where the K8s master and cluster nodes were running as VMs on a vSphere cluster. For this type of deployment, NSX-T extends the NSX-T vSwitch into the K8s node VM using VLAN tagging. The node VM vSwitch is standalone and only gets programmed by the NSX-T container networking interface (CNI) plugin.

The NSX-T Network Container Plugin (NCP) watches the K8s API for any events such as namespace or pod changes or additions. Changes in the K8s environment prompts the creation of a network topology to support the new namespace and pods. This includes the creation of container logical routers and switches and assignment of ports on each to facilitate connection of the pods to the network. Each pod is assigned a container interface (CIF). The CIF is where the NSX-T DFW is attached and the firewall policy can be applied. Figure 6 depicts the topology of the NSX-T container network. This illustration shows the communication path between pods in the K8s environment. All traffic into and out of the pod traverses the NSX-T DFW.

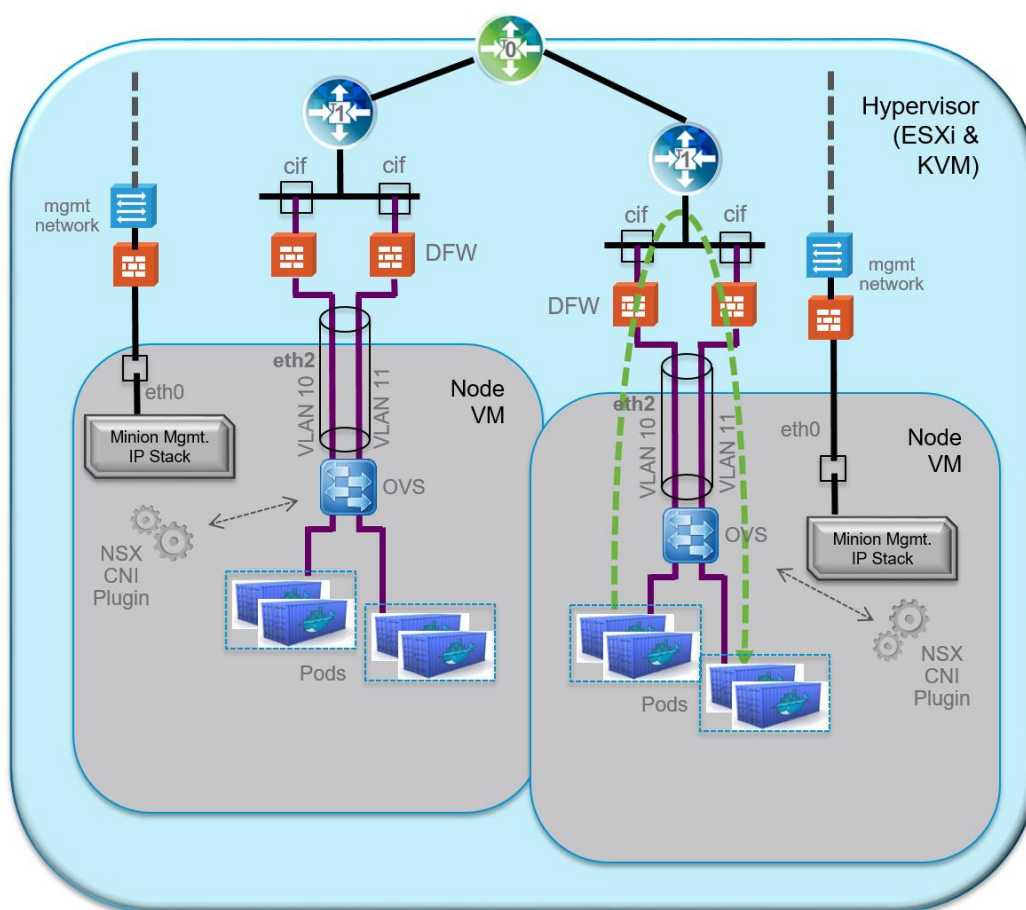


Figure 6: NSX-T Container Interface

Definition of Micro-Segmentation

According to VMware, “Micro-segmentation enables organizations to logically divide its data center into distinct security segments down to the individual workload level, and then define security controls and deliver services for each unique segment.” (Lawrence Miller, CISSP and Joshua Soto, 2015, p. 21) The benefit of micro-segmentation is that it denies an attacker the opportunity to pivot laterally within the internal network, even after the perimeter has been breached.

VMware NSX-T supports micro-segmentation as it allows for a centrally controlled, yet distributed firewall to be attached directly to workloads within an organization’s network. The distribution of the firewall for the application of security policy to protect individual workloads is effective as rules can be applied that are specific to the requirements of each workload. The additional value that NSX-T provides is that the capabilities of NSX are not limited to homogenous vSphere environments, but support the heterogeneity of platforms and infrastructure that is more commonly used with many organizations today. Figure 7 depicts micro-segmentation capabilities of NSX, where each workload is virtual secured with its own distributed firewall.

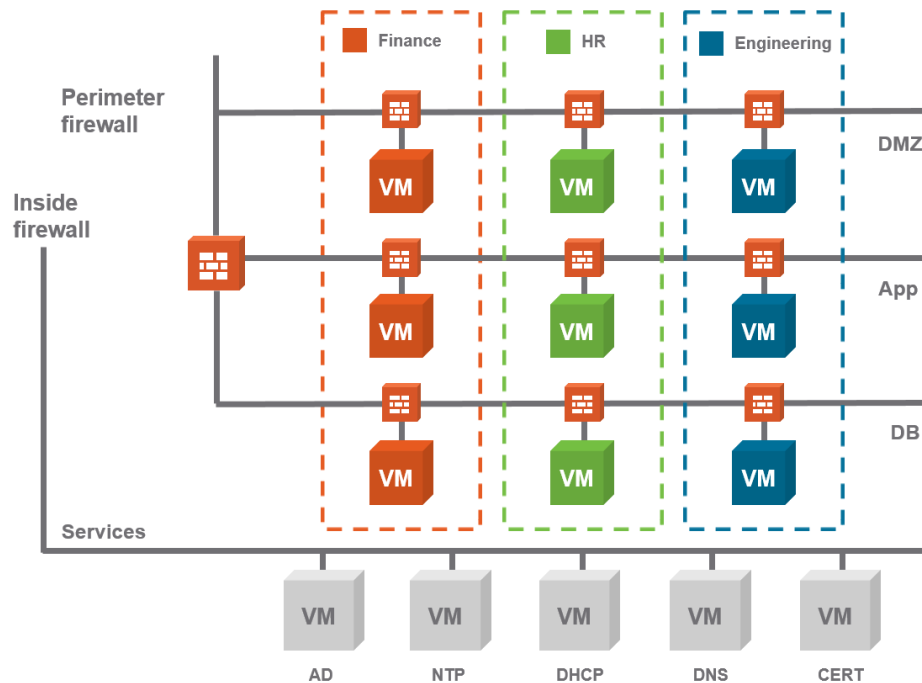


Figure 7: Example of Micro-Segmentation with NSX

Micro-segmentation provided by NSX-T better supports a Zero Trust architecture for IT security such that it allows for perimeters to be established around each workload. The Zero Trust architecture was introduced by analyst firm Forrester Research as an alternative approach to IT security architecture. Conventional security models assume that everything on the inside of an organization’s network can be trusted, whereas the Zero Trust model assumes the opposite: that nothing can be trusted and everything should be verified. The Zero Trust model for IT security is a principle that addresses the increased sophistication of network attacks and insider threats. Rather than simply placing firewalls at the edge of the organization’s network to prevent attacks from external networks, the Zero Trust model looks at ways to better control and manage network traffic within the organization’s network. The intent is that for each system in an organization’s network, trust of the underlying network is completely removed. To do this, organizations can define perimeters within the network to limit the possibility of lateral (east-west)

movement of an attacker. Implementation of a Zero Trust model of IT security with traditional network security solutions designed primarily to protect the organization's edge can be costly and complex. Moreover, the lack of visibility for organization's internal networks can slow down implementation of a Zero Trust architecture and possibly leave gaps that may only be discovered during a breach. Additionally, internal perimeters may only have granularity down to a VLAN or subnet, as is common with many traditional DMZs. However, network virtualization solutions like NSX and NSX-T can provide a more cost effective and efficient means to implement a Zero Trust network.

Alignment of NSX-T Micro-Segmentation with PCI-DSS Recommendations for Network Segmentation to Support Assessment Scope Reduction

PCI DSS, currently at version 3.2, is a proprietary information security standard that was created with the intent to reduce credit card fraud by stipulating a series of required controls regulating the use of information systems that handle CHD and sensitive authentication data (SAD). PCI DSS is not an optional standard. As stated, all entities who process, store, or transmit CHD and/or SAD must comply with the standard or they can be fined or refused access to the card brand's payment system (PCI SSC, 2016).

PCI DSS 3.2 security standards currently include a series of recommendations as well as twelve requirement topics across six security categories. This paper focuses on the relevant to NSX-T recommendations and requirements. One such recommendation from PCI DSS 3.2 is regarding network segmentation. Though not a requirement, PCI DSS strongly recommends the use of network segmentation to isolate CDE from non-CDE. The rationale for the recommendation is to reduce the scope of a PCI DSS assessment to only include the isolated CDE and peripheral systems. Reduction in assessment scope could likely reduce the cost of PCI DSS assessments. Furthermore, utilizing network segmentation should reduce the cost and difficulty of implementing PCI DSS controls as the efforts and program can be narrowed down on systems requiring compliance. Most importantly, the use of network segmentation to protect CDE should reduce the risk to the organization "by consolidating cardholder data into fewer, more controlled locations." (PCI SSC, 2016)

It is important to understand the micro-segmentation capabilities of NSX-T and how they can support the recommended segmentation and therefore support the outlined benefits of segmentation as presented by PCI DSS. A key objective of this project was to test the efficacy of NSX-T to provided perimeter protections around key workloads to the extent that workloads outside of the perimeter are unable to access the protected workload.

Finally, if NSX-T is used to support network segmentation, it is important to understand how NSX-T aligns with relevant PCI DSS technical requirements. In this paper, Coalfire aligns the technical purpose of NSX-T with key PCI DSS requirements. Primary among these are:

- Requirement 1: Install and maintain a firewall configuration to protect cardholder data
- Requirement 10: Track and monitor all access to network resources and cardholder data

Additional requirements may be applicable as part of best practices given that NSX-T will be either directly or indirectly part of the CDE and therefore in scope for assessment. Coalfire will outline the evaluation of NSX-T as it pertains to compliance with those relevant PCI DSS requirements as well.

COALFIRE WHITE PAPER OBJECTIVES

The following were the key objectives of this white paper:

Perform a Micro-Audit to Demonstrate the Extent of Capability of NSX-T to Support Segmentation

To narrow the scope of testing, VMware provided Coalfire the following set of parameters for testing:

1. NSX-T provides PCI applicable segmentation for inter-microservice traffic within a K8s namespace.
2. NSX-T provides PCI applicable segmentation for inter-microservice traffic between PODs of a microservice.
3. NSX-T provides PCI applicable segmentation for microservice traffic between two K8s namespaces.
4. NSX-T provides PCI applicable segmentation and network control between K8s Apps (PODs) and VMs (e.g. DB VM).
5. Security policy within NSX-T can be specified using K8s labels and NSX.

Provide a Matrix of Alignment for NSX-T with PCI DSS 3.2 Requirements

Ultimately, this paper demonstrates the suitability of NSX-T to enable security initiatives in support of PCI DSS 3.2 compliant workloads.

NSX-T “MICRO-AUDIT” ON EFFECTIVENESS FOR PCI DSS SEGMENTATION

LAB CHARACTERIZATION

To facilitate testing, VMware, with consultation from Coalfire, built out a lab environment on a vSphere cluster of ESXi hypervisors managed by vCenter. On this platform, a K8s cluster was created made up of a single K8s master with two K8s nodes. VMware NSX-T is integrated into this K8s cluster to present a networking and networking security layer to provide instructions for switching, routing, and control of communication within the container environment. Like NSX for vSphere, NSX-T makes use of familiar networking constructs such as logical switches, logical routers, and distributed firewalls for controlling and isolating communication between services residing in the K8s environment. This same capability extends to the VM side of the virtual data center. For instance, persistent database servers may run as VMs. These database servers may be utilized by services running as containers in the data center. In summary, NSX-T presents a single network fabric that connects VMs, network services, and containers across an organization’s on-premise and public cloud environments.

In support of this testing, a few simulated workloads were deployed on a K8s infrastructure. The setup of the lab was based loosely on a VMware lab training exercise introducing K8s with NSX-T. The workloads were deployed as pods and representative of a non-CDE and a CDE. The CDE was comprised of a web service, an application service, an intermediary database service, and a database tier, where the database tier was made up of a VM hosted on vSphere ESXi cluster.

Lab Setup

The underlying platform was made up of a vSphere cluster controlled by vCenter. NSX-T also supports deployment and integration with a KVM cluster. Running on VMware VMs was a K8s cluster of two nodes with a single master running on a separate VM. The following process outlines the steps used to create the workload K8s pods. The lab as presented to Coalfire and included integration of VMware NSX-T with the K8s cluster. As such, the NSX-T NCP was running as a container image as a K8s pod in the K8s cluster with the K8s adapter.

Create Namespaces

NCP creates a watch on the K8s API for any namespace events. When a user creates a new K8s namespace, the K8s API server notifies NCP of any changes or additions of namespaces. Subsequently, NCP creates the network topology for the namespace inclusive of request for a new subnet from the pre-

configured NSX-T IP block, creation of a logical switch, and creation of a tier 1 (T1) logical router and attaches the new T1 router to the global tier 0 (T0) router. Finally, a router port is created on the T1 router, which allows for the attachment of the logical switch. The port is assigned an IP address from the new subnet.

Namespaces, according to K8s, help different projects, teams, or customers to share a K8s cluster. Typically, namespaces are used to separate either organizational functional groups or, in multi-tenant environments, provide separation for tenants. In this lab setup, Coalfire created two separate namespaces: one for CDE, named “nsx-cde”, and one for non-CDE, named “nsx-noncde”. For clarification, the “nsx-cde” namespace represents the CDE where pods were setup to simulate different functional aspects of a microservices deployment where each function can be secured with its own DFW. The “nsx-noncde” namespace represented all the non-CDE on the K8s cluster. Prior to creation of the workload namespaces, Figure 8 shows the existing namespaces from the K8s’s perspective.

```
root@k8s-master:~# kubectl get ns
NAME                STATUS    AGE
default             Active    5d
kube-public         Active    5d
kube-system         Active    5d
```

Figure 8: K8s Namespaces Prior to Workload Namespace Creation

Prior to the setup of the new namespaces to represent the workloads, Figure 9 represents the logical switches in NSX-T. The named logical switches and routers match that of the existing namespaces.

SWITCHES		
Logical Switch	ID	
k8s-cl1-default-0	8da1...302b	
k8s-cl1-kube-public-0	505d...098a	
k8s-cl1-kube-system-0	b88b...27f2	
k8s-mgmt	b1ac...198b	
k8s-node-vifs	7da5...6c70	
uplink-to-lab	8fd7...8197	

Figure 9: Pre-Workload Pod Setup NSX-T Switches Configuration

Figure 10 shows the default set of logical routers prior to the creation of the functional workload namespaces. The next steps illustrate the process for the creation of new namespaces with the ensuing creation of logical network devices by NSX-T. Note that there is a container logical switch and a container logical router for each K8s namespace.

ROUTERS		NAT	
+ ADD ▾ EDIT DELETE ACTIONS ▾			
<input type="checkbox"/> Logical Router ↑	ID	Type	
<input type="checkbox"/> Central-TO-Router	d4ea...011c	Tier-0	
<input type="checkbox"/> k8s-cl1-default	7540...7e85	Tier-1	
<input type="checkbox"/> k8s-cl1-kube-public	ed45...62e2	Tier-1	
<input type="checkbox"/> k8s-cl1-kube-system	e2c9...3f32	Tier-1	

Figure 10: Pre-Workload Pod Setup NSX-T Containers Configuration

A K8s namespace was created for each CDE and non-CDE deployment. As the namespaces are created from the K8s Master, Figure 11 shows the listed namespaces including the new “nsx-cde” and “nsx-noncde” namespaces.

```

root@k8s-master:~# kubectl get ns
NAME                STATUS    AGE
default             Active    5d
kube-public         Active    5d
kube-system         Active    5d
nsx-cde             Active    12s
nsx-noncde          Active    12m

```

Figure 11: List of K8s Namespaces

Figure 12 shows the addition of a logical switch for the new “nsx-cde” and “nsx-noncde” namespaces, which are labeled “k8s-cl1-nsx-cde-0” and “k8s-cl1-nsx-noncde-0” respectively.

SWITCHES		PORTS		SWITCHING PROFILES	
+ ADD EDIT DELETE ACTIONS ▾					
<input type="checkbox"/> Logical Switch ↑	ID				
<input type="checkbox"/> k8s-cl1-default-0	8da1...302b				
<input type="checkbox"/> k8s-cl1-kube-public-0	505d...098a				
<input type="checkbox"/> k8s-cl1-kube-system-0	b88b...27f2				
<input type="checkbox"/> k8s-cl1-nsx-cde-0	5bbd...96e1				
<input type="checkbox"/> k8s-cl1-nsx-noncde-0	18ca...2b6a				
<input type="checkbox"/> k8s-mgmt	b1ac...198b				
<input type="checkbox"/> k8s-node-vifs	7da5...6c70				
<input type="checkbox"/> uplink-to-lab	8fd7...8197				

Figure 12: Logical Switches After New Namespace Creation

Each container logical switch is assigned a unique subnet. This is assigned from a pre-configured in VMware NSX-T available IP block. Figure 13 shows details of the container logical switch for the “nsx-cde” namespace and the subnet that was assigned: 10.4.0.160/27. IP addresses are dynamically assigned on the container logical switch to the pods as they are created. The switch ports are ephemeral and exist only as long the containers that are attached are in existence.

k8s-cl1-nsx-cde-0			
Overview	Monitor	Manage	Related
Summary EDIT			
Name	k8s-cl1-nsx-cde-0		
ID	5bbd7e99-da96-441d-98e2-35c9c0e496e1		
Description			
Admin Status	● Up		
Replication Mode	Hierarchical Two-Tier replication		
VNI	54171		
Logical Ports	1		
Traffic Type	Overlay		
Transport Zone	tz1		
Created	8/8/2017, 12:43:07 PM by admin		
Last Updated	8/8/2017, 12:43:07 PM by admin		
Subnets			
IP Ranges	Gateway	CIDR	
10.4.0.162 - 10.4.0.190		10.4.0.160/27	

Figure 13: Example Container Logical Switch Details

Each namespace is also assigned a container logical router. This determines how traffic is routed inbound and outbound to the central T0 router and how the traffic is routed between namespaces. Figure 14 shows the listed container logical routers after the creation of the new “nsx-cde” and “nsx-noncde” namespaces. The routers are named “k8s-cl1-nsx-cde” and “k8s-cl1-nsx-noncde” respectively. The names of the logical routers and logical switches are constructed based on the namespace label and are easily identified within the NSX-T Manager.

ROUTERS		NAT	
+ ADD EDIT DELETE ACTIONS			
Logical Router	ID	Type	
<input type="checkbox"/> Central-T0-Router	d4ea...011c	Tier-0	
<input type="checkbox"/> k8s-cl1-default	7540...7e85	Tier-1	
<input type="checkbox"/> k8s-cl1-kube-public	ed45...62e2	Tier-1	
<input type="checkbox"/> k8s-cl1-kube-system	e2c9...3f32	Tier-1	
<input type="checkbox"/> k8s-cl1-nsx-cde	c4a9...db35	Tier-1	
<input type="checkbox"/> k8s-cl1-nsx-noncde	21d4...9afc	Tier-1	

Figure 14: Container Logical Router After Creation of cde and noncde Namespaces

Create Pods to Represent Workloads for Testing

To deploy pods in this environment, Coalfire used VMware-provided templates and container images. The container image was made up of a simple web service, as will be shown later. As this exercise was primarily concerned with testing the boundary protection capabilities of NSX-T, the function of the container was not of chief concern. Figure 15 shows an example of the template (.yaml) file that was used to generate the nsx-cdweb pods in the K8s environment. This template is for the deployment of a replication controller with the replication of four pods in the environment. The pods have been given labels to identify the application that the pods represent and a security tag. Port 80 is the container port used to communicate with the container service.

```

apiVersion: v1
kind: ReplicationController
metadata:
  name: nsx-cdeweb-rc
  labels:
    app: nsx-cdeweb
spec:
  replicas: 4
  template:
    metadata:
      labels:
        app: nsx-cdeweb
        secgroup: cdeweb-tier
    spec:
      containers:
      - name: nsx-demo
        image: yfauser/nsx-demo
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 80

```

Figure 15: Pod Template Used for “nsx-cdeweb” Pods

Pods were deployed using the template as depicted in Figure 16.

```

localadmin@k8s-master:~/demos$ kubectl create -f nsx-demo-controller-secgroup.yaml
replicationcontroller "nsx-cdeweb-rc" created

```

Figure 16: Creation of Replication Controller nsx-cdeweb-rc and Initial Pods.

After the replication controller and pod creation, checks were made to ensure that they started correctly and were running. As part of confirmation of capabilities, Coalfire verified that the environment and the services functioned as expected. Figure 17 shows the running pods, each with a unique identifier and the replication controller used for deployment of the pods.

```

Every 2.0s: kubectl get all -o wide                                     Wed Aug  9 09:35:59 2017

```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
po/mgmtpod	1/1	Running	0	3h	10.4.0.166	k8s-node2
po/nsx-cdeweb-rc-4s1c5	1/1	Running	0	1m	10.4.0.163	k8s-node1
po/nsx-cdeweb-rc-7fnc9	1/1	Running	0	1m	10.4.0.162	k8s-node2
po/nsx-cdeweb-rc-k411g	1/1	Running	0	1m	10.4.0.164	k8s-node1
po/nsx-cdeweb-rc-xk6bc	1/1	Running	0	1m	10.4.0.165	k8s-node2

NAME	DESIRED	CURRENT	READY	AGE	CONTAINER(S)	IMAGE(S)	SELECTOR
rc/nsx-cdeweb-rc	4	4	4	1m	nsx-demo	yfauser/nsx-demo	app=nsx-cdeweb,secgroup=cdeweb-tier

Figure 17: nsx-cdeweb Pods and Replication Controller

Figure 18 shows port details in NSX-T Manager for one of the deployed nsx-cdeweb pods. Included in this detail is the secgroup tag that was set during the creation as part of the deployment template. The secgroup label is tagged with cdeweb-tier. This matches the value that was set in the template. This will be important for setting up NSX security groups within NSX and for application of firewall rules.

k8s-cl1-nsx-cdeweb-rc-4s1c5

Overview Monitor Manage Related

Summary EDIT

Name k8s-cl1-nsx-cdeweb-rc-4s1c5
 ID 240e8b06-440f-483f-9190-17f60ff40ba1
 Description
 Admin Status ● Up
 Attachment VIF:8f5897e0-93eb-4919-99a9-4da33b61a57d
 Logical Switch k8s-cl1-nsx-cde-0
 Created 8/9/2017, 9:34:40 AM by admin
 Last Updated 8/9/2017, 9:34:40 AM by admin

Address Bindings REFRESH

Tags MANAGE

Scope	Tag
ncp/project	nsx-cde
ncp/version	0.0.1.dev541
ncp/cluster	k8s-cl1
ncp/pod	nsx-cdeweb-rc-4s1c5
ncp/ing_ctrl	False
app	nsx-cdeweb
secgroup	cdeweb-tier

Figure 18: Example Port Details for Deployed Pod

The steps for creating pods to represent additional service tiers within the “nsx-cde” namespace are repeated with slight nuances, including the defined tag for the secgroup label. Similarly, a web service was created in the “nsx-noncde” namespace.

Create Load Balancing Service

A load balancing service was setup to balance the load among the available pods. This allowed for any of the pods to respond to requests. Figure 19 shows the detail of the configuration yaml for creation of the load balancing service.

```
apiVersion: v1
kind: Service
metadata:
  name: nsx-cdeweb-svc
  labels:
    app: nsx-cdeweb
spec:
  ports:
    # the port that this service should serve on
    - port: 80
  selector:
    app: nsx-cdeweb
```

Figure 19: Load Balance Service Configuration Template

The next step was to create the load balancing service from the available yaml, as depicted in Figure 20.

```
localadmin@k8s-master:~/demos$ kubectl create -f nsx-demo-service.yaml
service "nsx-cdeweb-svc" created
```

Figure 20: Load Balance Service Creation

The next step was to check the status of the load balancing service as created. A cluster IP address is assigned to the load balancing service. The service allows for redirection to the correct pods within the K8s cluster. The selector determines the app that the load balancing service will support. Figure 21 depicts the load balancing service status.

```
localadmin@k8s-master:~/demos$ kubectl get svc -o wide
NAME                CLUSTER-IP      EXTERNAL-IP      PORT(S)    AGE      SELECTOR
nsx-cdeweb-svc     10.107.102.32   <none>           80/TCP     52s     app=nsx-cdeweb
```

Figure 21: Load Balancing Service Status

Running the describe command, as shown in Figure 22, against the service name shows some details about the load balancing service including: service name, namespace where the service resides, selector for the service, and assigned cluster IP address. Also displayed in the description details are the endpoints; these are the pods associated with the service.

```
localadmin@k8s-master:~/demos$ kubectl describe svc/nsx-cdeweb-svc
Name:                nsx-cdeweb-svc
Namespace:           nsx-cde
Labels:              app=nsx-cdeweb
Annotations:         <none>
Selector:            app=nsx-cdeweb
Type:                ClusterIP
IP:                  10.107.102.32
Port:                <unset> 80/TCP
Endpoints:           10.4.0.162:80,10.4.0.163:80,10.4.0.164:80 + 1 more...
Session Affinity:    None
Events:              <none>
```

Figure 22: Load Balancing Service Description

Create an Ingress Service

The next step is to create an ingress service associated with the service tier. The ingress service allows for connection from outside of the K8s environment and assigns a friendly hostname. Figure 23 provides details of the yaml.

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: nsx-cdeweb-ingress
spec:
  rules:
  - host: nsx-cdeweb.demo.corp.local
    http:
      paths:
      - path: /
        backend:
          serviceName: nsx-cdeweb-svc
          servicePort: 80
```

Figure 23: Ingress Service Template

Figure 24 depicts the ingress creation.

```
localadmin@k8s-master:~/demos$ kubectl create -f nsx-demo-ingress.yaml
ingress "nsx-cdeweb-ingress" created
```

Figure 24: Create Ingress Controller Service

A description of the ingress service, depicted in Figure 25, shows the default http backend that is being used for the application and the rules to forward requests for the fully qualified domain name (FQDN) to the load balancing service, which will then redirect to one of the available pods.

```
localadmin@k8s-master:~/demos$ kubectl describe ingress nsx-cdeweb-ingress
Name:                nsx-cdeweb-ingress
Namespace:           nsx-cde
Address:
Default backend:     default-http-backend:80 (<none>)
Rules:
  Host                Path      Backends
  ----                -
  nsx-cdeweb.demo.corp.local  /         nsx-cdeweb-svc:80 (<none>)
Annotations:
Events: <none>
```

Figure 25: Description of Ingress Controller

Once the service is created, checking the ingress rule allows for verification that it is directing to the correct service and that web application represented by the pods is reachable. The web application has some identifiers built in to help with the verification process. Included was the name and the IP address of the pod. Figure 26 shows the response from the web service when tested including details that the web service was designed to reveal for the purpose of demonstrating K8s and NSX-T functionality.

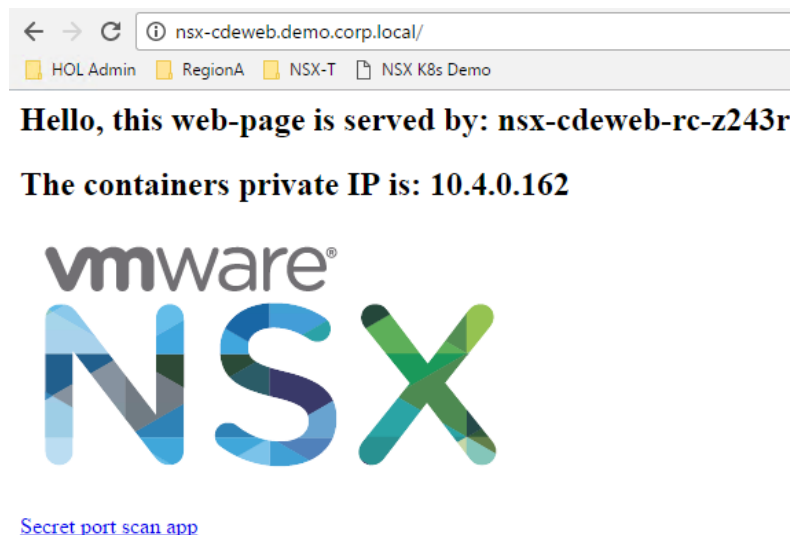
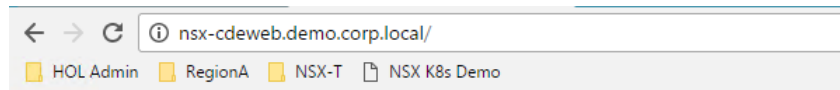


Figure 26: Working nsx-cdeweb Application

Refreshing the browser window allowed for verification of the functioning of the load balancing service. Each time the browser window was refreshed, the web page was serviced by a different pod. Figure 27 shows by the name of the pod and IP address that the refreshed web request was serviced by a different pod, which indicates the load balancing service is successful. The IP address reflected by the application was that of the deployed pod. Comparison of the details in the web application matched the values found when querying the application details within the K8s master and when viewed in the NSX-T management console.



Hello, this web-page is served by: nsx-cdeweb-rc-3qm9f

The containers private IP is: 10.4.0.164



[Secret port scan app](#)

Figure 27: Refreshing Browser Verifies Functioning Services for Load Balancing

To complete the lab setup, additional applications were deployed to represent dependent tiers of an application. The preceding steps were repeated with some variance to pod and security group labeling. Figure 28 shows the details of the lab once pods, replication controllers, and load balancing services were set up for each tier of the “nsx-cde” environment. The management pod in the list was used for some verification tasks such as checking pod to pod and intra namespace communication during the setup of the environment.

```
localadmin@k8s-master:~/demos$ kubectl get all -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE
po/mgmtpod    1/1     Running   0           3h    10.4.0.166   k8s-node2
po/nsx-cdeapp-rc-6fgmj  1/1     Running   0           21m   10.4.0.168   k8s-node1
po/nsx-cdeapp-rc-8gbw4  1/1     Running   0           21m   10.4.0.167   k8s-node2
po/nsx-cdeapp-rc-cn20  1/1     Running   0           21m   10.4.0.170   k8s-node2
po/nsx-cdeapp-rc-tfhfh  1/1     Running   0           21m   10.4.0.169   k8s-node1
po/nsx-cdedb-rc-mh7mr  1/1     Running   0           2m    10.4.0.171   k8s-node2
po/nsx-cdedb-rc-zwc6c  1/1     Running   0           2m    10.4.0.172   k8s-node1
po/nsx-cdeweb-rc-6jc2g  1/1     Running   0           6m    10.4.0.164   k8s-node1
po/nsx-cdeweb-rc-dlsgw  1/1     Running   0           6m    10.4.0.163   k8s-node2
po/nsx-cdeweb-rc-hjvix  1/1     Running   0           6m    10.4.0.165   k8s-node2
po/nsx-cdeweb-rc-j0hw0  1/1     Running   0           6m    10.4.0.162   k8s-node1
```

NAME	DESIRED	CURRENT	READY	AGE	CONTAINER(S)	IMAGE(S)	SELECTOR
rc/nsx-cdeapp-rc	4	4	4	21m	nsx-demo	yfauser/nsx-demo	app=nsx-cdeapp,secgroup=cde-app-tier
rc/nsx-cdedb-rc	2	2	2	2m	nsx-demo	yfauser/nsx-demo	app=nsx-cdedb,secgroup=cde-db-tier
rc/nsx-cdeweb-rc	4	4	4	6m	nsx-demo	yfauser/nsx-demo	app=nsx-cdeweb,secgroup=cde-web-tier

```
NAME          CLUSTER-IP      EXTERNAL-IP      PORT(S)  AGE   SELECTOR
svc/nsx-cdeapp-svc  10.106.101.255  <none>           80/TCP   18m   app=nsx-cdeapp
svc/nsx-cdedb-svc  10.98.79.123    <none>           80/TCP   1m    app=nsx-cdedb
svc/nsx-cdeweb-svc  10.107.102.32   <none>           80/TCP   3h    app=nsx-cdeweb
```

Figure 28: “nsx-cde” Pods, Replication Controllers, and Load Balancing Services

Additionally, the “nsx-noncde” environment was setup in a similar manner. This namespace was used for testing efficacy of controls for segmentation of the CDE from the non-CDE. Figure 29 depicts the running pods and replication controllers for the “nsx-noncde” namespace.

```
localadmin@k8s-master:~/demos$ kubectl get all -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE
po/nsx-noncdeweb-rc-4x2s6  1/1     Running   0           1m    10.4.0.98   k8s-node1
po/nsx-noncdeweb-rc-5zfd8  1/1     Running   0           1m    10.4.0.99   k8s-node2
```

NAME	DESIRED	CURRENT	READY	AGE	CONTAINER(S)	IMAGE(S)	SELECTOR
rc/nsx-noncdeweb-rc	2	2	2	1m	nsx-demo	yfauser/nsx-demo	app=nsx-noncdeweb,secgroup=noncde-web-tier

Figure 29: “nsx-noncde” Pods and Replication Controllers

Setup NSX-T to Enable Security Controls

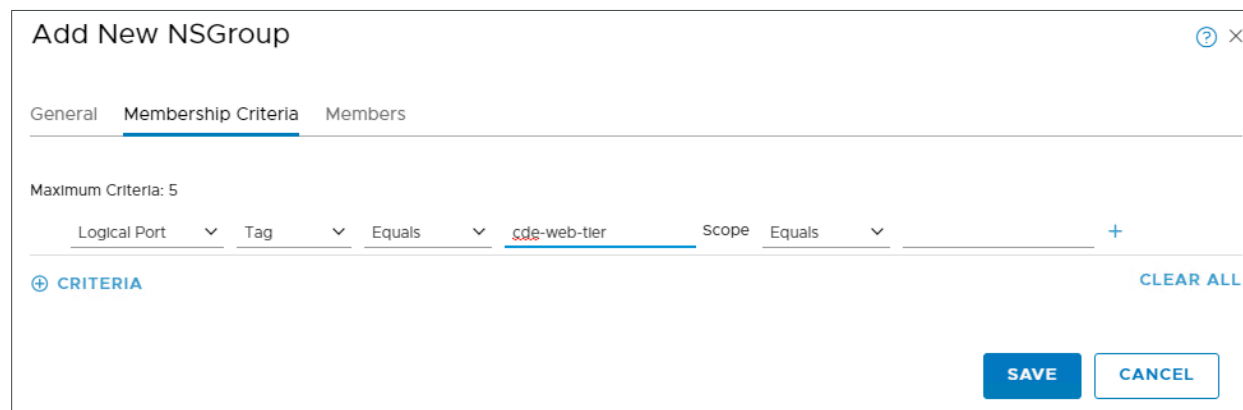
Within the NSX-T management console, NSGroups were setup. The NSGroups are capable of being configured for dynamic membership based on various criteria according to IP sets, MAC sets, logical switches, logical ports, and NSGroups. Membership of an NSGroup can be determined by more than one criterion. A NSGroup was created for each of the tiers of the application. This later allowed for policy creation with application of policy rules based on NSGroup membership. As an example, Figure 30 through Figure 33 depict the creation of the NSGroup.



The screenshot shows the 'Add New NSGroup' dialog box with the 'General' tab selected. The 'Name' field contains 'cde-web-tier' and the 'Description' field contains 'Cardholder Data Web Tier'. There are 'SAVE' and 'CANCEL' buttons at the bottom right.

Figure 30: Setting up NSGroup

In this example, the membership criteria for the NSGroups were defined by security tag values assigned to the pods. Defining the NSGroup by security tag, shown in Figure 31, allows for membership to this security group to be determined automatically by the security tag that was assigned to the pod as it was created.



The screenshot shows the 'Add New NSGroup' dialog box with the 'Membership Criteria' tab selected. It displays 'Maximum Criteria: 5' and a list of criteria: 'Logical Port', 'Tag', 'Equals', 'cde-web-tier', 'Scope', 'Equals'. There is a '+' button to add more criteria and a 'CLEAR ALL' button. There are 'SAVE' and 'CANCEL' buttons at the bottom right.

Figure 31: Defining Security Group Membership Criteria

Checking the members of the NSGroup shows which members fit the criteria and were automatically assigned to the group, as shown in Figure 32.

cde-web-tier				
Overview	Membership Criteria	Members	Applications Beta	Related
Members				
Object Type	Logical Port	Show Members	Effective	
Name	ID			
k8s-cl1-nsx-cdeweb-rc-6jc2g	79b6...f6c3			
k8s-cl1-nsx-cdeweb-rc-d1sgw	45b5...ee9b			
k8s-cl1-nsx-cdeweb-rc-hjjvx	23fe...2725			
k8s-cl1-nsx-cdeweb-rc-j0hw0	685e...e18d			

Figure 32: Dynamic Membership - Checking Members

When scaling up the available pods from the replication controller, the membership automatically increases to include the additional pods as shown in Figure 33.

cde-web-tier			
Overview	Membership Criteria	Members	Applications
Members			
Object Type	Logical Port	Show Members	Effective
Name	ID		
k8s-cl1-cde-kali-linux-3805672916-f9jdd	8544...e024		
k8s-cl1-nsx-cdeweb-rc-0tkk1	2623...27bc		
k8s-cl1-nsx-cdeweb-rc-6jc2g	79b6...f6c3		
k8s-cl1-nsx-cdeweb-rc-70bkg	af21...d9fe		
k8s-cl1-nsx-cdeweb-rc-d1sgw	45b5...ee9b		
k8s-cl1-nsx-cdeweb-rc-hjjvx	23fe...2725		
k8s-cl1-nsx-cdeweb-rc-j0hw0	685e...e18d		
k8s-cl1-nsx-cdeweb-rc-qlt43	ff06...300f		

Figure 33: Dynamic Membership - Scale Up Web Tier with Replication Controller

NSGroups were also created for the nsx-cdeapp and nsx-cdedb tiers. An additional NSGroup was created generically for the CDE as a whole. This more inclusive group was created to apply to all members of the CDE for overall application of the policy section.

A VM was also created on the vSphere hosts to represent a database server in the environment. The VM was a Linux, Apache, MySQL, and PHP server, primarily serving MySQL. To support the VM, a NSX-T

logical switch was created. The logical switch was attached to the T0 router and the port was assigned an IP range. Details of the logical switch to support the database tier of the application can be seen in Figure 34.

The screenshot shows the configuration page for a logical switch named 'vm-nsx-cde-db'. The page has tabs for Overview, Monitor, Manage, and Related. The Overview tab is active, showing a summary of the switch's configuration. The summary includes the name, ID, description, admin status (Up), replication mode, VNI, logical ports (2), traffic type (Overlay), transport zone (tz1), and creation/last updated dates. There are also expandable sections for Subnets and Tags.

vm-nsx-cde-db	
Overview Monitor Manage Related	
Summary EDIT	
Name	vm-nsx-cde-db
ID	67a43f92-b933-4c22-a540-126230ab4efa
Description	Cardholder Data Environment Database Virtual Machines
Admin Status	● Up
Replication Mode	Hierarchical Two-Tier replication
VNI	54161
Logical Ports	2
Traffic Type	Overlay
Transport Zone	tz1
Created	8/10/2017, 7:15:06 AM by admin
Last Updated	8/11/2017, 8:56:11 AM by admin
> Subnets	
> Tags MANAGE	

Figure 34: Logical Switch for DB VM

The router port was assigned 10.0.2.1/24 and the logical switch was attached to the T0 router. Figure 35 depicts the router logical port where the database vSwitch was attached.

The screenshot shows the configuration window for a router port named 'cde-db-switch'. The window includes fields for Name, Description, Type (Uplink, Downlink, Loopback), URPF Mode (Strict, None), Logical Switch (vm-nsx-cde-db), Logical Switch Port (Attach to existing switch port), IP Address/mask (10.0.2.1/24), and Relay Service. There are 'SAVE' and 'CANCEL' buttons at the bottom.

Edit Router Port - cde-db-switch	
Name*	cde-db-switch
Description	
Type	<input type="radio"/> Uplink <input checked="" type="radio"/> Downlink <input type="radio"/> Loopback
URPF Mode	<input checked="" type="radio"/> Strict <input type="radio"/> None
Logical Switch	vm-nsx-cde-db x
Logical Switch Port	<input type="radio"/> Attach to new switch port <input checked="" type="radio"/> Attach to existing switch port Switch Port Name 625f755f-b599-45d9-851e-b316872 x
IP Address/mask*	10.0.2.1/24
Relay Service	x
SAVE CANCEL	

Figure 35: Logical Port Configuration to Support DB Logical Switch

Figure 36 shows the cde-db-switch attached to the Central-T0-Router.

Logical Router Port	ID	Type	IP Address/mask	Connected To	Transport Node	Relay Service	Statistics
TIERO-RouterLinkPort	07d8...7c...	Linked Port	100.64.192.6/31	k8s-cl1-nsx-cde			
TIERO-RouterLinkPort	6fc4...3d3f	Linked Port	100.64.192.8/31	k8s-cl1-nsx-noncde			
TIERO-RouterLinkPort	b443...3cd9	Linked Port	100.64.192.4/31	k8s-cl1-kube-public			
TIERO-RouterLinkPort	c96c...f9a4	Linked Port	100.64.192.2/31	k8s-cl1-kube-system			
TIERO-RouterLinkPort	f816...e990	Linked Port	100.64.192.0/31	k8s-cl1-default			
cde-db-switch	fd2c...d687	Downlink	10.0.2.1/24	vm-nsx-cde-db 625f755f-b599-45d...			
k8s-mgmt-downlink	051b...0bb1	Downlink	10.0.1.1/24	k8s-mgmt ccfce6b7-142d-41ea...			
uplink-to-lab	f8f4...6a6a	Uplink	192.168.100.3/24	uplink-to-lab 894e0c54-a4c6-4b...	edge.corp.local		

Figure 36: T0 Router Configuration Showing cde-db-switch Connected

The vm-nsx-cde-db switch was added to the CDE NSGroup. The VM could either be added using membership criteria to include the members of the logical switch or a security tag could be manually added to the switch or switch ports. Figure 37 shows members of the CDE NSGroup inclusive of the members of the “nsx-cde” switch and the vm-nsx-cde-db switch members.

Name	ID
k8s-cl1-nsx-cde-0	5bbd...96e1
vm-nsx-cde-db	67a4...4efa

Figure 37: CDE Members

With the completion of the switch and security context values, the CDE database VM was added to the vm-nsx-cde-db switch and given a static IP address from the IP Pool. The IP address for the DB VM was 10.0.2.10.

Name	Type	Network Protocol Profile	VMs	Hosts
vm-nsx-cde-db	Opaque network		1	2

Figure 38: NSX Switch from vCenter Console

Figure 39 provides a characterization of the lab. The topology shows the relationship of the T1 and T0 routers as well as the subnets represented by each environment. The distributed firewall is illustrated as being deployed to the pods. The next section discusses the methodology that was used for segmentation testing. The diagram shows the location of Kali Linux pods to perform segmentation testing. For traditional segmentation testing, Coalfire placed a Kali Linux pod outside of the CDE in the “nsx-noncde” namespace. Additional testing was performed to demonstrate the efficacy of NSX-T to provide more granular segmentation capability within a namespace. For this testing, an additional Kali Linux pod was deployed inside the CDE.

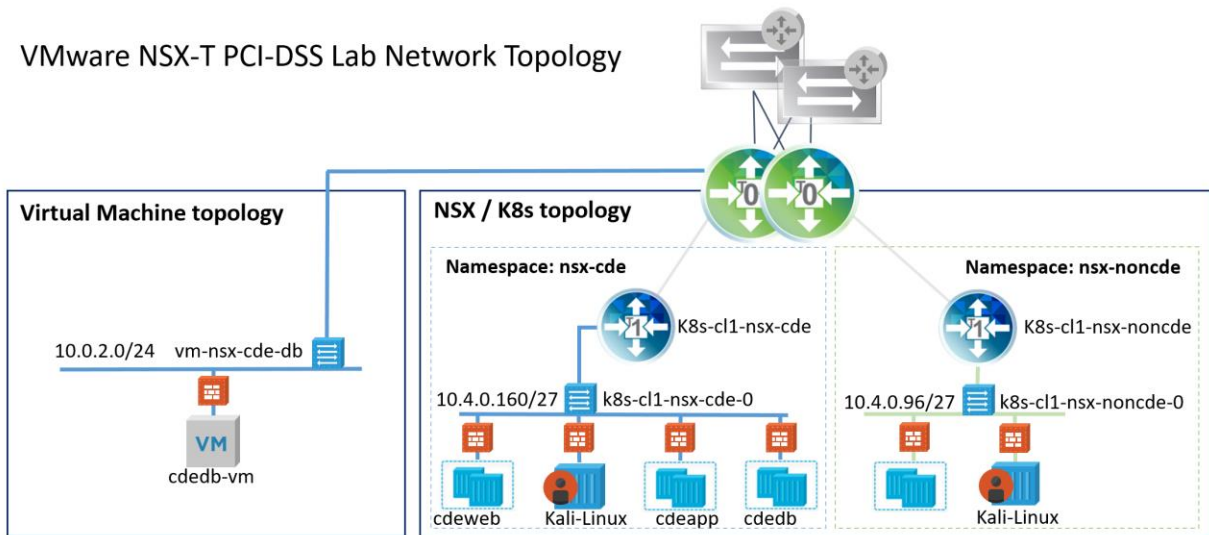


Figure 39: NSX-T Lab Network Topology Diagram

Figure 40 shows the NSX-T DFW section created for the CDE. The first rule in the policy denies all traffic that isn't specified by exception in the rules that follow as read from the bottom up. The ingress rule allows for authorized connections to be made to the applications web tier. The policy section was applied to members of the CDE logical switch to exclude any containers or VMs not attached to the switch.

#	Name	ID	Sources	Destinations	Services	Action	Applied To
CDE-MicroServices			1decff1f-8415-4440-99ee-e4b94f67a336	Stateful	Applied To: 1		
1	cde-app to cde-db	3088	cde-app-tier	cde-db-tier	HTTP	Allow	All
2	cde-web to cde-app	3089	cde-web-tier	cde-app-tier	HTTP	Allow	All
3	Ingress Rule	3091	Ingresses	cde-web-tier	HTTP	Allow	All
4	Default	3090	Any	Any	Any	Drop	All

Figure 40: Cardholder Data Environment Distributed Firewall Section

SEGMENTATION TESTING METHODOLOGY

Coalfire's testing included a baseline test without NSX-T controls in place as a starting point to provide comparison of NSX-T DFW efficacy for providing micro-segmentation of the workload components. For segmentation testing, Coalfire utilized db_nmap, a common tool used by the penetration testing community for discovery of networks and network resources. Coalfire used a comprehensive set of db_nmap commands to sufficiently test the boundary protection measures. A Kali Linux container was placed inside of each K8s namespace to facilitate the various testing parameters. This Kali Linux container was the source for running the db_nmap utility. A policy section was added to the NSX-T firewall configuration to define the access between the tiers of the CDE service. The policy included a deny-all rule to prevent communication from any VM, container, or pod outside of the defined CDE.

SEGMENTATION TESTING FINDINGS AND SUMMARY RESULTS

Baseline Results

The nmap utility was run for each variation testing. For the baseline, without security policies applied, it was expected that every host (vm, container, pod) in the environment would be found. As this was white box testing, the tester was previously aware of the devices on the network. For baseline testing, and to speed up the testing process, a host file was created to list each IP address that nmap should try to find. In the script, this file is defined with the -iL parameter and named cdescope. This lists the IP addresses of the pods in the CDE. The following shows the results of the nmap scan, which revealed the IP address and the available open ports for the pod.

```
# Nmap 7.50 scan initiated Thu Aug 10 17:41:32 2017 as: nmap -sT -Pn --disable-arp --open -n -iL cdescope -oA wfw0/scan0 -p- -vvv --min-hostgroup 26
# Ports scanned: TCP(65535;1-65535) UDP(0;); SCTP(0;); PROTOCOLS(0;);
Host: 10.4.0.166 () Status: Up
Host: 10.4.0.166 () Ports: 22/open/tcp//ssh///
Host: 10.4.0.168 () Status: Up
Host: 10.4.0.168 () Ports: 80/open/tcp//http///
Host: 10.4.0.167 () Status: Up
Host: 10.4.0.167 () Ports: 80/open/tcp//http///
Host: 10.4.0.170 () Status: Up
Host: 10.4.0.170 () Ports: 80/open/tcp//http///
Host: 10.4.0.169 () Status: Up
Host: 10.4.0.169 () Ports: 80/open/tcp//http///
Host: 10.4.0.172 () Status: Up
Host: 10.4.0.172 () Ports: 80/open/tcp//http///
Host: 10.4.0.171 () Status: Up
Host: 10.4.0.171 () Ports: 80/open/tcp//http///
Host: 10.4.0.164 () Status: Up
Host: 10.4.0.164 () Ports: 80/open/tcp//http///
Host: 10.4.0.163 () Status: Up
Host: 10.4.0.163 () Ports: 80/open/tcp//http///
```



```

Host: 10.4.0.165 () Status: Up
Host: 10.4.0.165 () Ports: 80/open/tcp/http///
Host: 10.4.0.162 () Status: Up
Host: 10.4.0.162 () Ports: 80/open/tcp/http///
# Nmap done at Thu Aug 10 17:43:36 2017 -- 12 IP addresses (12 hosts up) scanned in 123.26 seconds

```

With Firewall Rules in Place and Applied to NSGroups

The firewall was set up and enabled as shown in Figure 40. With this in place, several tests were performed including:

- From the Kali Linux instance in the “nsx-noncde” namespace to attempt to discover the “nsx-cde” namespace pods.
- From the Kali Linux instance in the “nsx-cde” namespace as part of the web tier to attempt to discover other pods in the web tier.
- From the Kali Linux instance in the “nsx-cde” namespace as part of the web tier to attempt to discover pods in the db tier.
- From the Kali Linux instance in the “nsx-cde” namespace as part of the web tier to attempt to discover pods from the app tier.
- From the Kali Linux instance in the “nsx-cde” namespace as part of the web tier to attempt to discover the database VM.

From outside of and targeting the cardholder data environment

With the firewall rules applied, the nmap scan from outside the CDE shows that the firewall rules prevented any hosts from being discovered.

```

# Nmap 7.50 scan initiated Fri Aug 11 16:25:22 2017 as: nmap -sT -Pn --disable-arp --open -n -oA wfw0/scan0
-p- -vvv --min-hostgroup 26 10.4.0.160/27 10.0.2.10
# Ports scanned: TCP(65535;1-65535) UDP(0;) SCTP(0;) PROTOCOLS(0;)
# Nmap done at Sat Aug 12 03:32:02 2017 -- 33 IP addresses (33 hosts up) scanned in 39999.49 seconds

```

The distributed firewall was successfully able to segment the CDE from the non-CDE.

From inside the cardholder data environment web tier to test intra segment distributed firewall micro-segmentation

With the firewall rules applied, the nmap scan from inside the CDE shows that the firewall rules prevented other pods/containers designated as web tier from being discovered. Additionally, the discovery of app and database tier pods and VMs were in alignment with expected results based on the defined rules supporting access over permitted ports. Moreover, the findings remained consistent as the replication controller was used to scale up the number of pods in the web tier.

```

# Nmap 7.50 scan initiated Thu Aug 10 19:16:14 2017 as: nmap -sT -Pn --disable-arp --open -n -oA wfw0/scan0
-p- --top-ports 1000 -vvv --min-hostgroup 26 10.4.0.160/27 10.0.2.10
# Ports scanned: TCP(1000;1,3-4,6-7,9,13,17,19-26,30,32-33,37,42-43,49,53,70,79-85,88-90,99-100,106,109-
111,113,119,125,135,139,143-144,146,161,163,179,199,211-212,222,254-
256,259,264,280,301,306,311,340,366,389,406-407,416-417,425,427,443-445,458,464-465,481,497,500,512-
515,524,541,543-545,548,554-555,563,587,593,616-617,625,631,636,646,648,666-
668,683,687,691,700,705,711,714,720,722,726,749,765,777,783,787,800-801,808,843,873,880,888,898,900-
903,911-912,981,987,990,992-993,995,999-1002,1007,1009-1011,1021-1100,1102,1104-1108,1110-
1114,1117,1119,1121-1124,1126,1130-1132,1137-1138,1141,1145,1147-1149,1151-1152,1154,1163-
1166,1169,1174-1175,1183,1185-1187,1192,1198-1199,1201,1213,1216-1218,1233-1234,1236,1244,1247-
1248,1259,1271-1272,1277,1287,1296,1300-1301,1309-1311,1322,1328,1334,1352,1417,1433-
1434,1443,1455,1461,1494,1500-
1501,1503,1521,1524,1533,1556,1580,1583,1594,1600,1641,1658,1666,1687-1688,1700,1717-
1721,1723,1755,1761,1782-1783,1801,1805,1812,1839-1840,1862-1864,1875,1900,1914,1935,1947,1971-
1972,1974,1984,1998-2010,2013,2020-2022,2030,2033-2035,2038,2040-2043,2045-2049,2065,2068,2099-
2100,2103,2105-2107,2111,2119,2121,2126,2135,2144,2160-2161,2170,2179,2190-
2191,2196,2200,2222,2251,2260,2288,2301,2323,2366,2381-2383,2393-

```

```

2394,2399,2401,2492,2500,2522,2525,2557,2601-2602,2604-2605,2607-2608,2638,2701-2702,2710,2717-
2718,2725,2800,2809,2811,2869,2875,2909-2910,2920,2967-2968,2998,3000-3001,3003,3005-
3007,3011,3013,3017,3030-3031,3052,3071,3077,3128,3168,3211,3221,3260-3261,3268-3269,3283,3300-
3301,3306,3322-3325,3333,3351,3367,3369-3372,3389-
3390,3404,3476,3493,3517,3527,3546,3551,3580,3659,3689-3690,3703,3737,3766,3784,3800-
3801,3809,3814,3826-
3828,3851,3869,3871,3878,3880,3889,3905,3914,3918,3920,3945,3971,3986,3995,3998,4000-
4006,4045,4111,4125-4126,4129,4224,4242,4279,4321,4343,4443-4446,4449,4550,4567,4662,4848,4899-
4900,4998,5000-5004,5009,5030,5033,5050-5051,5054,5060-5061,5080,5087,5100-
5102,5120,5190,5200,5214,5221-5222,5225-5226,5269,5280,5298,5357,5405,5414,5431-
5432,5440,5500,5510,5544,5550,5555,5560,5566,5631,5633,5666,5678-5679,5718,5730,5800-5802,5810-
5811,5815,5822,5825,5850,5859,5862,5877,5900-5904,5906-5907,5910-
5911,5915,5922,5925,5950,5952,5959-5963,5987-5989,5998-6007,6009,6025,6059,6100-
6101,6106,6112,6123,6129,6156,6346,6389,6502,6510,6543,6547,6565-6567,6580,6646,6666-
6669,6689,6692,6699,6779,6788-6789,6792,6839,6881,6901,6969,7000-
7002,7004,7007,7019,7025,7070,7100,7103,7106,7200-
7201,7402,7435,7443,7496,7512,7625,7627,7676,7741,7777-7778,7800,7911,7920-7921,7937-7938,7999-
8002,8007-8011,8021-8022,8031,8042,8045,8080-8090,8093,8099-8100,8180-8181,8192-
8194,8200,8222,8254,8290-8292,8300,8333,8383,8400,8402,8443,8500,8600,8649,8651-
8652,8654,8701,8800,8873,8888,8899,8994,9000-9003,9009-9011,9040,9050,9071,9080-9081,9090-
9091,9099-9103,9110-9111,9200,9207,9220,9290,9415,9418,9485,9500,9502-9503,9535,9575,9593-
9595,9618,9666,9876-9878,9898,9900,9917,9929,9943-9944,9968,9998-10004,10009-10010,10012,10024-
10025,10082,10180,10215,10243,10566,10616-10617,10621,10626,10628-10629,10778,11110-
11111,11967,12000,12174,12265,12345,13456,13722,13782-13783,14000,14238,14441-14442,15000,15002-
15004,15660,15742,16000-16001,16012,16016,16018,16080,16113,16992-
16993,17877,17988,18040,18101,18988,19101,19283,19315,19350,19780,19801,19842,20000,20005,20031,2
0221-20222,20828,21571,22939,23502,24444,24800,25734-25735,26214,27000,27352-27353,27355-
27356,27715,28201,30000,30718,30951,31038,31337,32768-32785,33354,33899,34571-
34573,35500,38292,40193,40911,41511,42510,44176,44442-44443,44501,45100,48080,49152-
49161,49163,49165,49167,49175-49176,49400,49999-
50003,50006,50300,50389,50500,50636,50800,51103,51493,52673,52822,52848,52869,54045,54328,55055-
55056,55555,55600,56737-
56738,57294,57797,58080,60020,60443,61532,61900,62078,63331,64623,64680,65000,65129,65389)
UDP(0:) SCTP(0:) PROTOCOLS(0:)
Host: 10.4.0.167 () Status: Up
Host: 10.4.0.167 () Ports: 80/open/tcp/http/// Ignored State: filtered (999)
Host: 10.4.0.168 () Status: Up
Host: 10.4.0.168 () Ports: 80/open/tcp/http/// Ignored State: filtered (999)
Host: 10.4.0.169 () Status: Up
Host: 10.4.0.169 () Ports: 80/open/tcp/http/// Ignored State: filtered (999)
Host: 10.4.0.170 () Status: Up
Host: 10.4.0.170 () Ports: 80/open/tcp/http/// Ignored State: filtered (999)
Host: 10.0.2.10 () Status: Up
Host: 10.0.2.10 () Ports: 443/open/tcp/https/// Ignored State: filtered (999)
# Nmap done at Thu Aug 10 19:20:57 2017 -- 33 IP addresses (33 hosts up) scanned in 282.31 seconds

```

The NSX-T DFW was able to successfully provide micro-segmentation capabilities to enable policy down to the pod/container in the environment.

NSX-T ALIGNMENT WITH PCI DSS REQUIREMENTS

The following table outlines the alignment of applicability of NSX-T 2.0 control capabilities with PCI DSS 3.2 requirements. This includes identification for where NSX-T can be used to fully or partially address a requirement when properly implemented by a payment entity for the protection of the CDE. This also includes requirements that should apply to the implementation of NSX-T where the technology takes place as either a direct part or indirect part of the CDE.

Requirements that are not applicable to NSX-T capabilities or scope have been excluded from the table. Coalfire determined that a requirement may not be applicable to NSX-T when the requirement narrative, testing procedures, and guidance indicate that it is relative to the production of organizational documented standards, procedures, diagrams, or other processes. All requirements are applicable to the payment entity and must be strictly followed to be compliant.

When a testing process, as part of a requirement, asks for verification of technical configuration as verification of follow through on a payment entity’s standard, NSX-T may be considered partially in scope. The assessment of applicability for this document does not establish a relationship between all possible standards documented by any given payment entity. Coalfire can determine that NSX-T could possibly be the source for assessment for verification of the payment entity’s adherence to that standard. Moreover, Coalfire may determine, based on assessment of the technology’s purpose, configuration options, and documented capabilities, what evidence of adherence to payment entity’s standards implementation may be identified with NSX-T.

When the properly implemented technology can address the requirement, the technology is determined to be fully applicable. This does not exclude the use of other technologies in combination with or instead of NSX-T by the payment entity. Any technology that a payment entity employs to address a technical control objective should be in scope of assessment for completeness and effectiveness of control enablement.

Each applicable requirement is described in the table in the following section. This table includes the findings of applicability along with a short narrative describing the capability. The table uses Harvey Balls to describe the qualitative applicability of the technology to the requirement. (see: https://en.wikipedia.org/wiki/Harvey_Balls)

The following key describes the meaning for the Harvey Balls.

DESIGNATION	KEY	MEANING
Fully Supported	●	The evaluated technology has the means, through configuration or by design, to fully support the requirement.
Partially Supported	◐	The evaluated technology has the means, through configuration or by design, to partially support the requirement. Additional technology or organizational policy, procedures, or standards may be required to fully meet the requirement objective.
Not Supported	○	The evaluated technology does not have the means, through configuration or by design, to support or meet the requirement objective. The technology either relies on an external technology, system, or organizational procedure to meet the requirement objective or the use of an external technology is highly recommended.
Does Not Apply	N/A	The requirement, though it may be technical in nature, is not applicable to the evaluated technology. Meeting the requirement is completely achieved independently of the evaluated technology.

DESIGNATION	KEY	MEANING
Non-Tech	Non-Tech	The requirement is purely operational requiring a defined and documented policy, procedure, or standard. This may also include operational, non-technical procedures such as the implementation of training, maintaining of paper logs, or performance of personnel identity verification.

Table 1: Harvey Balls Key

The following table represents Coalfire’s findings of applicability.



REQ ID	REQUIREMENT TEXT	SCOPE	FINDINGS
1.1.4	Requirements for a firewall at each Internet connection and between any demilitarized zone (DMZ) and the internal network zone	●	<p>This requirement is primarily related to the creation of standards documentation by the payment entity. The capability of NSX-T to support this requirement pertains to the ability to show as evidence adherence to the payment entity’s standards. Because NSX-T cannot possibly document the payment entity standards, Coalfire determines that it can only partially support the requirement.</p> <p>However, NSX-T can support implementation of a firewall to protect internal trusted network devices and to provide perimeter protection for those devices to restrict access from Internet sources and between any DMZ and the internal network zone. This capability can be demonstrated to an assessor as evidence of adherence to the payment entity’s documented standards. Whether NSX-T can fully support those standards is dependent upon the standards as they are written.</p> <p>Depending on the payment entity’s infrastructure, it is recommended to place traditional edge firewall(s) between the “untrusted” network and the “trusted” network, where “untrusted” network refers to a network that is outside the control of the payment entity.</p>




REQ ID	REQUIREMENT TEXT	SCOPE	FINDINGS
1.1.6	Documentation of business justification and approval for use of all services, protocols, and ports allowed, including documentation of security features implemented for those protocols considered to be insecure.	●	<p>This requirement is primarily related to the creation of standards documentation. The capability of NSX-T to support this requirement pertains to the ability to show as evidence adherence to the payment entity's standards. This requirement is not strictly a technical requirement. The ability of NSX-T to fully support implementation of the payment entity's standards is dependent upon the standards as they are written.</p> <p>NSX-T provides mechanisms to allow organizations to implement controls to restrict, based on the payment entity's business justification and documented standards, the services and ports authorized for use between devices. The payment entity can demonstrate compliance to the documented standard through routing and firewall policy rulesets in the NSX-T management console.</p>
1.2	<p>Build firewall and router configurations that restrict connections between untrusted networks and any system components in the cardholder data environment.</p> <p>Note: An "untrusted network" is any network that is external to the networks belonging to the entity under review, and/or which is out of the entity's ability to control or manage.</p>	●	NSX-T is capable of being used to establish and provide control for its managed network segments to isolate system components in the CDE from untrusted networks. Assuming a Zero Trust model for IT security, NSX-T can provide segmentation down to individual workloads and used to validate network transmissions between workloads as authorized according to policy.
1.2.1	Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment, and specifically deny all other traffic.	●	With the distributed firewall, NSX-T can be configured to provide a perimeter around CDE systems to restrict traffic to only what is necessary to support the CDE. Policy sets can be applied to the DFW with a default rule to deny all traffic and allow traffic only by, specified in the policy section, exception. The extent that perimeter protections can be applied is granular to the individual workload in the environment whether container or VM.

REQ ID	REQUIREMENT TEXT	SCOPE	FINDINGS
1.2.2	Secure and synchronize router configuration files.	●	Router and firewall configuration is applied and synchronized to the data plane from the control plane. Security engineers configure policies with the NSX Manager. The NSX Manager performs a validation and stores the firewall configuration elements such as rules, sections, and grouping objects. If there is a cluster of NSX Managers, the DFW configuration will be persistently stored across all NSX Managers in the cluster. The policy is then pushed to the control plane cluster. The controller takes the configured rules and converts the objects that are sent in the rule definition (such as logical switches and NSGroups) into IP addresses. The policy using IP addresses is then pushed to the hypervisor.
1.3	Prohibit direct public access between the Internet and any system component in the cardholder data environment.	●	For VMs and containers, NSX-T can be used to restrict access from the Internet to elements in the CDE. The NSX-T DFW can be assigned to each workload that includes policies to control communication from untrusted or lower trust networks.
1.3.1	Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.	●	NSX-T is capable of being used to create, manage, and monitor traffic around a segment classified by the payment entity as the DMZ. Each element within the DMZ, whether VM or container, can be micro-segmented to prevent east-west pivoting as well as limiting inbound traffic to DMZ asset from outside the DMZ. The DFW can isolate DMZ components and limit the external inbound traffic to only those services that require it.
1.3.2	Limit inbound Internet traffic to IP addresses within the DMZ.	●	NSX-T can be configured to support standard 5-tuple firewall rules to limit inbound traffic to specific IP addresses within the DMZ, where the destination is specified as an IP address or IP addresses in the DMZ

REQ ID	REQUIREMENT TEXT	SCOPE	FINDINGS
1.3.3	Implement anti-spoofing measures to detect and block forged source IP addresses from entering the network. (For example, block traffic originating from the Internet with an internal source address.)	●	Coalfire tested the anti-spoofing measures of NSX-T and found that they were successful in blocking forged source IP addresses from entering the protected network. Coalfire performed an ARP spoofing attack to attempt to man-in-the-middle the targeted hosts. NSX-T Spoofguard successfully prevented the attack from being successful. Spoofguard is enabled automatically on all logical switch ports. Spoofguard enforces MAC+IP+VLAN bindings to avoid spoofing scenarios.
1.3.4	Do not allow unauthorized outbound traffic from the cardholder data environment to the Internet.	●	NSX-T can be configured to block outbound traffic from the cardholder data environment to the Internet. The granularity of control for blocking outbound traffic is down to the individual workload – VM or container.
1.3.5	Permit only “established” connections into the network.	●	NSX-T DFW is a stateful firewall. This means that the DFW keeps track of every connection that comes through the firewall. For TCP traffic, the DFW looks at each attribute of each packet, such as flags and sequence numbers, to determine the state of connections and only allow packets that are either a SYN packet for a new connection explicitly allowed in the rule table or an expected packet, with the correct sequence number and correct flag based on the current state of the connection in the connection table. For UDP, state is tracked based on source, destination IP, and port.
1.3.6	Place system components that store cardholder data (such as a database) in an internal network zone, segregated from the DMZ and other untrusted networks.	●	NSX-T can be configured to further segment system components of the CDE such as the database in an internal network zone. The capability of segmentation is granular down to the workload or VM level.

REQ ID	REQUIREMENT TEXT	SCOPE	FINDINGS
1.3.7	<p>Do not disclose private IP addresses and routing information to unauthorized parties.</p> <p>Note: <i>Methods to obscure IP addressing may include, but are not limited to:</i></p> <ul style="list-style-type: none"> • Network Address Translation (NAT) • Placing servers containing cardholder data behind proxy servers/firewalls, • Removal or filtering of route advertisements for private networks that employ registered addressing, • Internal use of RFC1918 address space instead of registered addresses. 	<p>●</p>	<p>Through the distributed logical router, NSX-T can enable overlay and VLAN-backed logical switches to further obfuscate private IP addresses and routing information from unauthorized parties.</p> <p>Additionally, NSX-T can be configured to NAT addresses to prevent discovery of private addresses within each segment.</p> <p>NSX-T DFW can provide perimeter protections at each CDE server, as well as the entire CDE.</p>
2.1	<p>Always change vendor-supplied defaults and remove or disable unnecessary default accounts before installing a system on the network.</p> <p>This applies to ALL default passwords, including but not limited to those used by operating systems, software that provides security services, application and system accounts, point-of-sale (POS) terminals, payment applications, Simple Network Management Protocol (SNMP) community strings, etc.).</p>	<p>●</p>	<p>NSX-T appliances have a local administrative user, admin. Users cannot be created or deleted from the appliance. However, the password can be changed for the admin user on the NSX-T appliance. The change is performed at the NSX Manager CLI.</p>

REQ ID	REQUIREMENT TEXT	SCOPE	FINDINGS
2.2.1	<p>Implement only one primary function per server to prevent functions that require different security levels from co-existing on the same server. (For example, web servers, database servers, and DNS should be implemented on separate servers.)</p> <p><i>Note: Where virtualization technologies are in use, implement only one primary function per virtual system component.</i></p>		<p>The nature of virtualization and container solutions is that it provides greater efficiency to support implementation of a single function per workload component. This allows for the purpose of the individual workload whether container or VM to be narrowed.</p> <p>NSX-T further provides support for this isolation by enabling micro-segmentation to limit the functional communication of each device to its primary function where the VM or container have a single function, this reduces the surface area for attack. The combination of NSX-T distributed firewall with purpose built and hardened systems acts as a check and balance against the possibility of accidental misconfiguration or vulnerability resulting from change. Partial support is assigned to NSX-T, as the payment entity must deploy workloads in a fashion that adheres to the requirement. NSX-T cannot strictly limit the number of functions that the payment entity assigns to a VM or container.</p>
2.2.2	<p>Enable only necessary services, protocols, daemons, etc., as required for the function of the system.</p>		<p>While this applies to the host system configuration, It is worth noting that NSX-T is capable of further ensuring the limitation of ability for services to communicate and enabling only the ports and paths that are relevant to the function of the server. This is beyond the requirement defined here to harden the system by eliminating unnecessary services, protocols, daemons, etc. NSX-T is able to augment the hardened security of the VM or container by mirroring network policies to limit network transmission from and to the server to that which is necessary for the server's primary function.</p>

REQ ID	REQUIREMENT TEXT	SCOPE	FINDINGS
2.2.3	<p>Implement additional security features for any required services, protocols, or daemons that are considered to be insecure.</p> <p><i>Note: Where SSL/early TLS is used, the requirements in Appendix A2 must be completed.</i></p>		<p>NSX-T DFW can be deployed to provide granular control over communication between devices. This allows for specificity when setting up rules to limit the source and destination to single objects in the network.</p> <p>NSX-T Distributed Network Encryption can encrypt all transmissions within the secure network and inside the CDE for network transactions between endpoints. This can also be useful as a deterrent from network sniffing used for intercepting packets in transit.</p> <p>It is recommended to also implement advanced network security solutions in these circumstances (IDS/IPS, next generation application firewalls, and such) to more fully ensure the security of network transmissions where insecure services, protocols, or daemons are in use.</p>
2.3	<p>Encrypt all non-console administrative access using strong cryptography.</p> <p><i>Note: Where SSL/early TLS is used, the requirements in Appendix A2 must be completed.</i></p>		<p>NSX-T is capable of being configured to support encryption for non-console web-based administrative access to the management functions through SSL and through remote console access with SSH. Likewise, non-console API calls to the NSX-T Manager can also be encrypted.</p>
6.2	<p>Ensure that all system components and software are protected from known vulnerabilities by installing applicable vendor-supplied security patches. Install critical security patches within one month of release.</p> <p><i>Note: Critical security patches should be identified according to the risk ranking process defined in Requirement 6.1.</i></p>		<p>VMware provides regular updates to its software for its customers to deploy. It is recommended that the payment entity include VMware software in its patching procedures and apply security patches and fixes during regular payment entity-defined standard intervals.</p> <p>While VMware provides updates for NSX-T and other VMware solutions, it is ultimately the responsibility of the payment entity to apply the updates. NSX-T updates are not strictly enforced by the software and require intervention or action to be taken by the payment entity support staff.</p>

REQ ID	REQUIREMENT TEXT	SCOPE	FINDINGS
7.2	<p>Establish an access control system(s) for systems components that restricts access based on a user's need to know, and is set to "deny all" unless specifically allowed.</p> <p>This access control system(s) must include the following:</p>	<ul style="list-style-type: none"> ● 	<p>NSX-T contains a local CLI admin account. No additional user accounts can be added nor can the built-in CLI admin account be deleted. This admin account should be limited by vaulting the password inasmuch as it is possible to do so. Use of this account should be strictly monitored and the account should only be used in the event of an emergency where access cannot be granted by any other means.</p> <p>It is recommended to integrate NSX-T with VMware Identity Manager for enhanced authentication support. VMware Identity Manager provides support for AD-based LDAP, OpenLDAP, RADIUS, SmartCards/ Common Access Cards, and RSA Secure ID. Moreover, it can provide support for Enterprise Single Sign On. Through this integration, the payment entity may be able to fully comply with this requirement as it pertains to authentication and authorization for NSX-T.</p> <p>Per need to know requirements, NSX-T supports six pre-defined rules (Enterprise Administrator, Network Engineer, Network Operator, Security Engineer, Security Operator, and Auditor) with varying degrees of access available per role. These roles can be tied to individual users through the integrated access management solution. Access control is enforced on the feature level.</p>
7.2.1	Coverage of all system components.	<ul style="list-style-type: none"> ● 	Access controls for NSX-T provide coverage for all NSX-T system components whether by console access, web client access, SSH access, or API call.
7.2.3	Default "deny-all" setting.	<ul style="list-style-type: none"> ● 	NSX-T does not provide any anonymous access. All access to NSX-T administrative functions is done so with explicitly granting of access rights and assigning of payment entity users to NSX-T roles.

REQ ID	REQUIREMENT TEXT	SCOPE	FINDINGS
8.1.4	Remove/disable inactive user accounts within 90 days.	●	By itself, NSX-T only has one single local CLI admin account. As this violates requirements to assign individual access to users, It is recommended to integrate NSX-T with VMware Identity Manager connected to one of the supported AAA systems. Through this mechanism the payment entity may be able to fully comply with the requirement as it pertains to authentication to NSX-T.
8.1.6	Limit repeated access attempts by locking out the user ID after not more than six attempts.	●	<p>By itself, NSX-T only has the single local CLI admin account. As this violates requirements to assign individual access to users, it is recommended to integrate NSX-T with VMware Identity Manager connected to one of the supported AAA systems. In this case, the AAA system will provide the account lockout threshold. Through this mechanism, the payment entity will be able to fully comply with the requirement.</p> <p>For THE local CLI admin account, a threshold of five consecutive failed logon attempts is the default setting that results in account lockout. The threshold for lockout is configurable with NSX-T 2.0 and can be set through the CLI.</p>
8.1.7	Set the lockout duration to a minimum of 30 minutes or until an administrator enables the user ID.	●	<p>It is recommended to integrate NSX-T with VMware Identity Manager connected to one of the supported AAA systems. Through this integration, the payment entity may be able to fully comply with this requirement as it pertains to authentication and authorization for NSX-T.</p> <p>For THE local CLI admin account, the default lockout duration is 15 minutes. This must be increased to 30 minutes to support PCI DSS compliance. The value for lockout duration can be set in the CLI with NSX-T 2.0 with the “<i>set auth-policy api lockout-period</i>” and “<i>set auth-policy cli lockout-period</i>” commands.</p>

REQ ID	REQUIREMENT TEXT	SCOPE	FINDINGS
8.1.8	If a session has been idle for more than 15 minutes, require the user to re-authenticate to re-activate the terminal or session.	●	It is recommended to integrate NSX-T with VMware Identity Manager connected to one of the supported AAA systems. Through this integration, the payment entity may fully support the requirement as it pertains to authentication to NSX-T. No such restriction exists for the local CLI admin account.
8.2.1	Using strong cryptography, render all authentication credentials (such as passwords/phrases) unreadable during transmission and storage on all system components.	●	It is recommended to integrate NSX-T with VMware Identity Manager connected to one of the supported AAA systems. Through this integration, the payment entity may be able to fully comply with this requirement as it pertains to authentication and authorization for NSX-T.
8.2.2	Verify user identity before modifying any authentication credential—for example, performing password resets, provisioning new tokens, or generating new keys.	●	It is recommended to integrate NSX-T with VMware Identity Manager connected to one of the supported AAA systems. Through this integration, the payment entity may be able to fully comply with this requirement as it pertains to authentication and authorization for NSX-T.
8.2.3	Passwords/passphrases must meet the following: -Require a minimum length of at least seven characters. -Contain both numeric and alphabetic characters. -Alternatively, the passwords/passphrases must have complexity and strength at least equivalent to the parameters specified above.	●	It is recommended to integrate NSX-T with VMware Identity Manager connected to one of the supported AAA systems. Through this integration, the payment entity may be able to fully comply with this requirement as it pertains to authentication and authorization for NSX-T. The local CLI admin account password complexity rules require: <ul style="list-style-type: none"> • At least eight characters in length • At least one upper case character • At least one lower case character • At least one numeric character • At least one special character
8.2.4	Change user passwords/passphrases at least once every 90 days.	●	It is recommended to integrate NSX-T with VMware Identity Manager connected to one of the supported AAA systems. Through this integration, the payment entity may be able to fully comply with this requirement as it pertains to authentication and authorization for NSX-T. The local CLI admin account does not have a requirement to modify the password every 90 days.

REQ ID	REQUIREMENT TEXT	SCOPE	FINDINGS
8.2.5	Do not allow an individual to submit a new password/passphrase that is the same as any of the last four passwords/passphrases he or she has used.	●	<p>It is recommended to integrate NSX-T with VMware Identity Manager connected to one of the supported AAA systems. Through this integration, the payment entity may be able to fully comply with this requirement as it pertains to authentication and authorization for NSX-T.</p> <p>The local CLI admin account management does not support password history restrictions.</p>
8.2.6	Set passwords/passphrases for first-time use and upon reset to a unique value for each user, and change immediately after the first use.	●	<p>It is recommended to integrate NSX-T with VMware Identity Manager connected to one of the supported AAA systems. Through this integration, the payment entity may be able to fully comply with this requirement as it pertains to authentication and authorization for NSX-T.</p> <p>It is recommended to set the CLI admin password on first access.</p>
8.3	<p>Secure all individual non-console administrative access and all remote access to the CDE using multi-factor authentication.</p> <p>Note: Multi-factor authentication requires that a minimum of two of the three authentication methods (see Requirement 8.2 for descriptions of authentication methods) be used for authentication. Using one factor twice (for example, using two separate passwords) is not considered multi-factor authentication.</p>	●	<p>It is recommended to integrate NSX-T with VMware Identity Manager connected to one of the supported AAA systems. Through this integration, the payment entity may be able to fully comply with this requirement as it pertains to authentication and authorization for NSX-T.</p>
8.3.1	<p>Incorporate multi-factor authentication for all non-console access into the CDE for personnel with administrative access.</p> <p>Note: This requirement is a best practice until January 31, 2018, after which it becomes a requirement.</p>	●	<p>It is recommended to integrate NSX-T with VMware Identity Manager connected to one of the supported AAA systems and using an additional factor for authentication. Through this integration, the payment entity may be able to fully comply with this requirement as it pertains to authentication and authorization for NSX-T.</p>

REQ ID	REQUIREMENT TEXT	SCOPE	FINDINGS
8.3.2	Incorporate multi-factor authentication for all remote network access (both user and administrator, and including third-party access for support or maintenance) originating from outside the entity's network.	●	It is recommended to integrate NSX-T with VMware Identity Manager connected to one of the supported AAA systems and using an additional factor for authentication. Through this integration, the payment entity may be able to fully comply with this requirement as it pertains to authentication and authorization for NSX-T.
10.1	Implement audit trails to link all access to system components to each individual user.	●	NSX-T is capable of being configured to log all network traffic that passes through the controlled network. NSX-T also logs administrative access to its console and creates a record of changes that occur during administrative sessions. It is recommended to integrate NSX-T with VMware Identity Manager connected to one of the supported AAA systems.
10.2.1	All individual user accesses to cardholder data.	●	Typically access to cardholder data would be logged through the payment application responsible for processing, storing and transmitting CHD. However, network logs can possibly be used to recreate events pertaining to user accesses to CHD or CHD enclaves. The logs collected by NSX-T could be part of the complete picture. If used for this purpose, they should be combined together with application and system logs pertaining to the relevant VMs, containers, applications, and so forth related to the individual's access to the CHD. It may be possible to correlate these events using a separate SIEM solution.
10.2.2	All actions taken by any individual with root or administrative privileges.	●	Actions taken with root or administrative privilege through NSX-T direct console access, web consoles, or via SSH, or API call to components of NSX-T are logged.
10.2.3	Access to all audit trails.	●	Pertaining to access to audit trails generated by NSX-T, access to such audit trails within the confines of NSX-T are logged.

REQ ID	REQUIREMENT TEXT	SCOPE	FINDINGS
10.2.4	Invalid logical access attempts.	●	<p>Since the logical access mechanism pertains to the access of NSX-T console or components using local accounts, invalid logon attempts are logged. Outside of this, invalid logical access attempts through the network to NSX-T controlled devices are logged.</p> <p>It is recommended to integrate NSX-T with VMware Identity Manager connected to one of the supported AAA systems. Through this integration, the payment entity may be able to fully comply with this requirement as it pertains to authentication and authorization for NSX-T.</p>
10.2.5	Use of and changes to identification and authentication mechanisms—including but not limited to creation of new accounts and elevation of privileges—and all changes, additions, or deletions to accounts with root or administrative privileges.	●	<p>The local admin account cannot be deleted nor can new accounts be added. Modification of the CLI admin account is limited to password changes. The use of the CLI admin account for the purpose of making modifications including changes to the password of the admin account should be logged.</p> <p>It is recommended to integrate NSX-T with VMware Identity Manager connected to one of the supported AAA systems. Through this integration, the payment entity may be able to fully comply with this requirement as it pertains to authentication and authorization for NSX-T.</p>
10.2.6	Initialization, stopping, or pausing of the audit logs.	●	Attempts to stop, pause or impede the audit trail generation of NSX-T are logged. These would typically occur with a larger failure of the NSX-T appliances.
10.2.7	Creation and deletion of system-level objects.	●	Creation and deletion of system-level object events are logged.
10.3	Record at least the following audit trail entries for all system components for each event:	●	Audit trail entries contain the following:
10.3.1	User identification.	●	User identification.
10.3.2	Type of event.	●	Type of event.
10.3.3	Date and time.	●	Data and time of event.

REQ ID	REQUIREMENT TEXT	SCOPE	FINDINGS
10.3.4	Success or failure indication.	●	Success or failure of the event.
10.3.5	Origination of event.	●	Origination of the event.
10.3.6	Identity or name of affected data, system component, or resource.	●	The identity of the affected system.
10.4	Using time-synchronization technology, synchronize all critical system clocks and times and ensure that the following is implemented for acquiring, distributing, and storing time. <i>Note: One example of time synchronization technology is Network Time Protocol (NTP).</i>	●	To support effective audit logging, NSX-T must be able to use time synchronization technology to ensure that system clocks used by NSX-T are consistent with all other organizational devices and elements. An NTP server can be specified on the NSX-T appliances through the CLI or API.
10.4.2	Time data is protected.	●	Time data is protected inasmuch as changes in NTP settings or clock settings are logged. Access to make changes to time data is restricted in NSX-T through role-based access controls.
10.4.3	Time settings are received from industry-accepted time sources.	●	NSX-T can be configured to synchronize time with an NTP server.
10.5	Secure audit trails so they cannot be altered.	◐	It is recommended to use a solution outside of the NSX-T such as a log aggregator or security information and event management (SIEM) solution. Through configuration of NSX-T to send logs to a log aggregation solution, the log aggregation solution can provide the means to secure the audit trail in a way that prevents alteration.
10.5.1	Limit viewing of audit trails to those with a job-related need.	◐	Role-based access controls with feature specificity for assignment of roles can be used to limit the ability to review NSX-T logs from the NSX-T console. It is recommended, however, to use a third-party log aggregation utility and/or SIEM to support audit trail requirements of PCI DSS.
10.5.2	Protect audit trail files from unauthorized modifications.	◐	It is recommended to use a third-party log aggregation utility and/or SIEM solution to support audit trail requirements of PCI DSS. NSX-T can be configured to send logs to a logging server.

REQ ID	REQUIREMENT TEXT	SCOPE	FINDINGS
10.5.3	Promptly back up audit trail files to a centralized log server or media that is difficult to alter.	●	It is recommended to configure NSX-T to send audit logs to a log aggregation tool or SIEM to fully support the audit trail requirements of PCI DSS. NSX-T can be configured to send logs to an external log aggregation device.
10.5.4	Write logs for external-facing technologies onto a secure, centralized, internal log server or media device.	●	It is recommended to configure NSX-T to send audit logs to a log aggregation tool or SIEM. NSX-T can be configured to send logs to an external centralized log aggregation device or service or SIEM solution.
10.7	Retain audit trail history for at least one year, with a minimum of three months immediately available for analysis (for example, online, archived, or restorable from backup).	●	It is recommended to configure NSX-T to send audit logs to a log aggregation tool or SIEM that has retention capabilities. NSX-T is capable of shipping logs to an external log aggregation service or SIEM solution. NSX-T does not inherently fully support audit trail history requirements.

Table 2: NSX-T PCI DSS 3.2 Applicability

CONCLUSION

BENCHMARK TEST SUMMARY

Coalfire's test results can be summarized as follows:

1. NSX-T could provide sufficient boundary protections for pods within a K8s namespace.
 - a. Attempts to discover adjacent "web" pods on the cde K8s namespace, from the Kali Linux box on the same namespace, were unsuccessful when the NSX-T DFW was deployed with a deny-all rule applied to the policy section.
 - b. Ports authorized by policy were discoverable as expected between tiers of the CDE micro-service.
2. As the non-CDE namespace was not defined in the CDE policy section, CDE pods and VMs were not successfully discovered by the Kali Linux container in the non-CDE namespace.
3. NSX-T successfully provided firewall controls for communications between the pod environment and the VM environment. The CDE database VM was not discoverable from the source in the non-CDE. The CDE database VM and only available ports per policy definition were discoverable as expected from pods in the CDE tier.
4. K8s labels could be utilized to define and dynamically populate membership of NSX-T NSGroups. As pods were scaled using the service definitions provided, the NSGroup dynamically added the new pods. Policies applied to the NSGroup were automatically and immediately applied to the new pods.

Coalfire also tested NSX-T Spoofguard by utilizing an ARP poisoning attack from the Kali Linux machine to provide man-in-the-middle access to targeted pods and VMs. By policy, NSX-T logical switch ports have Spoofguard applied by default. With Spoofguard disabled, Coalfire could successfully launch the ARP

poisoning attack. With Spoofguard enabled, the ARP poisoning attack was not successful and logs in NSX-T were indicative of the attack being blocked.

PCI DSS 3.2 COMPLIANCE APPLICABILITY SUMMARY

Coalfire concludes that the function and features of NSX-T can be useful in supporting a PCI DSS 3.2 program of compliance. When combined with the payment entity's governance, risk management, and compliance program, the capabilities of NSX-T may, when properly deployed, enable and support PCI DSS 3.2 recommendations in support of implementation of network segmentation to reduce scope of PCI DSS assessment to isolated CDE networks and devices. Additionally, NSX-T may be able to be deployed and used to support PCI DSS 3.2 requirements when aligned with a payment entity's documented standards. When combined with other technical solutions and payment entity processes and procedures, NSX-T can meet requirements pertaining to authentication and authorization and audit logging. Beyond this, NSX-T may be useful for improving overall system security by supporting a Zero Trust IT model of security.

ABOUT THE AUTHORS

Jason Macallister | Author | Senior Consultant, Cyber Engineering, Coalfire Systems
Mr. Macallister consults on information security and regulatory compliance topics as they relate to advanced infrastructure and emerging products and solutions.

Chris Krueger | Contributor | Principal, Cyber Engineering, Coalfire Systems
As Principal, Mr. Krueger contributes as an author and thought leader on information security and regulatory compliance topics for Coalfire's clientele in the "new and emerging" technical areas.

Justin Angel | Pen Test Consultant | Senior Consultant, Coalfire Labs, Coalfire Systems
Mr. Angel provided essential consultation on Metasploit Framework vulnerability selections and consulted on the design and delivery of the exploited design patterns used in this publication.

Wade Holmes | VMware Project Consultant | Senior Manager, Technical Product Management NSBU, VMware
Mr. Holmes is the NSX Technical Product Manager with responsibility for network and security thought leadership.

Published November 2017

ABOUT COALFIRE

Coalfire is the cybersecurity advisor that helps private and public-sector organizations avert threats, close gaps, and effectively manage risk. By providing independent and tailored advice, assessments, technical testing, and cyber engineering services, we help clients develop scalable programs that improve their security posture, achieve their business objectives, and fuel their continued success. Coalfire has been a cybersecurity thought leader for more than 16 years and has offices throughout the United States and Europe. Coalfire.com

Copyright © 2014-2017 Coalfire Systems, Inc. All Rights Reserved. Coalfire is solely responsible for the contents of this document as of the date of publication. The contents of this document are subject to change at any time based on revisions to the applicable regulations and standards (HIPAA, PCI-DSS et.al). Consequently, any forward-looking statements are not predictions and are subject to change without notice. While Coalfire has endeavored to ensure that the information contained in this document has been obtained from reliable sources, there may be regulatory, compliance, or other reasons that prevent us from doing so. Consequently, Coalfire is not responsible for any errors or omissions, or for the results obtained from the use of this information. Coalfire reserves the right to revise any or all this document to reflect an accurate representation of the content relative to the current technology landscape. To maintain contextual accuracy of this document, all references to this document must explicitly reference the entirety of the document inclusive of the title and publication date; neither party will publish a press release referring to the other party or excerpting highlights from the document without prior written approval of the other party. If you have questions about any legal or compliance matters referenced herein you should consult legal counsel, your security advisor and/or your relevant standard authority.