NSX-T Security Reference Guide

Version 1.3:
June 15, 2021

NSX-T Security Reference Guide 1
1 Introduction 6
1.1 VMware Security 6
1.2 NSX Service-defined Firewall 6
1.3 Datacenter Security Layout/Challenges/Solution 8
2 NSX Use cases/Customer journey/ Deployment options 11
2.1 NSX Security Use Cases 11
2.1.1 Segmentation 11
2.1.2 Compliance 14
2.1.3 Advanced Threat Prevention (ATP) 16
2.1.4 Virtual Patching 17
2.1.5 Secure Virtual Desktop Infrastructure (VDI) 19
2.1.6 Consistent Security – Global/Region/Multi-site/DR 20
2.1.7 Consistent Security – VM/Container/Physical Server/Cloud 22
2.1.8 Simplified DMZ Security 24
2.2 Modern Security Journey 26
2.2.1 Segmentation 26
2.2.2 Security Growing Up 27
2.2.3 Application Focused Security 27
2.2.4 Security Through Migration 27
2.3 NSX firewall – For All Deployment Options 28
2.3.1 NSX distributed firewalling for VM & physical server 29
2.3.2 NSX gateway firewalling for VM & physical server 30
3 NSX-T Architecture Components 31
3.1 Management Plane and Control Plane 32
3.1.1 Management Plane 32
3.1.2 Control Plane 33
3.1.3 NSX Manager Appliance 33
3.2 Data Plane 33
3.2.1 ESXi Data Plane 34
3.2.2 KVM Data Plane 35
3.2.3 Physical Servers 36
3.2.4 Distributed Firewall For Containers 37
3.2.5 NSX Firewalling for Public Clouds 38
3.3 NSX-T Consumption Model 40
3.3.1 NSX-T Role Based Access Control 40

3.3.2 NSX-T Declarative API Framework 41
4 Virtual Firewalling 44
4.1 NSX Firewalling: A New Approach 44
4.2 Gateway Firewall 46
4.2.1 Zone Firewalling with the Gateway Firewall 46
4.2.2 Gateway Firewall Functions 49
4.3 Distributed Firewall 50
4.3.1 Zone Firewalling with the Distributed Firewall 52
5 NSX Firewall Policy Building 54
5.1 Rule Lookup 54
5.2 Applied To Field 55
5.3 Scale 58
5.4 Grouping 59
5.5 Tags 61
5.5.1 Tags With/Without Scope 62
5.5.2 Multiple Tags vs Combined Tags 63
5.5.3 Tagging VM vs Segment vs Segment-port 63
5.5.4 Tags in Automation 64
5.6 NSX-T Policy Structure 64
5.6.1 Gateway Firewall Policy Categories 65
5.6.2 Distributed Firewall Policy Categories 67
5.6.3 Firewall Policy Drafts 70
5.6.4 Exclusion List 70
5.6.5 Statistics 71
5.6.6 Logs 71
5.7 Security Profiles 72
5.7.1 Session Timers 72
5.7.2 Flood Protection 74
5.7.3 DNS Security 75
5.8 Policy Automation with vRNI 76
5.8.1 Discovery with vRNI 76
6 Container Security 80
6.1 NCP Components 81
6.2 Tanzu Application Service 86
6.3 OpenShift 87
6.4 NCP Features 88
6.4.1 Visibility 88

6.4.2 IPv6 89
6.5 Project Antrea 89
7 Firewall features 90
7.1 URL Analysis 90
7.2 Service Insertion and Service Chaining 91
7.2.1 North-South Service Insertion 92
7.2.2 East West Service Insertion and Service Chaining 93
7.3 NSX Endpoint Protection – Guest Introspection 97
7.3.1 NSX Endpoint Protection – Architecture and Components 98
7.3.2 User Interface/REST API 101
7.3.3 Management Plane Components 101
7.3.4 Control Plane Components 101
7.3.5 Data Plane 101
7.3.6 Partner Components 103
7.3.7 Workflow Object Definitions 103
7.3.8 NSX-T Endpoint Protection Deployment and Enforcement 104
7.3.9 NSX-T Endpoint Protection Design Considerations 105
7.3.10 Endpoint Protection Workflow: Registration, Deployment, and Consumption 109
7.3.11 Partner Supportability 111
8 Intrusion Detection and Prevention 112
8.1 NSX IPS Components 113
8.2 IPS Signatures 115
8.3 Profiles 115
8.4 IPS Rules 116
8.5 IPS Use Cases 119
8.5.1 IPS Use Case: Compliance 120
8.5.2 IPS Use Case: Creating Zones 120
8.5.3 IPS Use Case: Appliance Replacement 121
8.5.4 IPS Use Case: Detecting Lateral Threats 122
9 Federation 123
9.1 Managers in Federation 124
9.2 Groups 126
10 Management and Operations 128
10.1 vRealize Network Insight (vRNI) 128
10.2 NSX Intelligence 131
10.3 SIEM 133
10.4 NSX Operations 134

Appendix.....140

1 Introduction

1.1 *VMware Security*

At VMware, security is the mindset that continually strives to visualize the multiple layers of threats, vulnerabilities, and weaknesses, that could be leveraged by an attacker to gain a foothold. Ever-changing attack surface (workload and adjacency of threat) is very real and unique to modern datacenter/cloud environment. The fundamental value of VMware security solution is to shrink the attack surface and preventing the proliferation of the threat that goes undetected. Security is a multifaceted effort. One product or practice alone does not make answer the call for security. Security must be done with layers of practices supported by products in answer to business needs. IN answer to this need for multi-faceted security, VMware has provided security hardening guides as well as security products and features across its entire product portfolio. VMware sees security as an adjective, not a noun. Security is built-in; not bolted on.

VMware has a broad offering of security products and features across the heterogeneous infrastructure which is common today. Infrastructure today extends along a continuum from physical servers on prem to VMs in hypervisors (sometimes a variety of hypervisors like ESXi and KVM) to containers, on prem and in the cloud, to Software as a Service (SaaS) offerings like Office365 (O365) and SalesForce (SFDC). VMware offers the tools to secure this heterogeneous environment in a consistent manner, while allowing the qualities of each solution to shine. (For a listing of these products and features, see the Appendix)

This vast offering of products and features allows for pervasive and granular security policy definition from endpoints to servers to containers to microservices. It also allows for encrypting data both in flight and at rest. Finally, this also allows for the detection of suspicious behaviors on endpoints or in the network across a heterogeneous environment.

This document will focus on the security offerings of the NSX product portfolio and how to optimally design and use those offerings to achieve desired security objectives.

1.2 *NSX Service-defined Firewall*

The NSX Service-defined Firewall is one of the foundations of VMware Security. This solution is a unique distributed, scale-out internal firewall that protects all East-West traffic across all workloads without network changes. This radically simplifies the security deployment model. It includes a distributed firewall, advanced threat protection, and network traffic analytics. With the VMware NSX Service-defined Firewall, security teams can protect their organizations from cyberattacks that make it past the traditional network perimeter and attempt to move laterally. NSX Service-defined Firewall's key differentiating capabilities include:

- **Distributed, granular enforcement:** The NSX Service-defined Firewall provides distributed and granular enforcement of security policies to deliver protection down to the workload level, eliminating the need for network changes.
- **Scalability and throughput:** Because of the distributed nature, the Service-defined Firewall is elastic, with the ability to auto-scale as workloads spin up or down.
- **Intra-application visibility**: The Service-defined Firewall automatically determines the communication patterns across all types of workloads, makes security policy recommendations based on those patterns, and checks that traffic flows to conform to deployed policies.

- **Declarative API:** With the NSX Service-defined Firewall, security teams can move at the speed of development to deliver a true public cloud experience on-premises.
- **Advanced Threat Prevention:** With the NSX Service-defined Firewall security teams can easily deploy advanced threat prevention capabilities such as distributed IDS/IPS, network sandboxing, and network traffic analysis/network detection and response (NTA/NDR) to protect against known and zero-day threats.

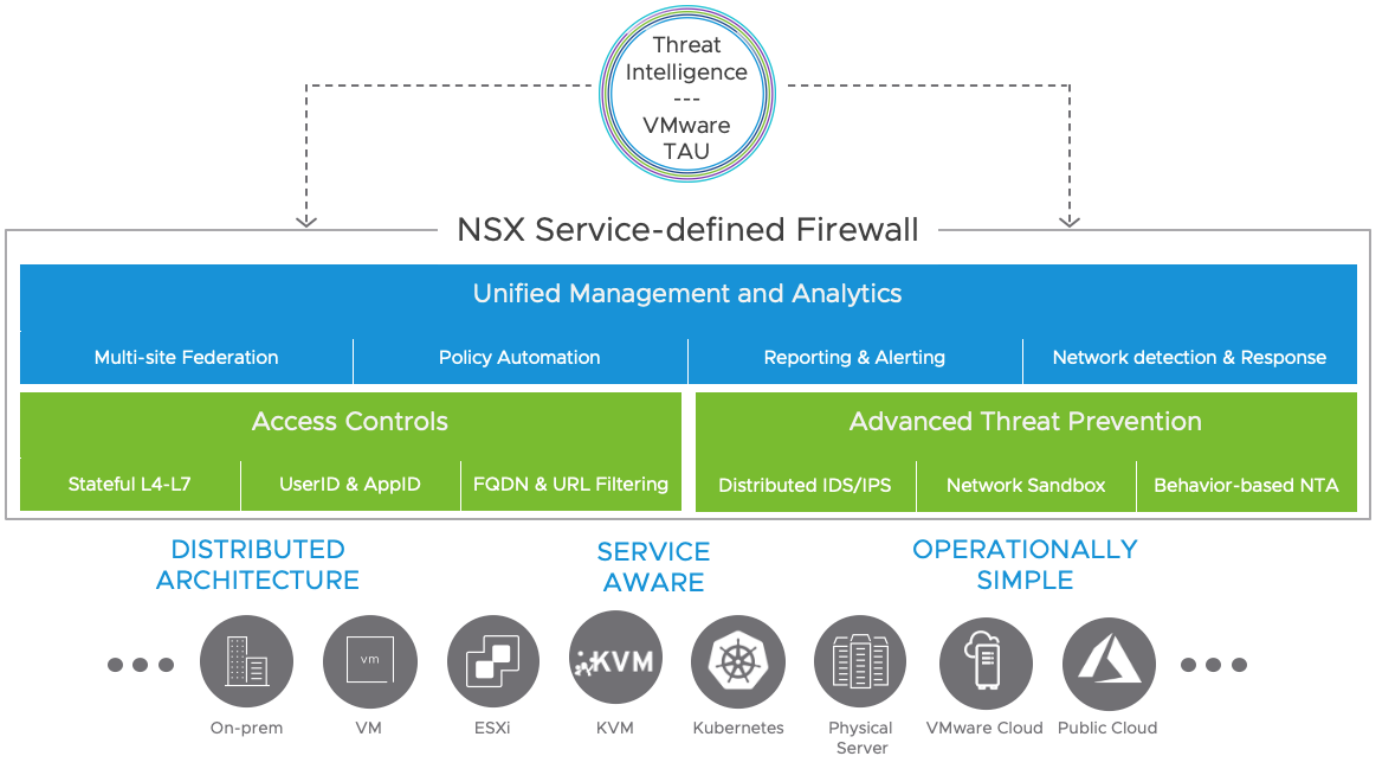

Figure 1-1: NSX Service-defined Firewall

With these capabilities, customers can deploy application workloads rapidly to get the speed and flexibility needed to quickly create and reconfigure virtual security zones by defining them entirely in software using declarative API. The NSX Service-defined Firewall also allows users to prevent lateral movement of attacks by extending East-West security with stateful Layer 7 firewalling, including App ID and User ID-based policies, as well as advanced threat protection. VMware's solution enables customers to meet regulatory requirements via its inspection of all traffic, which provides complete coverage to eliminate blind spots with a distributed IDS/IPS delivered in software. Finally, customers can easily create, enforce, and automatically manage granular micro-segmentation policies between applications, services, and workloads across multi-cloud environments to work towards a zero-trust security model.

Uniqueness of NSX Service-defined Firewall Architecture:

The NSX Service Defined Firewall architecture is unique and intrinsically built into the hypervisor at the VNIC level, with no additional firewall appliance or agents to manage. This allows NSX distributed firewalling and advanced threat prevention enforced for every flow at the VNIC level closer to the workload in a network-agnostic manner.

More details on the NSX Service-defined Firewall architecture and the advantages covered in following section and the use cases chapter.

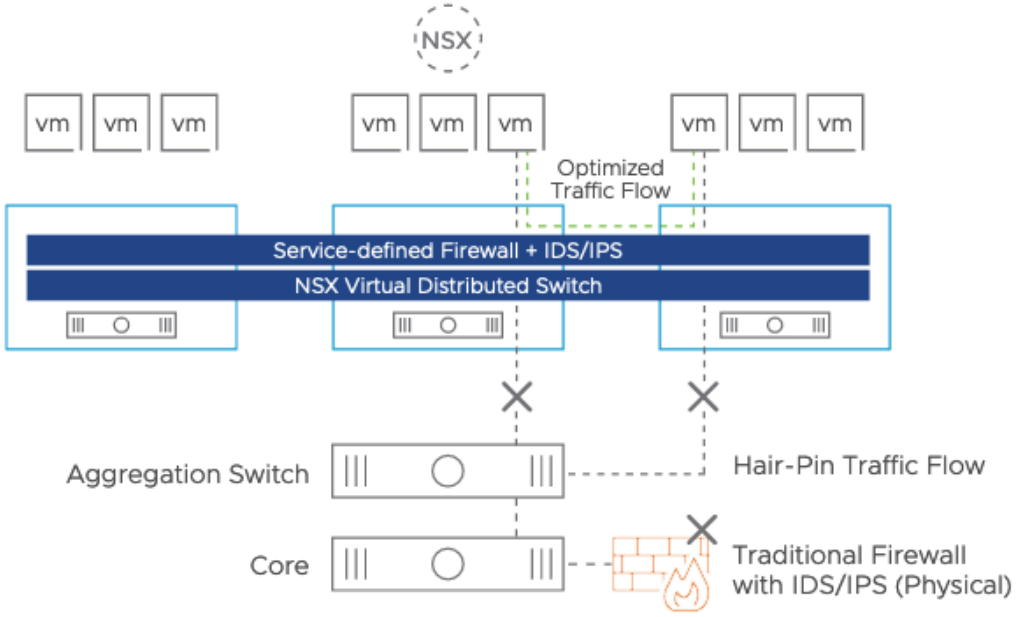

Figure 1-2: NSX Service-defined Firewall – Distributed Architecture

1.3 *Datacenter Security Layout/Challenges/Solution*

The traditional datacenter security approach relied primarily on perimeter defense—securing the north-south traffic, but assuming that East-West traffic in the data center was inherently safe. However, the growing number of sophisticated attacks on corporate information assets and data breaches have changed the organization's security mindset and requirements. Every organization should be working towards enhancing its enterprise security posture to a zero-trust model.

Even though the zero-trust security model for the data center is a very important and ambitious goal, at the same time, it is challenging because this needs to consider multiple factors: the scale of workloads, compliance requirements, heterogeneous application form-factor, multi-cloud/multi-site deployment, and more.

The figure below shows the typical data center security layout and classification framework typically considered to define the data center application's security posture. An organization can consist of multiple data centers, environments, security zones, business units, applications, platforms, and services. Each environment, security zone, and application have their security requirement based on the workloads hosted and how they are exposed to threats. Given every workload would have affiliation with an environment, zone, platform, and/or application, the zero-trust model needs to factor in all these variables and come up with a policy that intersects all the requirements and is easy to manage and operate.

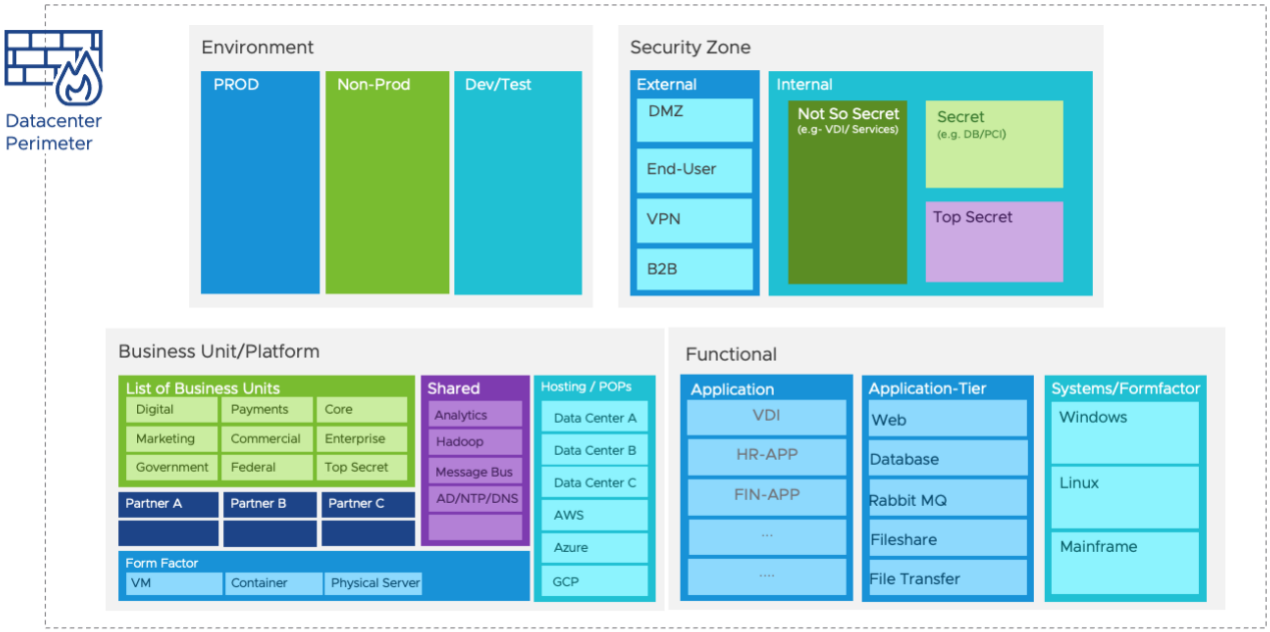

Figure 1-2: The datacenter security classification layout

In short, the zero-trust model for data center security is not a product or solution; it's a journey an organization needs to take. This journey needs to handle the complexity of the environments and assess the right security technology and platform to achieve the zero-trust security model. Most organizations take this journey in phases with a combination of the following approach: a fence around broader zones, security around most valuable assets like critical applications and databases, or most exposed application/resources to external threats or low hanging/easy ones to secure.
As you take the zero-trust model journey, you may be tempted to repurpose or reposition your existing physical appliance firewalls that were purpose-built for protecting the perimeter to solve for the data center's East-West security. This is not a good, wise or effective approach; in fact, traditional appliance firewalls do not help achieve the goal of the zero-trust security model because they will not see all traffic and because they often lack the context of the traffic they see. On the other hand, NSX Service-defined firewall, because it is integrated into the hypervisor and is distributed to every virtual NIC, sees all packets and has context around the applications and all the workloads using the applications, making it a more effective and vastly simpler means to help you achieve your goal of zero-trust. The following section highlights some of the key challenges with traditional appliance-based firewalls and how NSX distributed firewall removes those challenges that organizations face.

Traditional Firewall Appliances vs NSX Service-defined Firewall:

The traditional security approach has relied primarily on perimeter defense—securing the north-south traffic, but assuming that East-West traffic in the data center was inherently safe. The traditional firewalls are not built to address a new set of data center security challenges. Traditional appliance-based firewalls cannot provide the least-privileged access model to the data center application and have the following challenges to deal with:

- **Network Topology Dependency:** Traditional physical appliance firewalls have network topology dependency, so firewalling can be done only at the network boundary and for North-South traffic, not for East-West traffic.
- **Hair Pinning of Traffic:** For firewall enforcement, traffic needs to be hair pinned to the centrally hosted traditional firewall/IPS appliances. This makes the appliance firewall a

network chokepoint, and it adds to the application latency and unnecessary use of network bandwidth.

- **Blind Spots:** Traditional model has blind spots at many levels. Starting with East-West traffic, the legacy approach cannot see intra-host or intra-VLAN traffic. Then there is the challenge of vendor software backdoor (analytics, support, collection) legacy end-of-support OS. All of these flows do not have visibility or firewalling.
- **Unable to dynamically scale:** Adding more applications or workloads could choke the physical firewall capacity. The solution is to upgrade current appliances or add newer appliances to accommodate the growing need of business and datacenter. More endpoint, cost, cabling, power, cooling etc
- **Only broader Segmentation:** Only possible to do broader network segmentation without having option to do granular application and micro-segmentation, which is needed to protect organizations from East-West lateral movement within the datacenter.
- **Static Policies:** Only allows to define security policy based on IP or gateway Interface, no dynamic workload context-based policy, which is needed for modern datacenters.

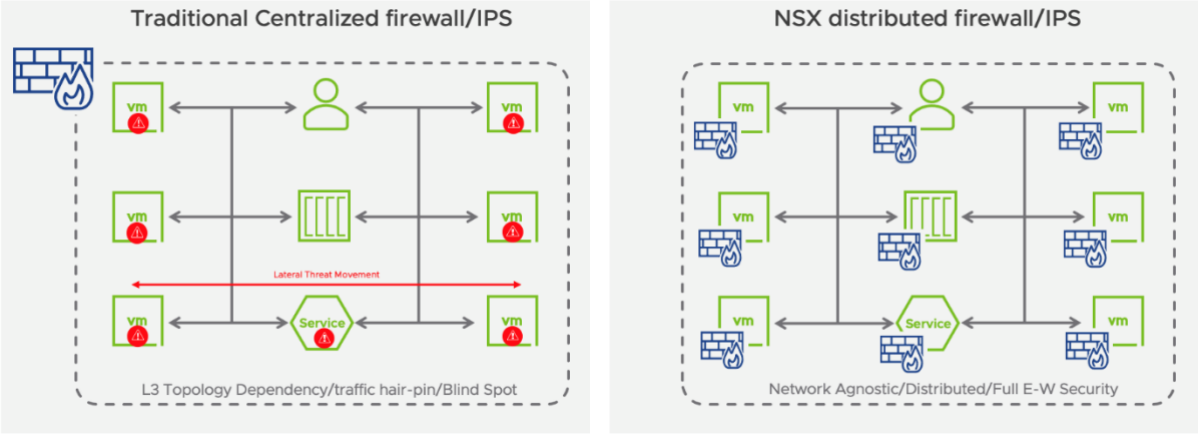

Figure 1-3: Traditional Appliance Firewall vs NSX distributed firewall

On the other hand, the NSX firewall removes all these challenges and trade-offs organizations need to do with the traditional appliance model. NSX firewall is purpose-built for data center security and built into the infrastructure to provide macro and micro-segmentation policies. NSX firewall architecture enables to provide zero-trust model to organizations datacenter

- **Network Topology Agnostic:** NSX firewall is built into hypervisor kernel. Each workload would have its firewall and IPS instance. Easy to insert without having to change any logical or physical topology.
- **Distributed architecture:** NSX firewall distributed architecture inspects traffic at the source, so removes the need to hair-pin traffic to traditional centralized appliances and reduce network congestion.
- **Complete Visibility/Security:** NSX distributed firewall inspects every flow from every workload within your datacenter. So, customers have complete security and visibility into all the application traffic.
- **Elastic throughput:** No more capacity planning required with NSX. NSX firewalling/IPS capacity dynamically scales linearly as you add more compute to accommodate the organization's growth.
- **Context-Aware Policies:** The NSX firewall provides context-aware dynamic macro and micro-segmentation policies using a single pane of glass. NSX policies can be totally decoupled

from network/infrastructure (IPv4 or IPv6 or VLAN) as NSX allows policy based on Tag/Object names.

- **Granular segmentation:** NSX designed to provide more granular application-segmentation & micro-segmentation, in addition to traditional more broader network segmentation.
- **Automated policy lifecycle management:** The NSX policy model enables the automatic creation of security policies for new workloads and the tear down of old policies when workloads are decommissioned. Security policies remain consistent with deployed workloads, preventing the accumulation of stale policies, a common challenge with traditional network security appliances.
- **Policy and state mobility:** When workloads move, the policies and the state move with the workload. Workloads are automatically secured at their new location without manual configuration or dropped flows.
- **Cloud-Native ready:** NSX firewall is built to protect all kinds of workloads: Virtual Machine, Physical Server, Public Cloud instance, and Container microservices.

Many of our customers have already embarked on this journey using the NSX firewall successfully. The following chapter talks about NSX use cases & NSX customer success stories to help you start the journey and successfully achieve your goal of a zero-trust model with NSX Service-defined Firewall.

2 NSX Use cases/Customer journey/ Deployment options

This chapter covers NSX security use cases, customer journey and NSX deployment options for different data center scenarios.

2.1 *NSX Security Use Cases*

This section will look at following different use cases for NSX Service-defined Firewall:

1. Segmentation
2. Compliance
3. Advanced Threat Prevention (ATP)
4. Virtual Patching
5. Secure Virtual Desktop Infrastructure (VDI)
6. Consistent Security – Global/Region/Multi-site/DR
7. Consistent Security – VM/Container/Physical Server/Cloud
8. Simplified DMZ Security

In each of these cases, NSX brings a unique set of functionalities which addresses the challenges with legacy infrastructure is unable to. All of the use cases inherit the key value of NSX Service-defined Firewall architecture discussed in earlier chapter: Single pane of Management, Context-Aware Tag/Object based policies, Network Topology Agnostic, Distributed architecture, Complete Visibility/Security, Elastic throughput.

2.1.1 Segmentation

2.1.1.1 What is the requirement?

Every major breech in the last two decades have been examples of land and expand: the hackers have leveraged access to the weakest link as a toehold into the rest of the infrastructure. Segmentation covers the case where there is a desire to create a smaller scatter area in the case of a breach. By segmenting, any compromised endpoint will have less access to other endpoints, even if credentials are compromised. Segmentation can be Zone segmentation (separating the environment in half or thirds - production and non-production , for example), VLAN segmentation, application segmentation or micro-segmentation (where each endpoint is segmented).

With the legacy approach using physical firewalls, segmentation was limited to Zone and VLANs. Any packet or flow that needed to be inspected for access had to be directed to the firewall appliance which would then determine access based on the ruleset provided. This model of segmentation has several limitations:

- Lack of segmentation flexibility
- Traffic must be directed to the firewall (complexity)
- Lack of scalability (high cost)
- Blindspots (less effective)

The lack of flexibility is challenging because often applications span VLANs, with any given VLAN containing more than 1 application. Because there is no means of segmenting in a more granular manner than VLAN, this would mean that 2 two endpoints on the same VLAN would not be isolated from each other – thus limiting the size of the security domain. Because traffic must be directed to the firewall, careful traffic engineering is required to avoid firewalls being routed around. Moreover, any change in segmentation requires you to reengineer the network and change the IP address of the applications.

Modern infrastructure requires the ability to keep up with modern application creation, updates, and deletion. Legacy firewalls were never designed for dynamic environments.

2.1.1.2 Why NSX?

By using NSX-T DFW, it is possible to segment in any matter desired. There are four basic types of segmentation, many of which will coexist – each applied in different sections of the environment:

- Zone Segmentation
- VLAN Segmentation
- Application Segmentation
- Micro-segmentation

Zone Segmentation may be as general as segmenting production from non-production, or it may be a far more detailed segmentation by business unit, function, or product offering. The point is that each zone is defined independently of segments, VLANs, datacenters, or other constructs. Zones are entirely logical definitions which can be used to define security policy.

VLAN segmentation is most commonly used by customers replacing their legacy firewall infrastructure. In this model, an IP segment is the defining element for a source or destination of the security policy.

Application segmentation is used to define a logical security ring around an application. Because applications are not frequently understood in detail, it may be convenient to simply define a tag for a given application and apply this tag to all of its components and allow full communication between

said elements. This brings greater security than a large zone definition which may be multiple applications, without requiring the detailed understanding need for micro-segmentation.

Micro-segmentation is a security model where communication between elements are defined as explicitly as possible. At its extreme, micro-segmentation would be the explicit definition of communication between pairwise elements. Clearly this is operationally complex, thus NSX offers micro-segmentation based on tags which allows explicit definition by groups. For example, one may define a rule which allows SSL but only TLS version 1.3 to my tagged Secure Web servers.

In any enterprise environment, the fact is that there will be a desire to segment in each of those manners in different areas. With NSX, all of these segmentation approaches are not exclusive, but can coexist. One may decide to segment a lab in a zone model by just setting up a boundary around it and a DMZ environment in a micro-segmentation. Non Prod applications may be segmented just by applications whereas Prod Applications containing sensitive customer data may be segmented further maybe VLAN. The value proposition of NSX in segmentation as that it accommodates all segmentation strategies so that they may coexist. Furthermore, the change of one security model to another is accomplished through a simple policy push, without the need to reIP or rearchitecting any networking infrastructure. As described above, legacy firewalls cannot go any further in segmentation that VLAN segmentation due to architectural limitations.

2.1.1.3 How to take a journey with NSX

NSX firewall allows organizations to achieve the least privileged access model using segmentation in phases, starting broader network/zone segmentation to more granular application-segmentation and micro-segmentation, using distributed firewalling/IPS capabilities. The idea here is to reduce the attack surface progressively in phases.

Phase-1: Zone Segmentation:

Start with broader network segmentation by creating virtual zones to divide the data center into smaller zone and have a security fence around them. Define necessary NSX firewalling/IPS policy based on the organization's zonal security requirements. For example, NSX Tag/Object-based dynamic grouping can be leveraged to create DMZ, Prod, Non-Prod, or Services zone and use that zone group to define respective security controls/policies for inter-zone traffic. NSX allows defining zonal policy without needing a workload to be separated by a VLAN or network boundary. Customers can create a virtual zone by grouping virtual interfaces, using tag for the relevant workloads, into a zone and define relevant FW/IPS policies. The policy moves with the workload during vMotion or DR events, even if it has to be moved to a new network or with new IP address.

Phase-2: Application Segmentation:

The application segmentation provides the next step in achieving a zero-trust model to reduce the attack surface further. This phase builds a fence around an application. So that all workloads within an application can communicate; however, any outside communication is restricted by application-segmentation policy. Organizations typically have 100's of applications in different environments. One can start with few critical or easy ones to segment and begin building this security posture for all applications over time.

Phase-3: Micro-segmentation:

This is the final state any organization wants to be in to provide zero-trust model where only necessary traffic is allowed between any application/application-tiers/services. This is the challenging phase as one needs to understand ports and protocols for all applications. Like application segmentation, this also will be done in stages, starting with few applications and extending to all

applications over time. NSX Intelligence, vRealize Network Insight (VRNI) solution can help profile organizations applications at scale and achieve this phase faster.

All the segmentation phases discussed above inherit the advantage of NSX Service-defined Firewall architecture: Single pane of Management, Context-Aware Tag/Object based policies, Network Topology Agnostic, Distributed architecture, Complete Visibility/Security, Elastic throughput.

On the other hand, traditional appliance firewalls cannot provide segmentation beyond zone segmentation. That too inefficiently with challenges discussed in an earlier section like L3 topology dependency, hair-pinning of traffic, East-West blind-spot, chokepoint, and more.

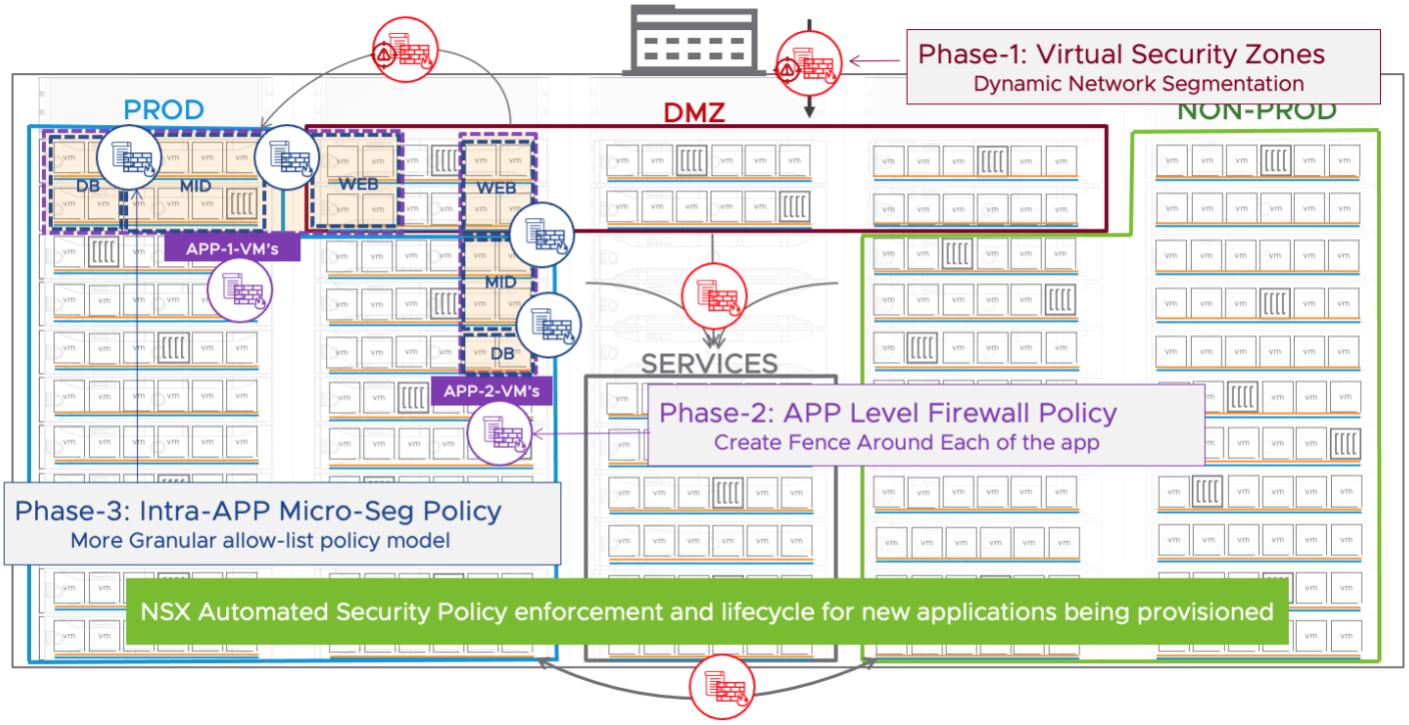

Figure 2-1: Segmentation In Phases with NSX

2.1.2 Compliance

2.1.2.1 What is the requirement?

One of the greater drivers of security architecture is compliance. Compliance mandates are varied in their level of prescriptiveness for architecture. There are models such as HIPAA which will merely fine based on the breach of information and there are those who prescribe the architecture such as PCI. Regardless of the mandate in question, there is a need to provide a security architecture and then to meet regular audits to ensure continued compliance across a dynamic security environment.

2.1.2.2 Why NSX?

NSX Service-defined Firewall helps organizations to meet regulatory compliance requirements such as the Health Insurance Portability and Accountability Act (HIPAA), the Payment Card Industry DataSecurity Standard (PCI DSS), and the Sarbanes-Oxley Act (SOX).

NSX distributed firewall architecture and micro-segmentation capabilities help propagate regulation-specific security policies to all relevant workloads and track traffic flows to and from sensitive applications. NSX firewalls also eliminate the need to buy and deploy discrete appliances to support compliance. In addition, organizations inherently would get the advantage of NSX Service-defined Firewall architecture: Single pane of Management, Context-Aware Tag/Object based policies, Network Topology Agnostic, Distributed architecture, Complete Visibility/Security, Elastic throughput.

For example, to meet the PCI compliance requirement, organizations can leverage the NSX firewall to define a virtual PCI zone and protect the zone using firewall and IPS security control, as mandated by the compliance. NSX allows this without rearchitecting the network topology and allowing every workload to have the firewall/IPS at the vnic level. Furthermore, NSX firewalling/IPS policies/profiles can be customized for the PCI workloads. This includes both zone segmentation as well as micro-segmentation to protect critical PCI workloads.

The value proposition extends beyond the NSX product family. With tools such as vRNI, there is a means to streamline audit requirements, translating to a tangible ROI for customers.

2.1.2.3 How to take a journey with NSX

The compliance brings many requirements, including segmentation and IPS policies based on the exposure to outside network or criticality of the application or service. For example, compliance may require stricter layer 7 firewalling with intrusion detection policies applied to external/DMZ zone which is exposed to internet. Similarly, PCI workloads needs to be fully isolated and protected with firewalling and IPS.

NSX helps organizations in achieving this compliance goal to define firewalling and advanced threat prevention policies at zone level, Application level or Micro-segmentation level. NSX further helps to customize the distributed firewall policies & IDS/IPS profiles based on the zone or workload type or severity of signature. The following example shows simple NSX IDS/IPS policy with customized profile for PCI and DMZ zone.

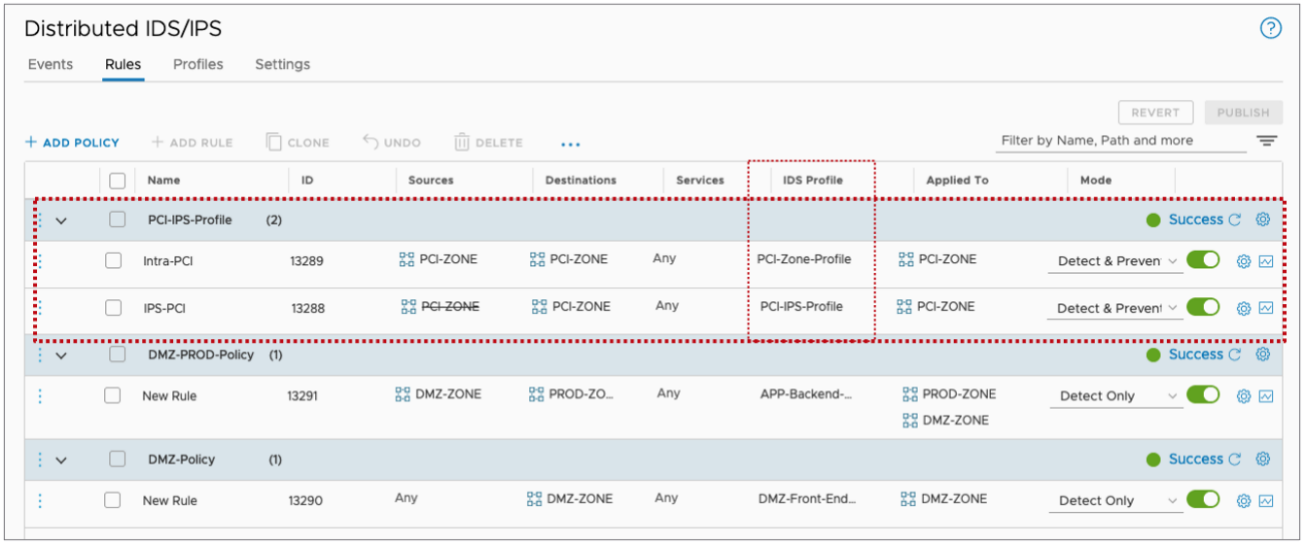

Figure 2-2: NSX compliance Policy

2.1.3 Advanced Threat Prevention (ATP)

2.1.3.1 What is the requirement?

With the rise of distributed applications and microservices, internal network traffic now dominates traditional north-south traffic. At the same time, the data center boundary has diffused with edge and cloud applications as well as with end-user devices. Modern-day attackers noticed these changes and learned to move laterally, aggressively, from their initial point of attack. As a result, inspecting internal East-West (server-to-server) traffic with an advanced threat detection capability is increasingly critical to securing workloads and enterprise data.

2.1.3.2 Why NSX?

NSX Distributed IDS/IPS is an application-aware traffic inspection engine purpose built for analyzing internal East-West traffic and detecting lateral threat movements. The engine runs within the hypervisor to optimize packet inspection. NSX Distributed IDS/IPS combines industry-leading signature sets, protocol decoders and anomaly detection-based mechanisms to hunt for known and unknown attacks in the traffic flow. It also benefits from rich application context, driving lower false positive rates while incurring minimal computational overhead on the host.

Key capabilities:

- **Distributed analysis** - The IDS/IPS engine is distributed out to each workload, eliminating blind spots while maintaining a simple operational model. The inspection capacity scales linearly with the number of workloads, eliminating the throughput constraints typically experienced with discrete appliances.
- **Curated, context-based signature distribution** - The management plane enables only the relevant threat signatures for evaluation at each workload based on knowledge of the running applications. This reduces computational overhead on the host and results in higher fidelity matches with lower false positive rates.
- **Application context-driven threat detection** - The IDS/IPS engine has definitive knowledge of applications running on each host, eliminating guesswork regarding the source or target application context. This knowledge allows for better alert classification and operator ability to prioritize alerts for further investigation.
- **Detects and prevents Lateral Threat Movement** - Distributed IPS front-ending every workload enables exploit-detection regardless of it being initial attack vector, lateral spread or exfiltration.
- **Policy and state mobility:** When workloads move, the policies and the state move with the workload. Workloads are automatically secured at their new location without manual configuration or dropped flows.
- **Real time Intrusion Detection dashboard with workload context:** Provides insight into threat detection with workload context, vulnerability an exploit trail.

2.1.3.3 How to take a journey with NSX

VMware NSX Distributed IDS/IPS provides security operators with a software-based IDS/IPS solution that enables them to achieve regulatory compliance, create virtual zones and detect and prevent lateral movement of threats on East-West traffic.

Security admin can leverage the NSX advanced threat detection and prevention capability in detect-only mode or prevent mode. In addition, NSX provides more granular control to inspect subset of

traffic allowed by distributed firewall policy for IPS/IDS. In addition, user can customize IDS/IPS signature profile and policy per application, workload context, that way only relevant signature are inspected. The context-aware profiles could be based on the application tier or based on hosted platform or based on intrusion severity of the signature. For example, IDS/IPS policy can be applied to External DMZ workloads with signature profiles relevant to front-end. Similarly backend specific profiles can be applied to backend-services or database services workloads. This helps in reducing false positive and helps threat analyst to focus on real threats.Please refer to NSX-T Intrusion detection and response chapter later in the guide for more details.

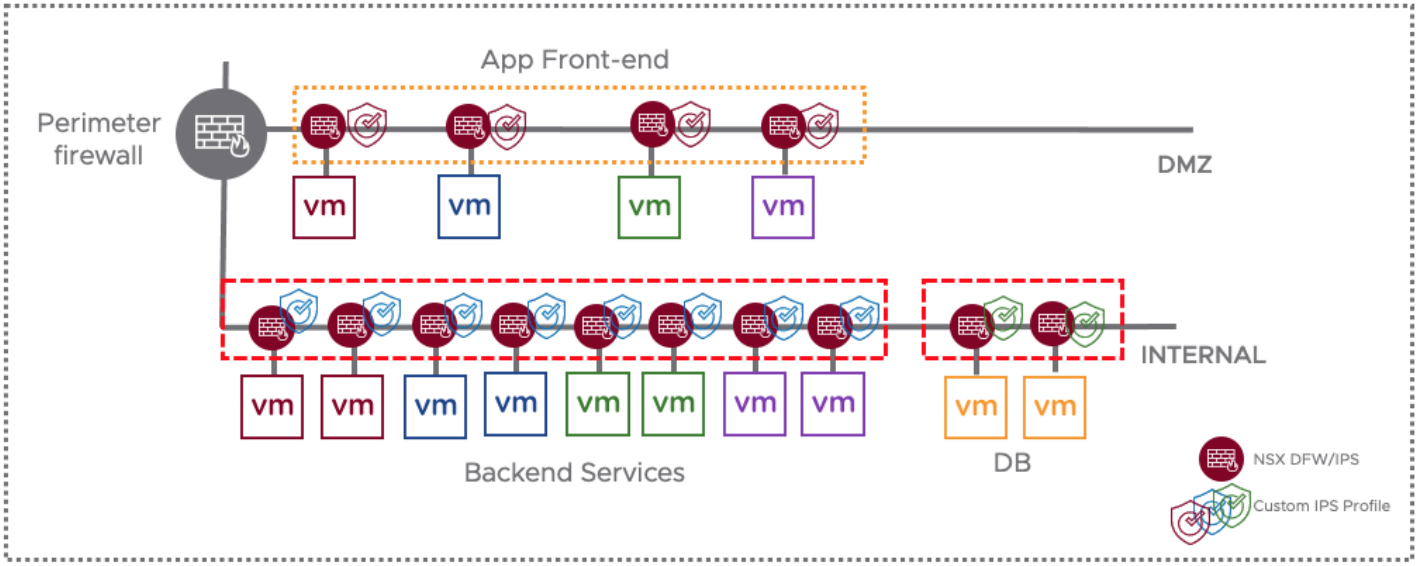

Figure 2-3: NSX Advanced Threat Prevention – IDS/IPS

2.1.4 Virtual Patching

2.1.4.1 What is the requirement?

Every organization is familiar with the challenge of vulnerability patch management for their application workloads. The patch update management could take a long time because of scale, frequency of new vulnerability & patch availability. The patching process involves many stages - availability of the patch from the vendor, testing patch in the development environment, knowing vulnerable workloads, scheduling window for the patch update & finally apply that patch on a production workload.

This delay in the patching process leaves the application open for exploitation, putting an organization in danger. The concept of virtual patching helps in protecting these vulnerable application/platforms from exploitation during this phase. Virtual patching is the workflow to virtually patch the vulnerable workload using a network security control by creating a policy to stop any exploitation attempts against that known vulnerability before the workload is patched with the actual patch.

2.1.4.2 Why NSX?

NSX sService-defined Firewall with its IDS/IPS capability is uniquely positioned to address the virtual patching use case because of following reasons:

- **Per workload IDS/IPS:** NSX IDS/IPS is distributed into the hypervisor and enforced at the workload virtual interface level closer to the workload being protected. Every single packet can be inspected against any targeted exploit against the workload being protected.
- **Vulnerability/Exploit specific IPS Profile:** NSX allows user to define and apply custom IDS/IPS profile relevant to the vulnerability and the exploit.
- **Simplicity at Scale:** NSX simplifies this as the virtual patching policy can be applied at scale across different environment based on the workload context.
- **Tag Based Policy:** Identify vulnerable workloads with tags and define dynamic policy using tag to provide virtual patching.

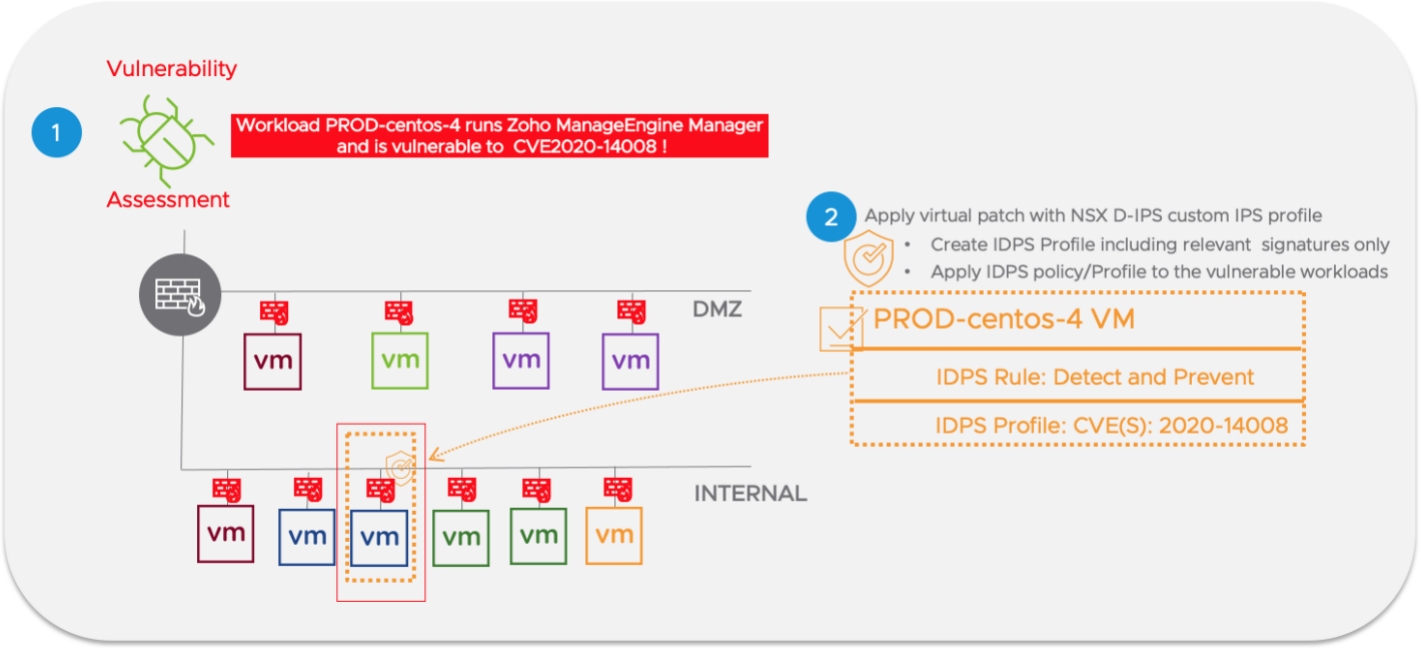

Figure 2-4: NSX Virtual patching – per workload

2.1.4.3 How to take a journey with NSX

NSX helps in protecting vulnerable host by providing more targeted IDS/IPS signature profile until workload is fully patched with the actual patch. For example, as part of vulnerability assessment organization security team found new or existing vulnerability on the version of application platform which is used – it could be Apache Struts framework, Windows, Linux platform. With NSX, user can create a CVE, Exploit, Product specific profile and policy to provide the protection against any attempt to exploit that vulnerability – until they are patched with actual patch. These vulnerable workloads can be grouped using Tags, OS name etc., that is used to define the scope of that policy to only those Web server, Windows server or Linux servers, this further reduces false positive as only relevant signatures are inspected against traffic to only relevant workloads.

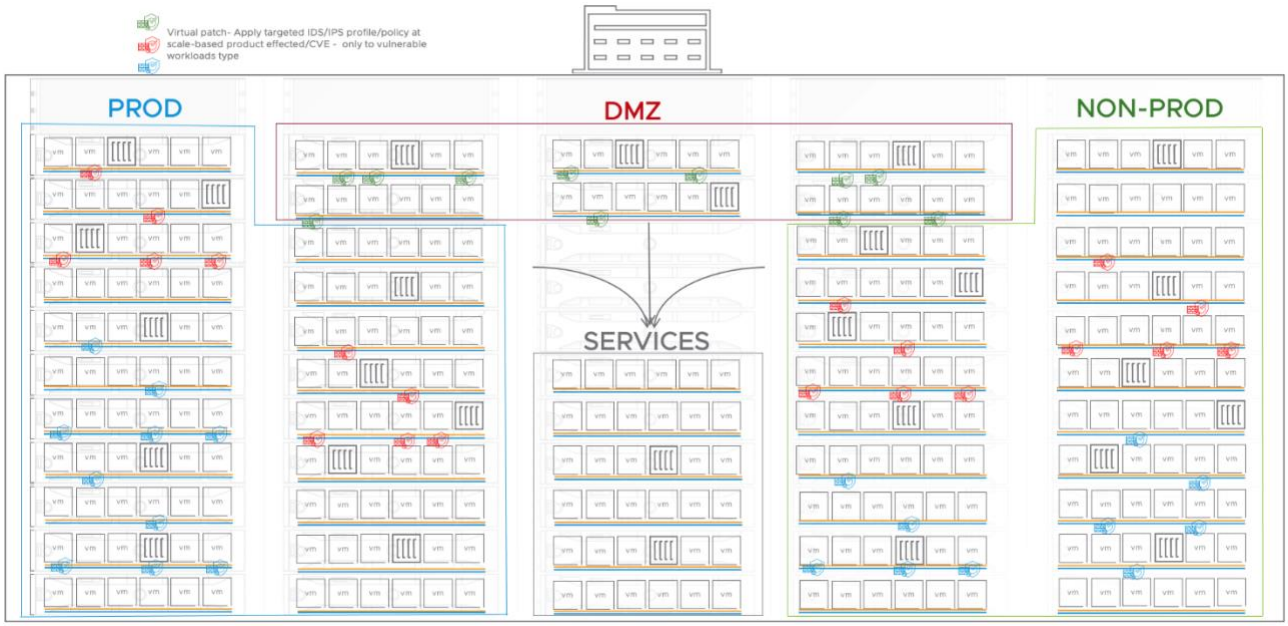

Figure 2-5: NSX Virtual patching at scale

2.1.5 Secure Virtual Desktop Infrastructure (VDI)

2.1.5.1 What is the requirement?

Virtual desktop infrastructure (VDI) solutions like VMware Horizon or Cirtrix XenDesktop enables centralized hosting of users' desktop sessions using either Remote Desktop Session Host (RDSH) or virtual desktop pools. The consolidation of end users' applications and data reduces infrastructure costs and improves manageability and data protection. However, since users' desktops are occasionally breached, their proximity to sensitive data center infrastructure presents a new threat. An attacker might take over a user desktop and use it to infiltrate nearby servers. Security teams must isolate virtual desktops and block lateral attacks. In order to provide VDI security, organizations need firewall which can define access policy based on identity of the user in the virtual desktop environment.

2.1.5.2 Why NSX?

Typically, users have different access rights to applications and resources based on their role (e.g., only the finance group can access financial systems). However, virtual desktop sessions share IP addresses between users, complicating enforcement of proper access rights using just IP addresses. The Service-defined Firewall's identity-based firewalling capability seamlessly integrates with Active Directory. Thus, admins can use the Service-defined Firewall to control user access to resources based on their Active Directory groups and identity.

Some of the key capabilities which will help with protecting VDI environment:

- **Rapid deployment with simplicity:** Using just two NSX-T tags one can isolate diverse VDI environment e.g., external vs internal VDI user VMs each get unique tag blocking all the communication.
- **User-Based Policies:** Through its integration with Active Directory (AD), the Service-defined Firewall enables user-specific security policies. User access to critical data center resources is governed by their AD group membership and access rights.
- **Distributed firewall enforcement:** The policy enforcement is done at the hypervisor in a distributed manner at the vnic level of virtual desktops, closer to the workload.

- **Object-Based Policy Model:** Security policies are based on a high-level object model, using attributes such as OS type, VM names, and Active Directory entries. This model eliminates dependencies on ephemeral IP addresses and low-level traffic attributes while enabling isolation of virtual desktops with just a few policies.

2.1.5.3 How to take a journey with NSX

Below are some of the ways an organization can leverage NSX service-defined firewall capabilities to micro-segment VDI environment to isolate desktops and block the lateral movement of threats:

- **Protecting VDI Infrastructure:** Leverage the distributed architecture of the Service-defined Firewall to protect the VDI infrastructure itself, including the Horizon management components.
- **Isolating desktop pools:** Isolate vulnerable user desktops from the rest of the data center infrastructure, via the network segmentation capabilities of the Service-defined Firewall.
- **User-based access control:** Define security policies based on users' identity and Active Directory group membership. Use the Service-defined Firewall to inspect and enforce user access control rights to designated applications and data center resources. For an example HR group can access HR-APP, Finance Group can access FIN-APP or restrict Employees vs contractor to certain resources etc.

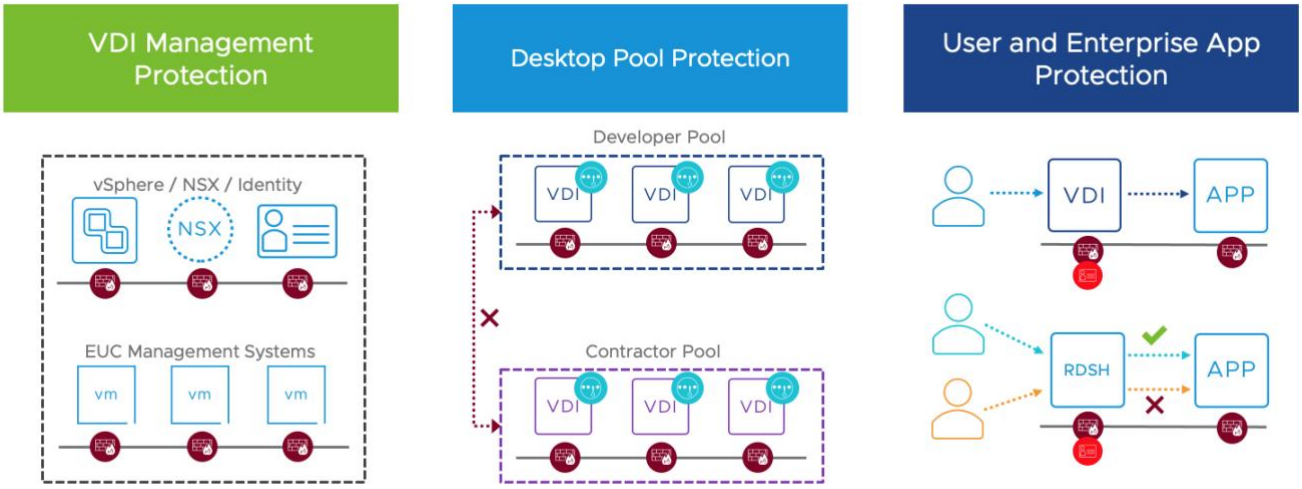

Figure 2-6: NSX Secure VDI

2.1.6 Consistent Security – Global/Region/Multi-site/DR

2.1.6.1 What is the requirement?

The multiple datacenters are very common for any organization to provide application availability, business continuity, scale, compliance requirements, or the organization's global-geo nature. The multi-datacenter/sites introduce few challenges to security operations with policy management across multiple data centers/sites or multiple deployments of the same solution. The organizations look for the following options:

1. Have a single pane of glass to manage policy across all the locations/deployments.
2. The ability to have cross-location dynamic policies.
3. The policy moves and updates as application workloads move from one data center to another for DR, migration or resource balancing, or site maintenance.

2.1.6.2 Why NSX?

NSX Service-defined Firewall inherently provides the single pane of glass through NSX manager to manage consistent policy across thousands of workloads across heterogeneous environments for a given deployment within a data center. Organizations can have multiple NSX deployments within the same data center to accommodate scale or isolate the environment as per organizations policy/compliance (PCI/Non-PCI, prod/non-prod etc). An organization can also have NSX deployment at a geographically dispersed datacenter location for Disaster Recovery or to accommodate the global nature of the business.

NSX federation solution with its NSX Global manager helps to provide a single pane of management across these separate NSX local deployments within the same data center or across different geographical sites. Here are the key capabilities of the solution:

- **Consistent Policy Configuration and Enforcement:** NSX global manager is a single management pane to define consistent security policies.
- **Dynamic Object-Based Policy across sites:** NSX federation helps to define dynamic policy between the location using Tag or other objects. This helps in many ways a) organizations can have zone or application workloads distributed across different locations, and dynamic policy can be applied between them. NSX federation control plane helps realize the policy correctly by syncing relevant group members between the sites based on the configuration.
- **Policy moves with workload between the site:** In case of disaster recovery, site maintenance or resource balancing policy moves and gets updated as workload moves between the site.
- **Customizable Policy:** NSX global manager allows the policy configuration based on deployment needs: Global policies – pushed to all locations specific NSX manager, Regional policies – pushed to only region-specific locations, DR-Site-pair policies – pushed to only disaster recovery protected and recovery site or Location specific – pushed only to a specific location.
- **Operational Simplicity:** The single management console brings simplicity to the overall security operation.

2.1.6.3 How to take a journey with NSX

NSX Federation enables an organization to manage consistent policy for their multi-site/DR use case and provides a single pane for managing global security policy across different NSX deployments, locations, and regions. NSX federation consists of Active/standby NSX global managers and multiple NSX local managers managing the workload policy locally at the location.

The administrator configures NSX global manager with relevant global, regional, or location-specific policies, then NSX global manager pushes that configuration to NSX local manager internally based on the span of the policy – global, regional or location-specific. NSX local manager takes the pushed configuration from GM to workloads to provide the intended security posture.

In addition to the management plane, the NSX federation also initiates a full mesh control plane between all the local managers. This control plane helps sync dynamic group members between the location based on the group/policy span. This capability of the NSX federation allows deploying the application in a distributed manner across the site.

The NSX federation is the ideal solution for disaster recovery as DR sites could be in either Active/Standby or Active/Active mode, allowing for better application availability and better data

center resource usage. With Active/Active DR, zone/application workloads can be distributed across sites. The policy dynamically gets updated in the normal state based on the workload location, and the policy moves with the workload in case of vMotion/site-recovery. An update is sent over the control plane for updating policy across federation local manager, based on the intended config.
In short, the NSX federation makes security operations simple by providing the same security posture irrespective of the physical location of the workload, and policy moves with applications as it moves between sites for DR migration, resource balancing, or site maintenance purposes.

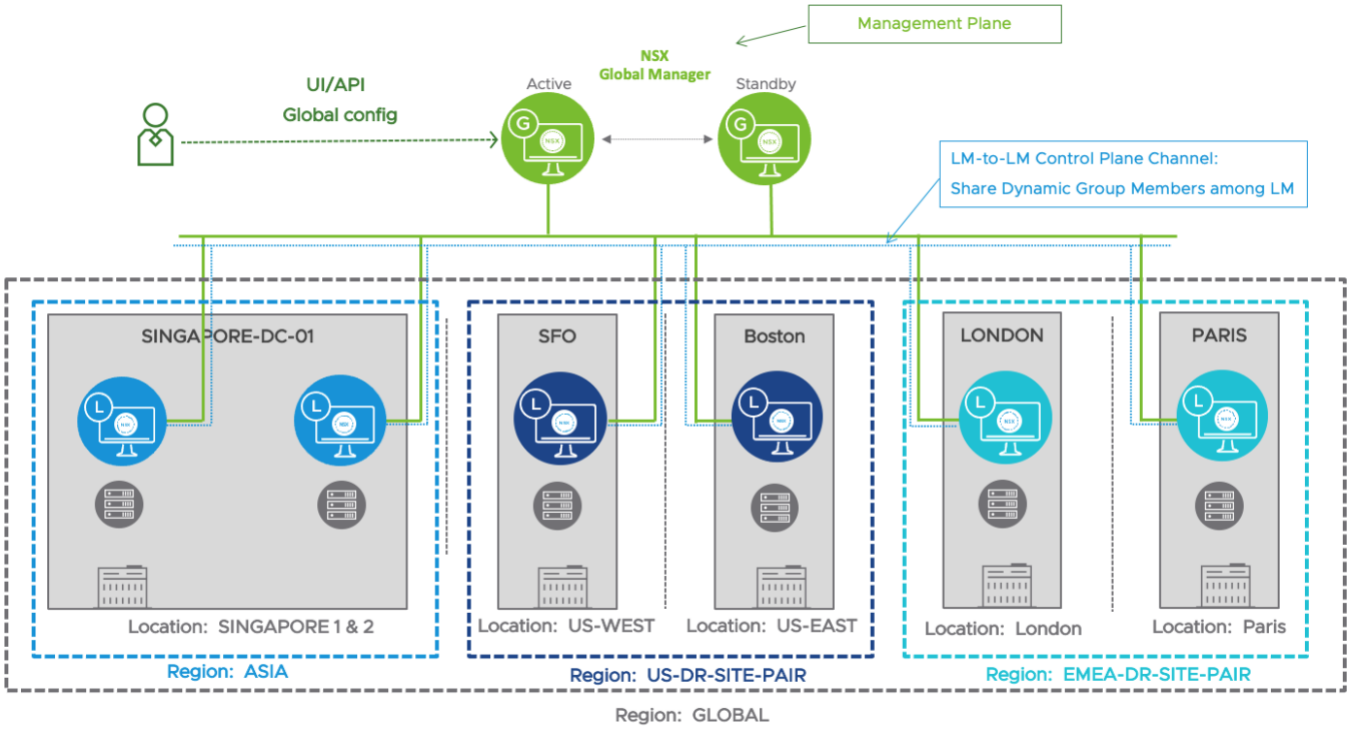

Figure 2-7: NSX Federation Management and Control Planes

2.1.7 Consistent Security – VM/Container/Physical Server/Cloud

2.1.7.1 What is the requirement?

Securing workloads across an entire environment is the fundamental goal of the security team. However, workloads come in various form factors: virtual machines, containers, and physical servers. In addition, workloads are hosted in different environments: on-prem, native cloud, or managed-cloud. The heterogeneity of the workload form factor and deployment type further challenges the organizations regarding security coverage, policy consistency, number of platforms to be managed, and overall operational simplicity. The organization's requirement is to have an operationally simple platform that provides consistent policy across virtual machines, containers, physical servers, and native cloud workloads without compromising the application and data security.

2.1.7.2 Why NSX?

NSX Service-defined Firewall is a purpose-built internal firewall for an organization's application and data security to provide consistent policy across the heterogeneous workloads and deployment type. NSX manager is the single pane of glass to define dynamic policy between and across all kinds of workloads. NSX manager treats each of these form factors as an application workload that needs to be protected, and the user can define dynamic network agnostic policies for all workloads as if they are of

the same type. In addition, here are more specific benefits for each of the workload type with NSX firewall:

- **Physical Server firewall options:** In any organization, a small percentage of workloads are still physical servers that are not virtualized for different reasons: no means to virtualize (AIX/Solaris), policy restrictions or performance requirements, or device-specific systems in place. Many organizations consider some of the physical servers as "Crown Jewel" because of the nature of the application running on them. NSX provides two ways to protect these "crown jewels" based on the deployment need:
    - Distributed firewalling using NSX agents running on supported Windows and Linux operating systems.
    - Gateway firewalling when deploying NSX agents on to physical server is not an option.

- **Container Security:** With NSX Firewall, the container/micro-services are considered similar to virtual machines with respect to networking and security. In short, a uniform operational model for virtual machines & containers, which is not possible with other solutions. That essentially means organizations have complete visibility into the containers. Each container pod would have its own distributed firewall policy at the container interface, just like a virtual machine. NSX enables this using NSX container plugin (NCP) integration with different container orchestrator platforms like VMware Tanzu, Red Hat Openshift, and native Kubernetes (K8s).

  The NSX container firewall policy can be configured:
    - K8s Network Policy – This is a dev-ops model that allows the application owner to define firewall policy as deployed using the native K8s construct.
    - K8s Label based - Define dynamic policy based on K8s labels assigned to container pods (maps to NSX tags).

- **Native Cloud:** NSX manager supports Amazon AWS and Microsoft Azure to help multi-cloud strategy customers have. NSX cloud solution comes in two forms to provide flexibility to customer based on their organizational requirement:
    - NSX enforced Mode – This is an agent-based solution on cloud instances.
    - Cloud-Native Enforced Mode – This mode provides an option for customers who do not want to install agents on a cloud instance. This uses native security groups to enforce the policy.

  In both cases, policy configuration/automation is done through the NSX Manager user interface or API. NSX firewall can leverage existing AWS or Azure tags to define firewall policy.

2.1.7.3 How to take a journey with NSX

NSX helps in having consistent policy across the virtual machine, container, Physical server, and cloud instances. NSX allows an organization to have a zero-trust model even for distributed multi-tier applications across different environments and/or different form factor.

For example, a multi-tier application can have its front-end deployed on multiple clouds and/or on-prem for high availability and business continuity. The back-end services could be hosted on on-prem as a virtualized service, or containerized micro-service and back-end databases are hosted on a physical server. This type of multi-cloud, multi-form factor distributed application can be protected using NSX micro-segmentation policy to have the zero-trust model.

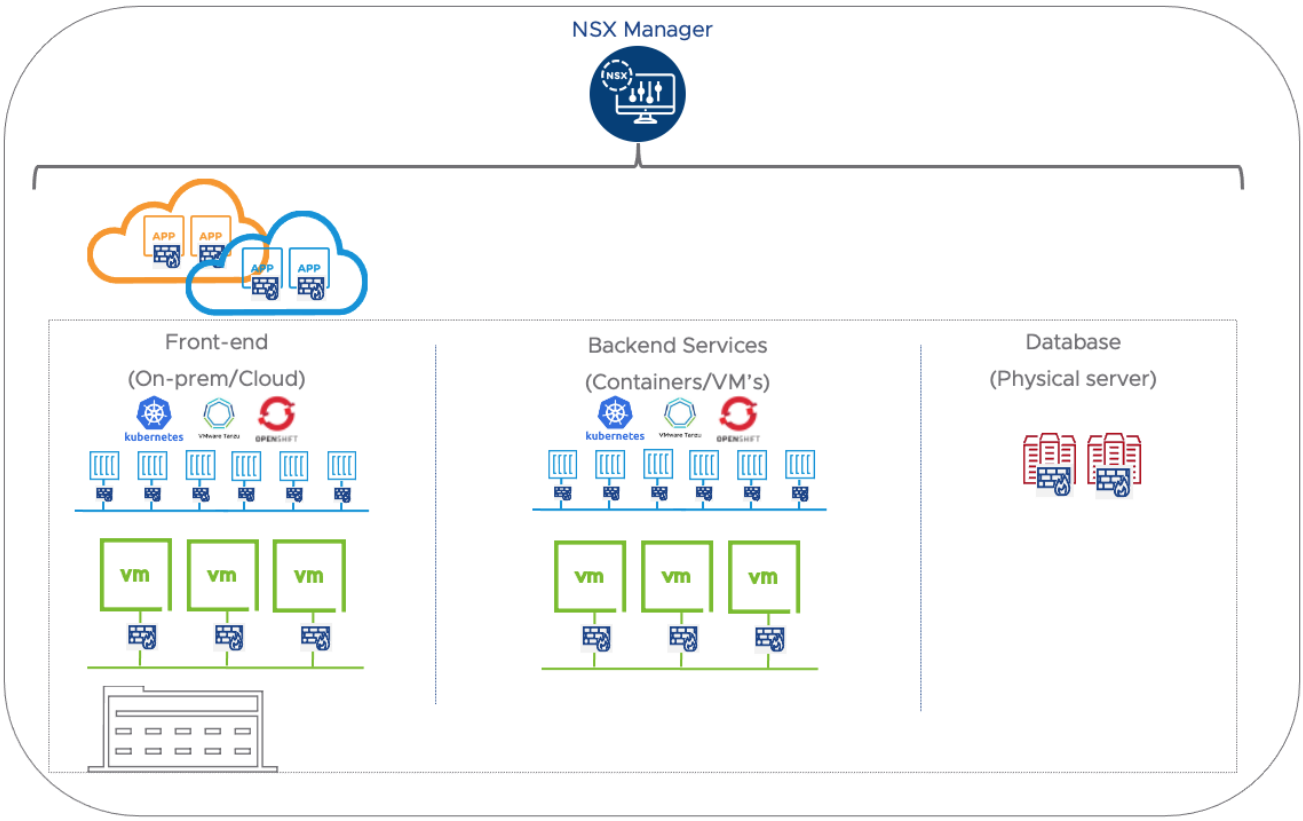

Figure 2-8: Consistent policy across diverse workloads

2.1.8 Simplified DMZ Security

2.1.8.1 What is the requirement?

DMZ designs have evolved over time to accommodate business requirements and how users or businesses access the datacenter application and internal resources. This introduced many sub DMZs, based on the data center's entry point: Internet/VPN/Branch/Business-to-Business. This required additional zoning to isolate and protect the data center's internal resources from each of these zones. This added complexity to the overall design and less optimal use of the overall compute and firewall resource.

The current DMZ design is based on legacy security measures and isolating DMZ resources. Not only does this design lack the security described in the segmentation section above, but it strands compute resources.
Furthermore, DMZs often use IPS functionality. In traditional architectures such as the one shown in the figure below, the IPS functionality lacks ubiquity and context for IPS. This provides a two-fold dilemma:

1. The lack of ubiquity allows one endpoint on a DMZ segment to be leveraged to attack another. Essentially it will enable lateral threat movement within DMZ.
2. The lack of context means that every packet must be inspected against the entirety of signatures for every endpoint.

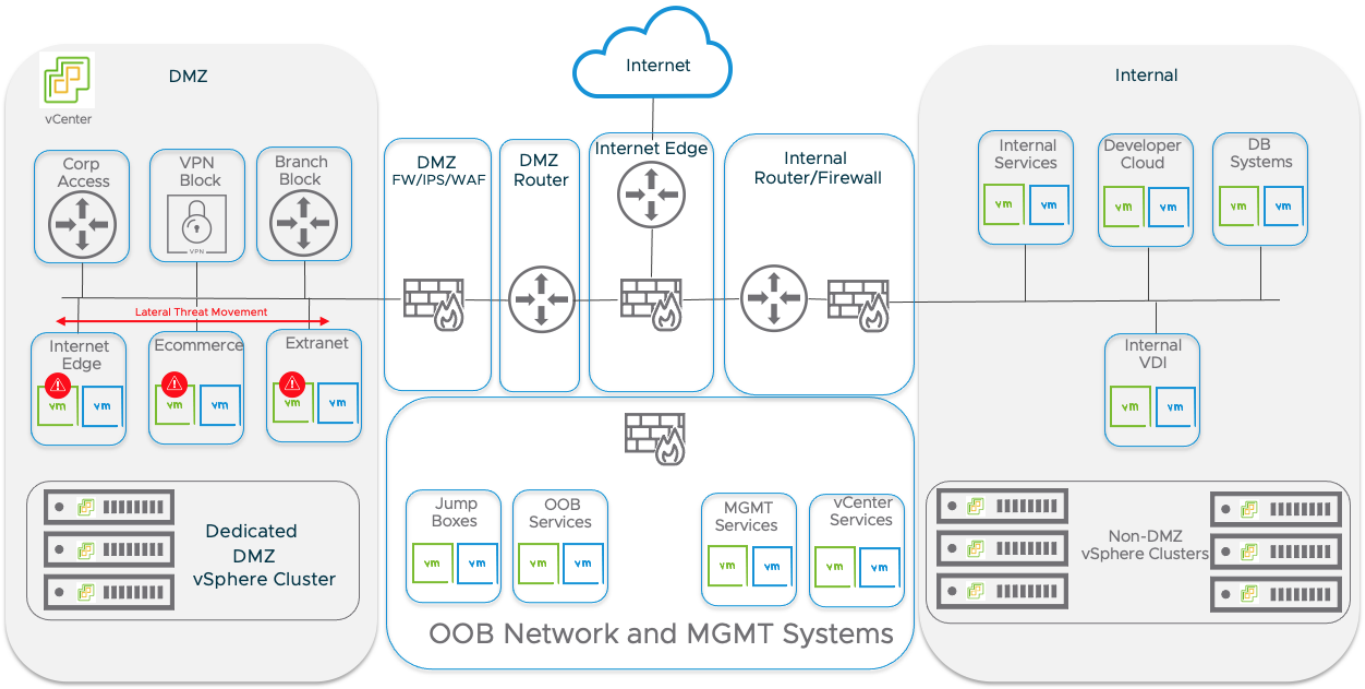

Figure 2-9: Legacy DMZ

2.1.8.2 Why NSX?

NSX has a software-defined architecture which means it is entirely independent of hardware. NSX provides scale, dynamic object definition, and a platform that provides expanded security architecture without the need to re IP or change security architecture. The scale provides a distributed, ubiquitous data plane that can secure all workloads and provide all the necessary security services. Here are few additional advantages of NSX for DMZ use case, some of which applies to other use cases as well:

- **Virtualize security zones** – Create and customize multiple virtual security zones for DMZ, internal teams and partners without requiring physical separation of the network.
- **Improve capacity utilization –** Reuse existing stranded compute capacity, eliminating the need for dedicated appliances.
- **Detect/Prevent lateral movement of threats** – Inspect East-West traffic at each workload using signature-based techniques, anomaly-based detection and protocol conformance checks.
- **Replace discrete appliances** – Leverage IDS/IPS capabilities native to NSX to replace traditional IDS/IPS appliances, reducing cost and complexity.
- **Custom IPS Profiles:** With the use of custom Profiles, NSX can streamline the signatures used to inspect traffic. This streamlining helps in greater efficiency and the reduction of false positives.
- **Dynamic Context-Based policy:** The dynamic object definition allows for a segmentation policy that keeps pace with dynamic infrastructure. If there is a web front end that auto-scales with load, the policy needs no modification as it can be defined based on a subset of the VM name, **dmz-web**, for example, or tag.

2.1.8.3 How to take a journey with NSX

A customer can take a similar approach for the DMZ use case with NSX as discussed above in the Segmentation, Compliance, and Advanced Threat Detection use case. NSX distributed firewalling and IPS capabilities help simplify DMZ security and more secure and agile than before. This helps to achieve compliance and protects the lateral movement of threats with DMZ and the data center.

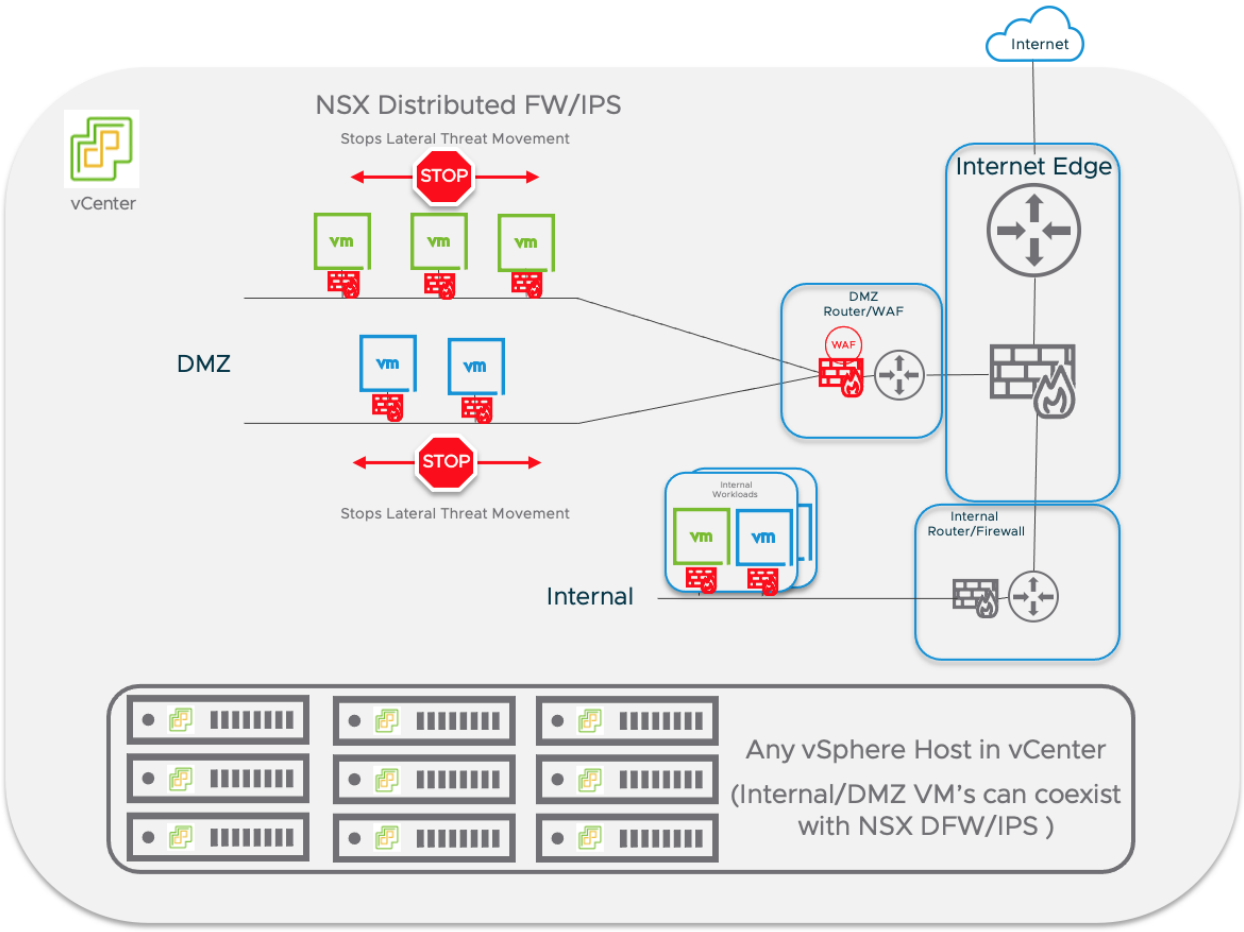

Figure 2-10 : Simplified DMZ Security with NSX

2.2 *Modern Security Journey*

One of the most common questions customers ask is: "How do I move to a modern security architecture from where I am now?" This is like asking what you should do to upgrade your house – the answer is: it depends on your lifestyle, where you want to grow, and where your house is lacking. This section will briefly provide an overview of a few customers who have undertaken the journey to a modern security infrastructure. As you will see, the important thing is that they have started, not how they started or what they did.

2.2.1 Segmentation

The first sample customer is a customer from a large company of over 50,000 employees with over 1800 hosts running 30,000 VMs. This is an old, solid, well-established company that has been in business for over 175 years. Their infrastructure includes pretty much every technology across the course of computer history from mainframes to modern containers and micro services. For this company, the first step in adopting a modern security strategy was as simple as separating prod from nonprod. This effort took them 18 months due to the complex nature of their environment. There were applications that were in use for decades and whose architecture and even ownership was poorly understood. Sorting through those details took a long time. But, at the end of the effort, every application was inventoried along with its use and ownership. One executive noted that the inventory

effort alone improved their security posture. The segmentation was a bonus. They are now poised to further segment their prod environment by business unit, continuing in an iterative manner.

2.2.2 Security Growing Up

The next company's journey is another large company with close to 16,500 employees. This company took the approach of starting at their branches and securing those first because the physical security at those branch locations was wildly varying. They started with their smaller branches and secured those, allowing them to get comfortable with the technology and its operational nuances. Should one of those branches go down due to operational unfamiliarity, the impact to the company as a whole was minimal. By the time they had secured all the smaller locations, there was a degree of comfort that gave them confidence to take on their medium branches and with that they grew confidence for their large branches and on to their corporate data center environment. Because of the pure software architecture of NSX, they were easily able to revise earlier implementations based on lessons from later stages as the project progressed.

2.2.3 Application Focused Security

The next example looks at a large hospital. This hospital has close to 1 million outpatient visits a year, over 500 beds and 6,000 employees. This hospital chose to secure their most precious asset first: their Electronic Health Records (EHR) application. This application is a multitiered, complex application which interfaced with every other application in their hospital: timeclock, billing, etc. Due to the complexity of the application, this customer chose to take this on as part of a 6-week professional services engagement with VMware. The environment was identified and tagged, with rules written, within 2 weeks. The rest of the engagement was about scheduling maintenance windows to enabling the deny rule at the end of each section in the policy, watching the logs and updating anything that may have been missed.

This engagement took place almost 4 years ago. Since then, the customer has maintained the policy and updated code. This is the value of NSX: when there is an effective tagging model that is chosen, the maintenance of the infrastructure is minimal and new features of the later releases were easily added in.

2.2.4 Security Through Migration

On occasion, a golden opportunity presents itself in which to adopt a new security model such as a new infrastructure migration. This last customer use case took advantage of a hardware refresh to build a new environment with security built-in. This customer is a SaaS software supplier which is subject to compliance. They are a $6B company with 5500 employees. Because a hardware refresh requires mapping out applications as they get migrated over, it does present an opportunity to build the new environment with the appropriate policies in place and settle the applications in to a new environment, with security built in from ground zero. In this instance, they used vRNI to map out their environment to size the new hardware environment. As part of that same vRNI assessment, they were able to map out their applications and their flows. With this policy suggestion exported from vRNI, they were able to preload the policy prior to migration. So, the applications migrated into a secure, modern infrastructure from the start.

As the above examples show, there are many ways to embark on the modern security journey. There really is no right or wrong way to start. The important thing is to start.

2.3 *NSX firewall – For All Deployment Options*

NSX firewall provides different security controls: Distribute Firewall, Distributed IDS, Gateway Firewall & Bridge Firewall, as an option to provide firewalling to different deployment scenarios.

A typical data center would have different workloads: VM's, Containers, Physical Server, and a mix of NSX managed and non-managed workloads. These workloads may also have a combination of a VLAN-based network or an NSX based overlay network.

The following Figure summarizes different datacenter deployment scenarios and associated NSX firewall security controls, which best fits the design. You can use same NSX manager as a single pane of glass to define Security policies to all of these different scenarios using different security controls.

1- NSX Managed Workloads with standard VLAN based networking: NSX distributed firewalling capability can be used to protect NSX managed VM's & Physical Server workloads.

2- Non-NSX Managed workloads on traditional VLAN based network: NSX gateway firewalling capability can provide the Inter VLAN routing and Firewalling. The Service Interface on NSX Tier-1 Gateway or External Interface on Tier-0 Gateway is used as a gateway & firewall for all non-NSX managed VLAN workloads.

3- NSX Managed Workloads with NSX Overlay for networking:

a) NSX Distributed Firewall can be used to protect NSX managed VM's, Containers (using NSX container plugin) & Physical Server workloads from East-West traffic perspective. This can be used for Zone-Segmentation, Application-segmentation & Micro-segmentation with both L3-L7 firewalling and IDS/IPS capabilities.

b) NSX Gateway firewall can be used as inter-tenant/zone firewall from north-south perspective, along with distributed firewall.

4- NSX managed Overlay workload bridged to Non-NSX managed VLAN: This is not a common scenario. This is an option if customer wants to have a NSX bridge to Overlay network is extended at layer-2 into a VLAN network using NSX Bridge. In this case, NSX managed Overlay workloads can use DFW/D-IPS, and bridge firewalling capability can secure traffic at the boundary between VLAN and overlay network.

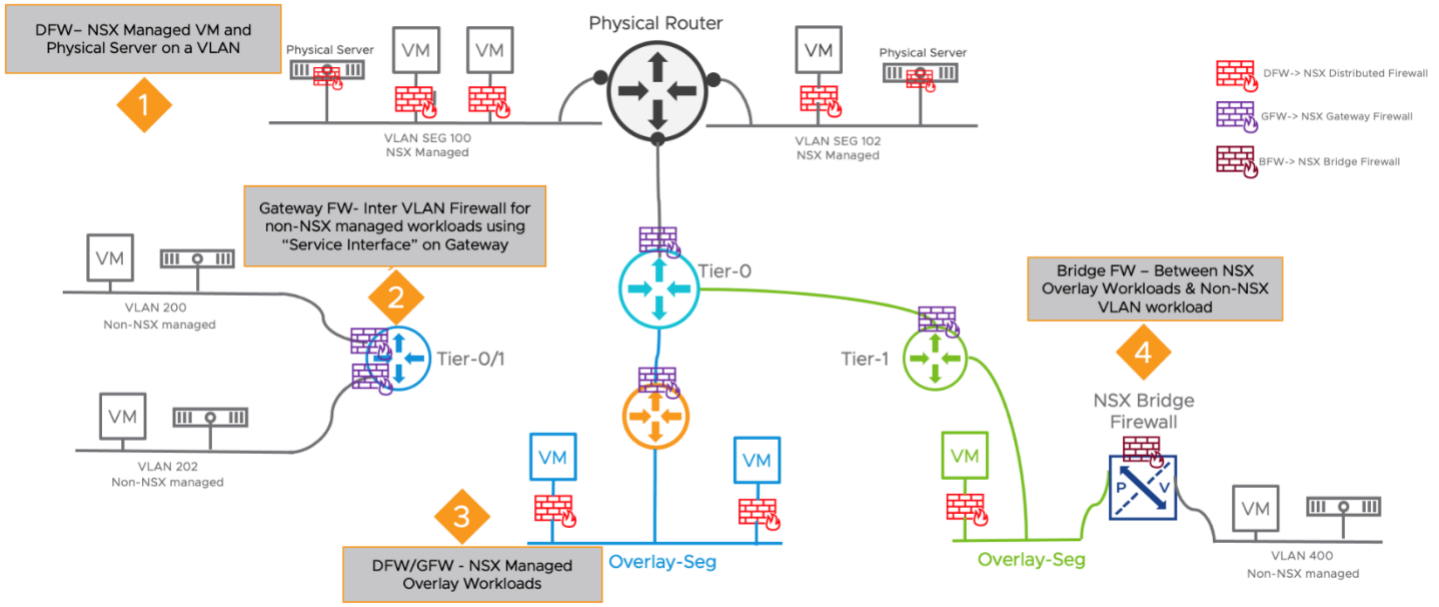

Figure 2-11: NSX Firewall For all Deployment Scenario

2.3.1 NSX distributed firewalling for VM & physical server

NSX distributed firewalling can be used for Network/Micro segmentation for NSX managed virtual or/and physical server. From implementation perspective NSX uses uses hypervisor kernel module on ESX for For vCenter virtualized workloads and NSX agents on supported Windows and Linux OS physical servers.

NSX firewall Deployment workflow:

- Deploy NSX Manager
- For VM security - Connect vCenter and Prepare vCenter ESX Cluster for NSX
- For Physical Server - Install NSX agent on Physical Server
- Define Network/Micro-segmentation policies.

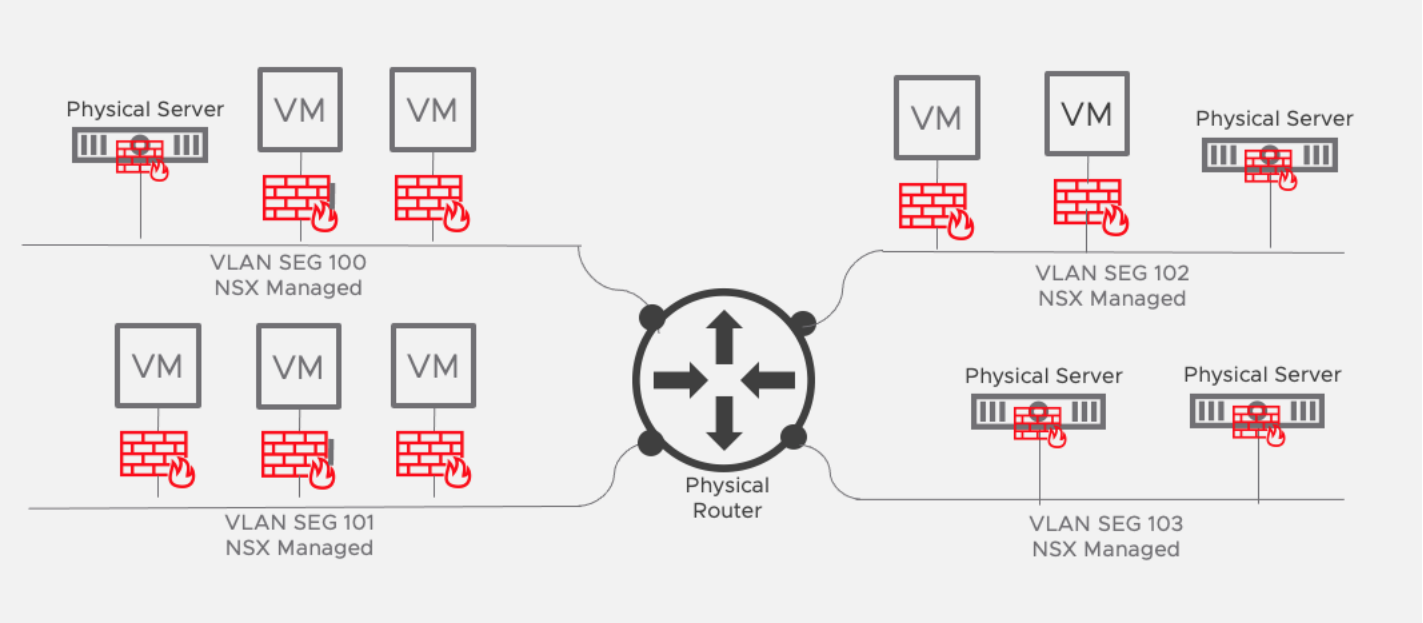

Figure 2-12: NSX Distributed Firewall for virtualized & physical server workloads

2.3.2 NSX gateway firewalling for VM & physical server

NSX gateway firewalling can be used for Network segmentation for non-NSX managed virtual or/and physical server.

From implementation perspective NSX uses Service Interface on Tier-1 gateway or External Interface on NSX Tier-0 Gateway as a L3 gateway/firewall interface for all VLAN workloads.

NSX firewall Deployment workflow:

- Deploy NSX Manager
- Provision Edge Cluster
- Create NSX Tier-0/1 Gateway
- Create Service Interface on Tier-1 or External Interface on Tier-0 with gateway IP per VLAN
- Define Zone/Inter-VLAN FW policies

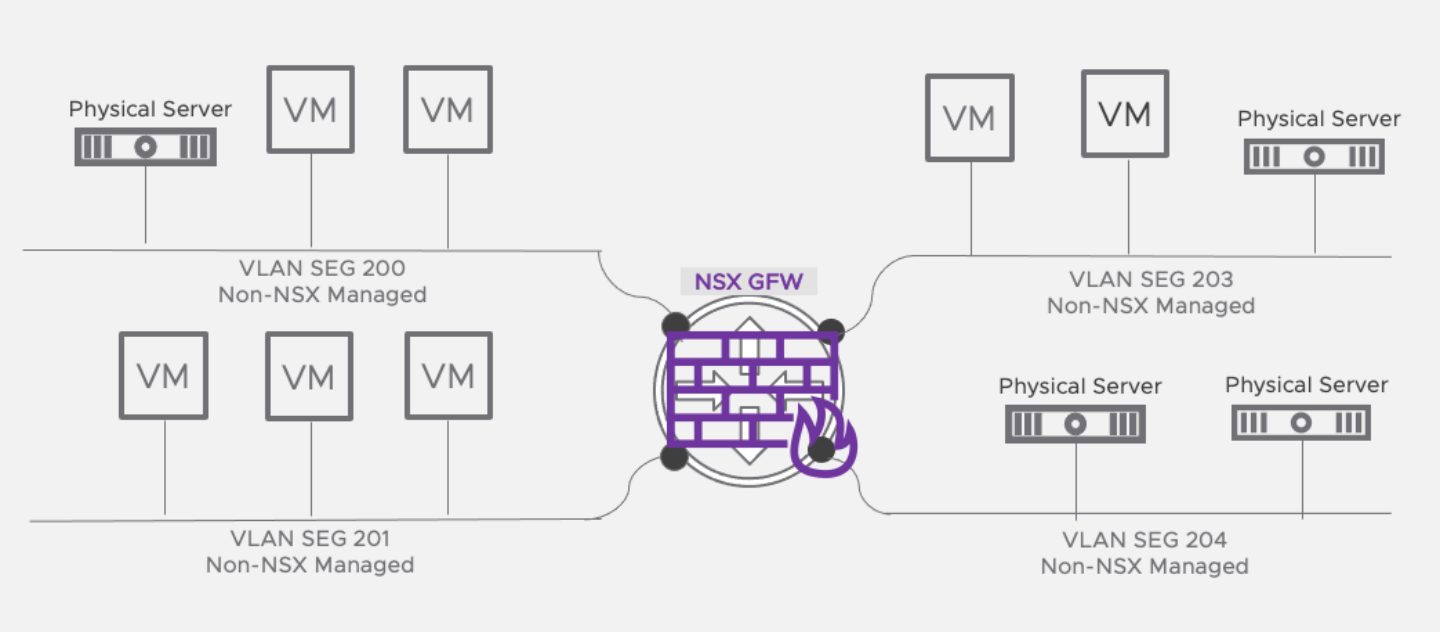

Figure 2-13: NSX Gateway Firewall for virtualized & physical server workloads

3 NSX-T Architecture Components

NSX-T reproduces the complete set of networking services (e.g., switching, routing, firewalling, load balancing, QoS) in software. These services can be programmatically assembled in arbitrary combinations to produce unique, isolated virtual networks with complete security in a matter of seconds. Although NSX does not require overlay networking, there is an added security assurance when overlay is used in that it is less likely that external networking mechanisms bypass NSX security controls.

NSX-T works by implementing three separate but integrated planes: management, control, and data. The three planes are implemented as sets of processes, modules, and agents residing on two types of nodes: manager appliance and transport.

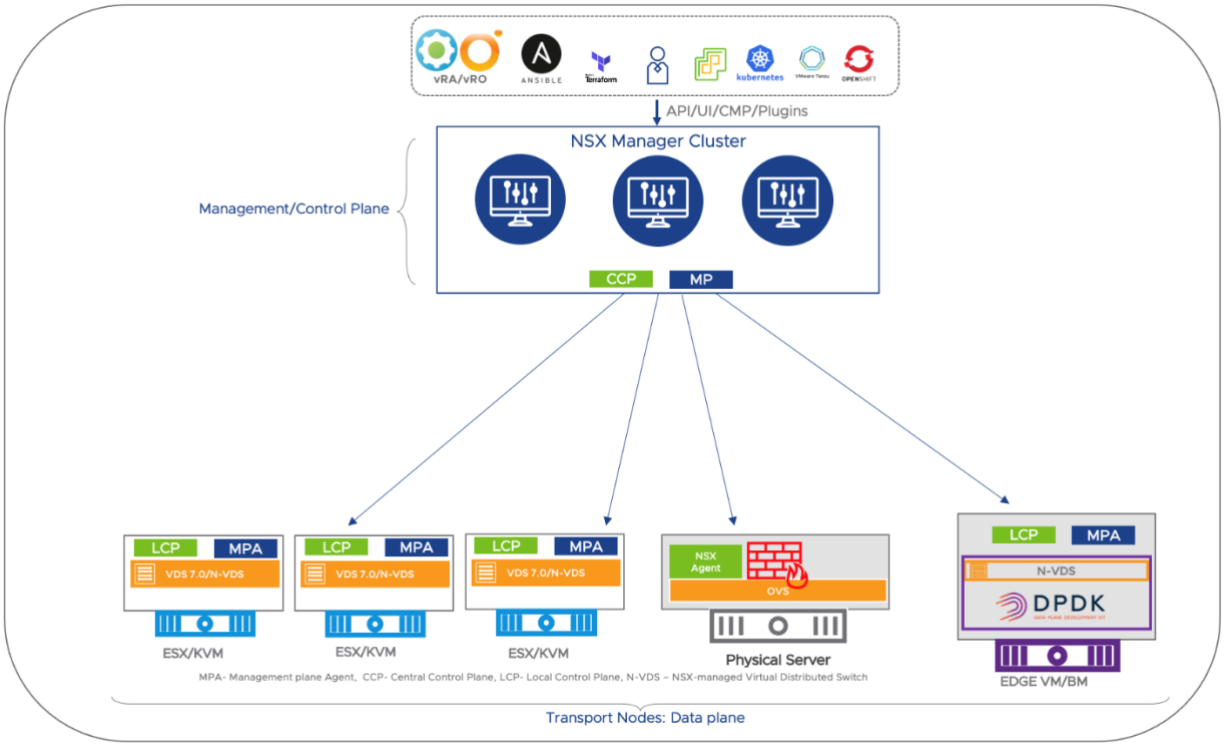

Figure 3-1 NSX-T Components

3.1 *Management Plane and Control Plane*

NSX architecture splits out the management and control plane functionality. The management plane provides the interface through which one interacts with the system whereas the control plane translates changes in system configuration and propagates dynamic system state.

3.1.1 Management Plane

The management plane provides an entry point to the system for API as well NSX-T graphical user interface. It is responsible for maintaining user configuration, handling user queries, and performing operational tasks on all management, control, and data plane nodes.

The NSX-T Manager implements the management plane for the NSX-T ecosystem. It provides an aggregated system view and is the centralized network management component of NSX-T. NSX-T Manager provides the following functionality:

- Serves as a unique entry point for user configuration via RESTful API (CMP, automation, including third party security managers) or NSX-T user interface.
- Responsible for storing desired configuration such as security policy in its database. The NSX-T Manager stores the final configuration request by the user for the system. This configuration will be pushed by the NSX-T Manager to the control plane to become a realized configuration (i.e., a configuration effective in the data plane).
- Expands rules and converts object to IP addresses and pushes rules to data plane
- Maintain object to IP database, updated via IP discovery mechanism
- Retrieves the desired configuration in addition to system information (e.g., statistics).
- Provides ubiquitous connectivity, consistent enforcement of security and operational visibility via object management and inventory collection and for multiple compute domains – up to 16 vCenters, container orchestrators (PKS, OpenShift & Tanzu) and clouds (AWS and Azure)

Data plane components or transport nodes run a management plane agent (MPA) that connects them to the NSX-T Manager.

3.1.2 Control Plane

The control plane computes the runtime state of the system based on configuration from the management plane. It is also responsible for disseminating topology information reported by the data plane elements and pushing stateless configuration to forwarding engines.

NSX-T splits the control plane into two parts:

- **Central Control Plane (CCP)** – The CCP is implemented as a cluster of virtual machines called CCP nodes. The cluster form factor provides both redundancy and scalability of resources. The CCP is logically separated from all data plane traffic, meaning any failure in the control plane does not affect existing data plane operations. User traffic does not pass through the CCP Cluster.
- **Local Control Plane (LCP)** – The LCP runs on transport nodes. It is adjacent to the data plane it controls and is connected to the CCP. The LCP is responsible for programing the forwarding entries and firewall rules of the data plane.

3.1.3 NSX Manager Appliance

Instances of the NSX Manager and NSX Controller are bundled in a virtual machine called the NSX Manager Appliance. Three unique NSX appliance VMs are required for cluster availability, for scaling out, and for redundancy. Because the NSX-T Manager is storing all its information in a database immediately synchronized across the cluster, configuration or read operations can be performed on any appliance.

Each NSX Manager appliance has a dedicated IP address and its manager process can be accessed directly or through a load balancer. Optionally, the three appliances can be configured to maintain a virtual IP address which will be serviced by one appliance selected among the three. The design consideration of NSX-T Manager appliance is further discussed in the NSX Design Document: https://nsx.techzone.vmware.com/resource/nsx-t-reference-design-guide-3-0.

3.2 *Data Plane*

The data plane performs forwarding or transformation of packets based on tables populated by the control plane. The data plane reports topology information to the control plane and maintains packet level statistics. The NSX-T data plane is the term which applies to all packet handling software which is part of the NSX-T scope. This data plane includes physical servers, hypervisors, NCPs, cloud enforcement mechanisms be they agents or gateways, and edge nodes which are handling traffic, be they bare metal or VM form factors.

Hosts running the local control plane daemons and forwarding engines implementing the NSX-T data plane are called transport nodes. Prior to NSX-T 3.0, transport nodes could only be run on an instance of the NSX-T virtual switch called the NSX Virtual Distributed Switch, or N-VDS. The N-VDS is so close to the ESXi Virtual Distributed Switch (VDS) that NSX-T 3.0 introduced the capability of installing NSX-T directly on the top of a VDS on ESXi transport hosts. For all other kinds of transport nodes and for all edge nodes, the N-VDS is required. The N-VDS is based on the platform independent Open vSwitch (OVS) and serves as the foundation for the implementation of NSX-T in other environments (e.g., cloud, containers, etc.). The NSX data plane supports both IPv4 and IPv6. In cases when only one protocol is used, the other one can be disabled to free up system resources.

As represented in Figure 2-1-, there are three types of transport nodes in NSX-T:

- **Hypervisor Transport Nodes:** Hypervisor transport nodes are hypervisors prepared and configured for NSX-T. NSX-T provides network services to the virtual machines running on those hypervisors. NSX-T currently supports VMware ESX and KVM hypervisors.
- **Workload Transport Nodes:** Workload transport nodes are nodes where the dataplane is instantiated not in a hypervisor, but in the workload itself. This would include physical servers and cloud workloads.
- **Edge Transport Nodes:** VMware NSX-T Edge™ nodes are service appliances dedicated to running centralized network services that cannot be distributed to the hypervisors (Gateway firewall, NAT, DHCP, VPN, and Load Balancing). They can be instantiated as a bare metal appliance or in virtual machine form factor. They are grouped in one or several clusters, representing a pool of capacity.

3.2.1 ESXi Data Plane

NSX-T provides network virtualization and security services in a heterogeneous hypervisor environment with ESXi and KVM hosts part of the same NSX-T cluster. The NSX-T DFW management and control plane components are identical in both ESXi and KVM hosts. Functionally, the NSX-T distributed firewall (DFW) is identical on both flavors of hypervisors. However, architecture and implementation have some differences between ESXi and KVM environment for DFW. The data plane implementation differs as they use a different type of Virtual Switch for packet handling. NSX-T uses the VDS7 or N-VDS or "NSX-T vSwitch" on ESXi hosts, along with VSIP kernel modules for firewalling. On KVM, on the other hand, NSX uses the "Open Virtual Switch" (OVS) and its utilities. The following section highlights the implementation details and differences between ESXi and KVM environment from data plane perspective.

NSX uses the N-VDS on NSX Edge Transport Nodes and older (6.5 and 6.7) ESXi host transport nodes. The N-VDS is a variant of vCenter VDS which was used prior to vSphere 7.0 where NSX-T manager fully manages NSX-T vSwitch. The NSX-T DFW kernel space implementation for ESXi is same as implementation for NSX for VSphere (NSX-v), it uses VSIP kernel module and kernel IO chains filters. NSX-T does not require vCenter to be present. For installations in vSphere 7.0 environments and going forward, NSX can use the VDS 7.0 for host transport nodes. With the VDS7, you can:

- Manage NSX transport nodes using a VDS switch
- Realize a segment created in NSX as an NSX Virtual Distributed Port Group in vCenter
- Migrate VMs between vSphere Distributed Virtual Port groups and NSX Distributed Virtual port groups.
- Send VMs traffic running on both types of port groups

Which VDS is running can have significant implications in vMotion events and other feature support. Of note, SR-IOV was not supported in the N-VDS, but is supported in the VDS 7.0. For full details of impacted features, see the NSX Documentation.

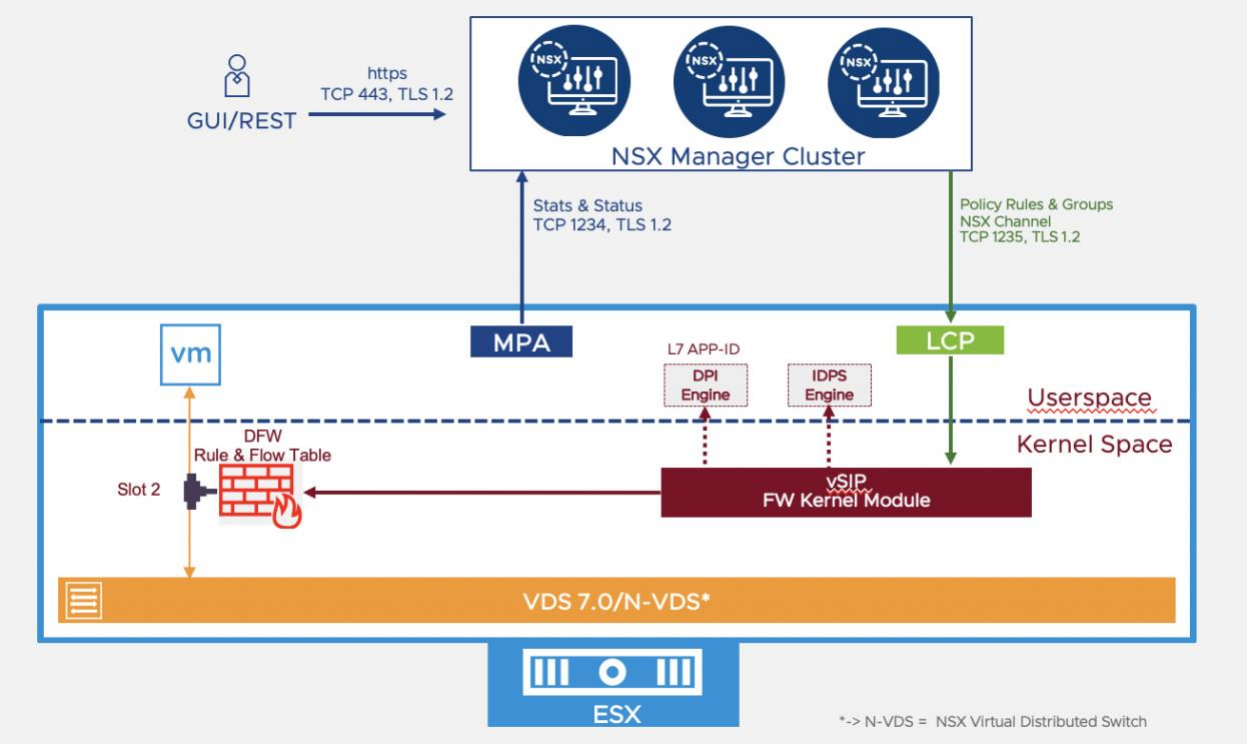

Figure 3-2 ESXi Data Plane

Regardless of which virtual switch is used in ESXi hosts, the DFW uses the VSIP kernel module and kernel IO chain filters. The LCP intelligently programs the FW rules table for every vNIC based on the "Applied To" field in the policy.

3.2.2 KVM Data Plane

As mentioned earlier, the NSX-T distributed firewall (DFW) functionally is identical on both hypervisors. This section will examine the details of the KVM data plane.

On KVM, the NSX Agent is the primary LCP component. It receives the DFW configuration from the central control plane. The NSX agent has a DFW wiring module as a component. It's used to generate Openflow flows based on the firewall rules that were pushed from the CCP. The actual DFW is implemented through the OVS.KO FastPath module. Stateless filtering is implemented through OVS-Daemon, which is part of openVswitch distributions. It implements the wiring implementation it received from LCP in the form of openflows. The Linux conntrack utility is used to keep track of 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. Statistics are exported through the Management Plane Agent directly to the NSX Manager. Figure 3-3 NSX-T KVM Data Plane details the KVM data plane.

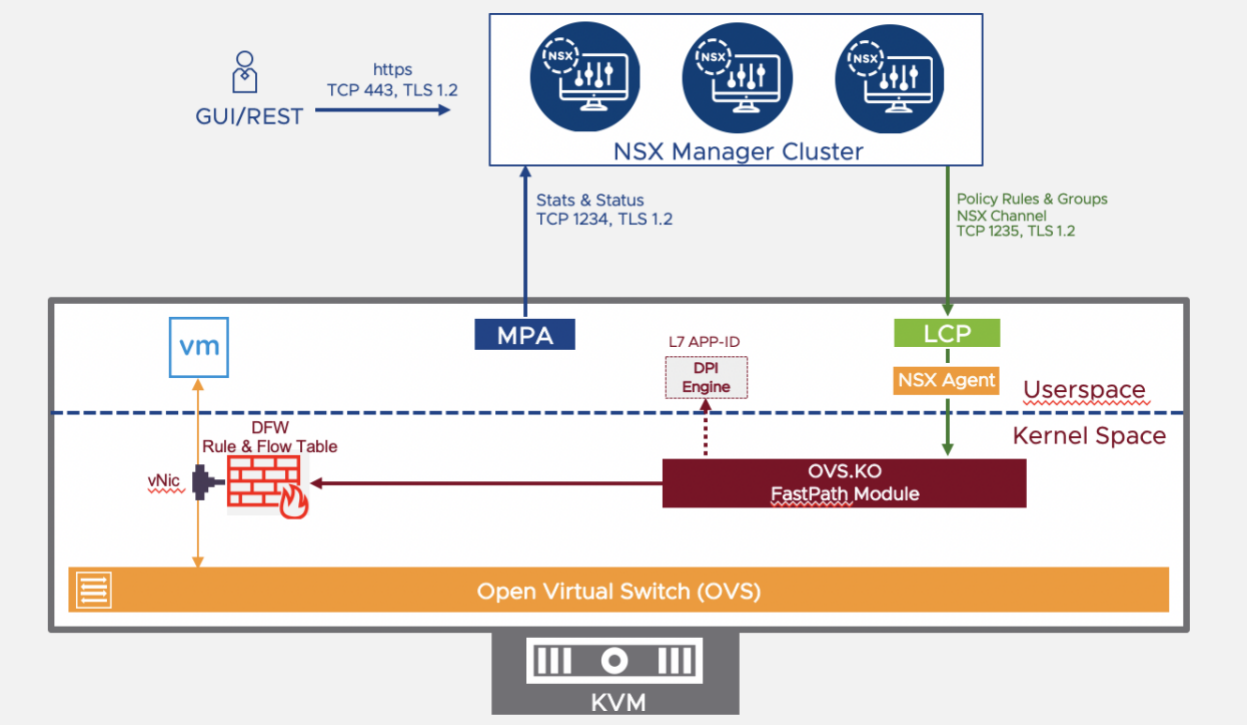

Figure 3-3 NSX-T KVM Data Plane

3.2.3 Physical Servers

NSX can provide security for physical servers as well as virtual servers by installing an NSX agent on the server. These servers can connect to the NSX environment either on overlay or VLAN backed networks. It is recommended that the servers be integrated into NSX-T using Ansible scripts. (After 3.1, this is no longer the recommended manner, but it is still supported.) To support NSX, the server must support third-party packages, and be running a supported OS per the Bare Metal Server System Requirements described here. The following terms are relevant in the physical server security:

Application: This represents the actual application running on the server (web server or data base server).
Application Interface: This represents the network interface card (NIC) which the application uses to send and receive traffic. One application interface per server is supported.
Management Interface: This represents the NIC which manages the server.
VIF: This is the peer of the application interface which is attached to the logical switch (This is similar to the VM vNIC).

To add physical servers to the NSX data plane, perform the following steps:

1. Install required third party packages on the Server. List of packages needed vary depending on the Operating System of the Application as listed here
2. Create an Application Interface for the workload.
3. Set up the Ansible and download and extract the integration from Github
4. Establish connectivity to the NSX Manager
5. Secure Workloads with DFW

Once configured, the physical servers will be with DFW rules which are pushed from the NSX Manager.

3.2.4 Distributed Firewall For Containers

The DFW is implemented in containers using the NSX Container Plug-In. The NCP is detailed in its own chapter. The discussion in this section will be limited the DFW implementation in container environments, as depicted in Figure 3-4 DFW for Containers.

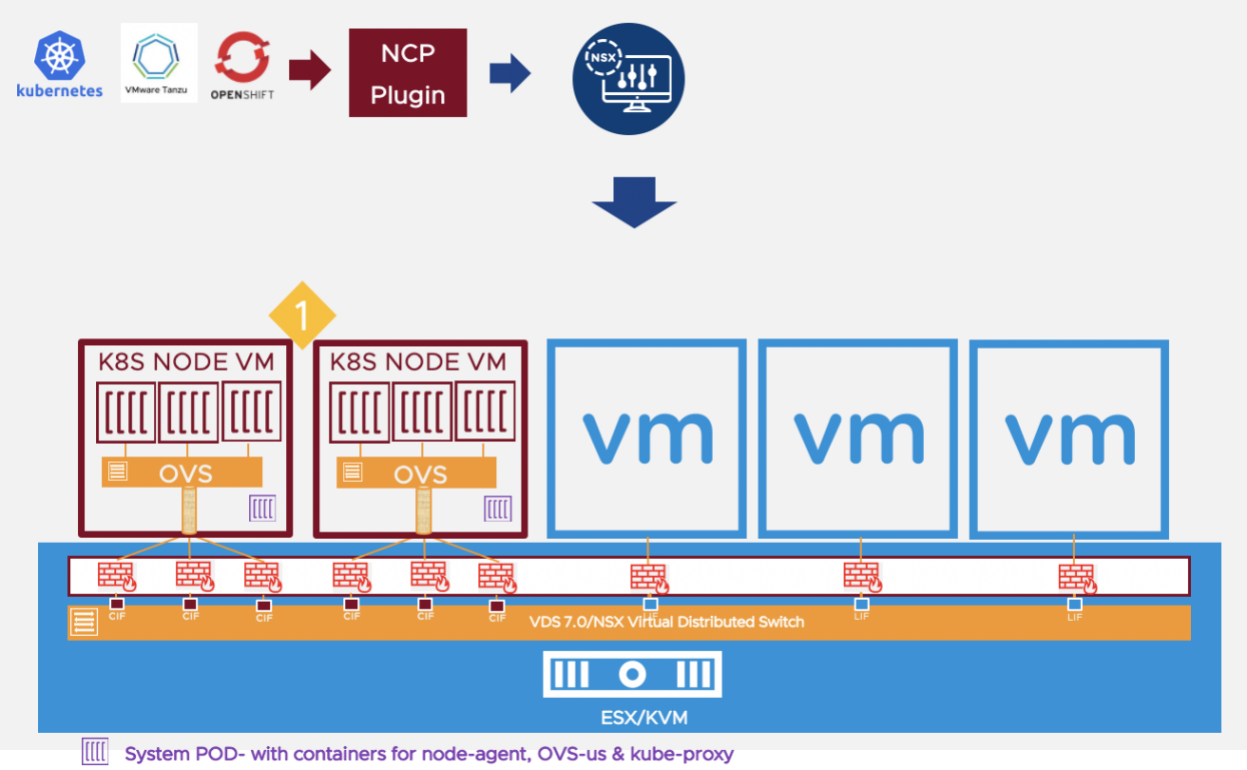

Figure 3-4 DFW for Containers

In containers, every Pod/Container has rules applied to its interface. This allows security policy to be implemented from container to container AND from container to/from physical or virtual servers. This allows for a uniform security policy application, regardless of the implementation details of the environment. For example, if there is a corporate policy that prohibits FTP and SSH to servers which source SQL, that policy can be implemented uniformly across physical servers, virtual servers and even any pods inside containers.

As is shown in Figure 3-4 DFW for Containers, containers are hosted on a Node VM which acts as Node VM for K8S/VMWare Tanzu. Each of the Container connects to the OVS in the Node VM. The OVS has VLAN trunk to N-VDS or VDS in the hypervisor on which it is hosted and connects each of the containers to virtual switch on a CIF (Container Interface). The OVS within the node does not switch traffic locally, but always sends it to the virtual switch in the hypervisor. Traffic between the CIF and OVS is carried over a locally significant unique VLAN tag per container. This allows each CIF to have a DFW to provide segmentation for each of the container pods.

3.2.5 NSX Firewalling for Public Clouds

NSX Cloud integrates NSX core components (the NSX Management cluster) with your public cloud to enable consistent network and security across your entire infrastructure. Currently, NSX Cloud supports only AWS and Azure. NSX cloud brings the agility needed for dev and test environments AND the structural integrity needed for production. When combined with VMware Secure State for auditing purposes, VMware security makes enterprise-ready.

In the public cloud implementation, NSX adds a few extra components:

- **Cloud Service Manager**: The CSM integrates the NSX Manager to provide cloud-specific information to the management plane. Think of it as the interpreter which is bilingual in both NSX and public cloud.
- **NSX Public Cloud Gateway**: The PCG provides connectivity to the NSX management and control planes, the NSX Edge gateway services, and for API-based communications with the public cloud entities. The PCG is not in the datapath.
- **NSX Agent**: This provides the NSX-managed datapath for workload VMs.

There are two modes for enforcing NSX policy in public clouds: *Cloud Enforce* mode which leverages native means such as AWS Security Groups and Azure Network Security Groups, and *NSX Enforce* mode which leverages NSX Tools for enforcement. Dynamic policy enforcement is based on Instance Attributes. This policy is fully configurable to each VPC with exclusion lists

3.2.5.1 Cloud Enforce Mode

In Cloud Enforce mode, NSX manages the security policy and provides security using Azure/AWS (network) security groups. In this implementation, there are no NSX tools inside the cloud instance, although the PCG is still required. Management is at the VNET/VPC level. This provides a common policy framework by translation NSX Policies to native cloud security policies. In this mode, default quarantine policies do not apply.

The installation steps for Cloud enforce mode are:

1. Install the Cloud Service Manger (CSM) on prem and register the CSM with NSX Manger & Cloud Provider Azure/AWS with right credentials
2. Install the NSX Public Cloud Gateway in your cloud Account
3. Push the micro segmentation security Policy to NSX Cloud Gateway, which in turn pushes policy to VPC/VNET

Unfortunately, NSX cannot overcome the limitations set by current cloud providers such as the number of security groups or the number of rules, nor the scope of NSG in Azure (regional) or SG in AWS (VPC).

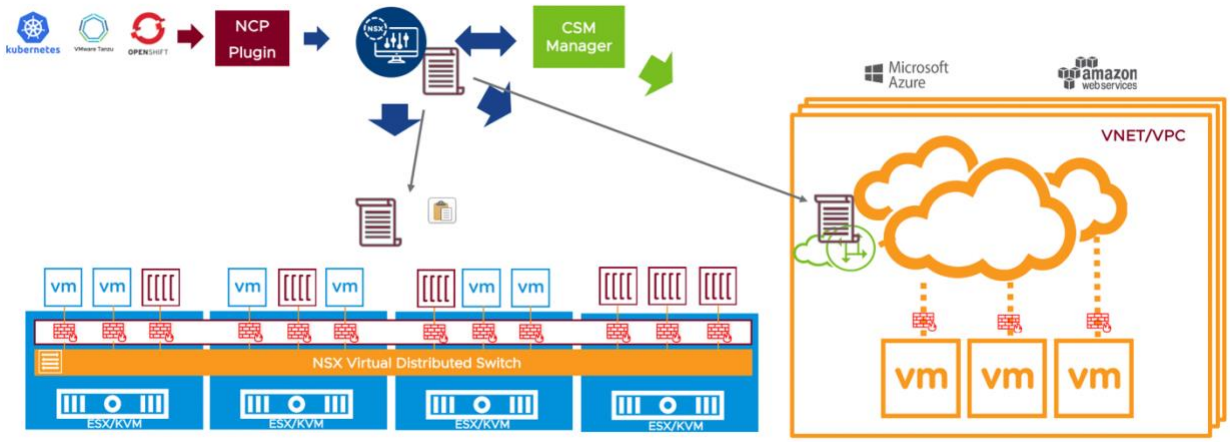

Figure 3-5 DFW on Public Clouds, Native Enforce Mode

3.2.5.2 NSX Enforce Mode

NSX Enforce Mode leverages NSX Tools inside the VMs to enforce a consistent security policy framework. NSX Enforce Mode allows for control at the individual VM level and a default quarantine. The list of currently supported operating systems for NSX Tools is listed in the NSX Documentation. The list of installation steps is:

1. Install the Cloud Service Manger (CSM) on prem & register with the NSX Manger & Cloud Provider Azure/AWS with right c redentials
2. Install the NSX Cloud Gateway in your cloud Account
3. Install NSX Tools on Cloud VM instances (Note: On Azure VNets, NSX Tools can automatically be installed if *Auto-Install NSX Tools* is enabled.)
4. Push the micro segmentation security Policy to NSX Cloud Gateway, which in turn pushes policy to NSX managed instances

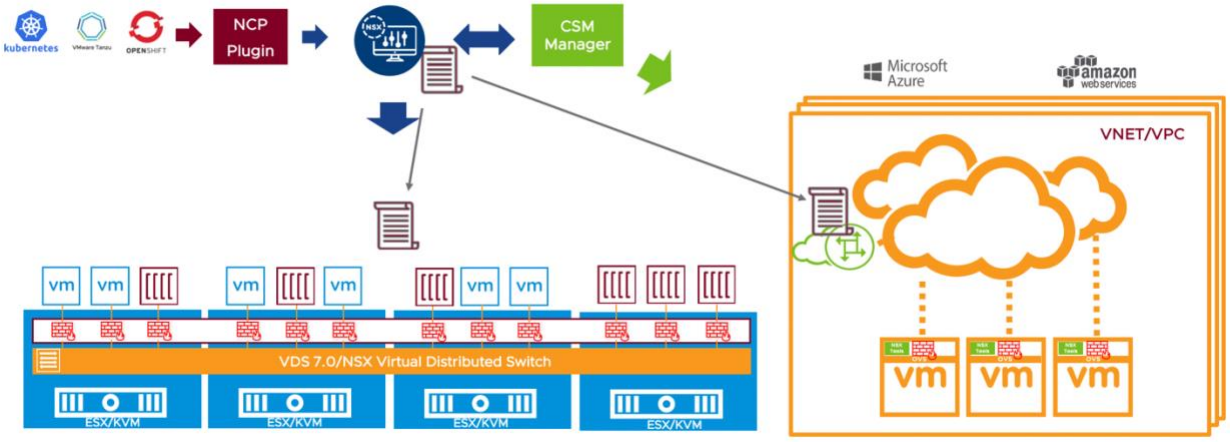

Figure 3-6 DFW on Public Clouds, NSX Enforce Mode

3.3 *NSX-T Consumption Model*

A user can interact with the NSX-T platform through the Graphical User Interface or the REST API framework.

3.3.1 NSX-T Role Based Access Control

NSX-T offers Role Based Access Control (RBAC). Roles can be assigned through integration with direct LDAP Identity Sources such as Microsoft Active Directory (AD) and OpenLDAP using LDAP, LDAPS, and StartTLS. Multiple domains or identity sources are supported to accommodate large enterprise configurations. Either users or groups can be assigned to roles. NSX provides four basic permissions: full access, execute, read, and none. Full access gives a user all the permissions. The execute permission includes the read permission. NSX comes with predefined RBAC role to meet different enterprise operations requirement, which are listed below. Starting 3.1.1 release, user can create custom RBAC rules to further customize the RBAC permissions on top of predefined roles. These RBAC roles can be assigned to remote AD/OpenLDAP/VIDM users/user groups and to local guest users.

RBAC Role	Permission
Enterprise Administrator	Super user; full access on all
Auditor	Read access on all
Network Admin	Full access on networking services, e.g. switching & routing
Network Operator	Read access on networking services, with the permission to run monitoring & trouble shooting tools
Security Admin	Full access on security Features
Security Operator	Read access on security services, with the permission to run monitoring & trouble shooting tools
Load Balancer Admin	Full access to Load Balancer configuration
Load Balancer Auditor	Read access to Load Balancing Configuration
NETX Partner Admin	Network Introspection workflow and policy.
GI Partner Admin	Guest Introspection workflow and policy.
VPN Admin	VPN workflow admin.

Figure 3-7: NSX-T Predefined RBAC Role

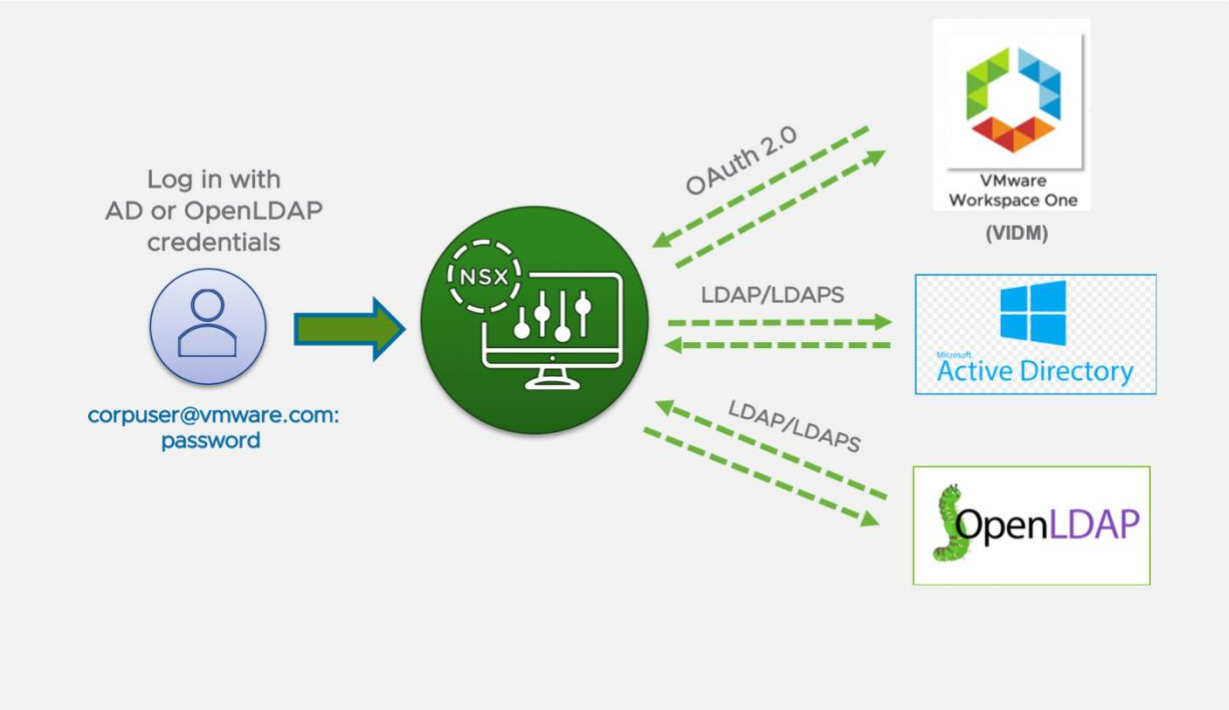

Figure 3-8 NSX-T RBAC with LDAP/VIDM Integration

Note that when integrated with Active Directory, if the username is changed on the AD server, the NSX role will need to be reassigned to the new username.

3.3.2 NSX-T Declarative API Framework

The NSX-T declarative API framework provides an outcome driven config option. This allows a single API call to configure multiple NSX networking & security objects for an application deployment. This is more applicable for customers using automation and for CMP plugins. Some of the main benefits of declarative API framework are:

- **Outcome driven**: Reduces the number of configuration steps by allowing a user to describe desired end-goal (the "what"), and letting the system figure out "how" to achieve it. This allows users to utilize user-specified names, not system generated IDs
- **Order Independent:** create/update/delete in any order and always arrive at the same consistent result
- **Prescriptive:** reduces potential for user error with built-in dependency checks
- **Policy Life Cycle Management:** Simpler with single API call. Toggle marked-to-delete flag in the JSON request body to manage life cycle of entire application topology.
- **Simplified and Performant Scripting:** Because here is no need to iterate through arrays, this simplifies the scripting and troubleshooting.

The NSX-T API documentation can be accessible directly from the NSX Manager UI, under Policy section within API documentation, or it can be accessed from the code.vmware.com link.

There are many scripts available on github that can be used to import policy configurations or even export information from the API. There is nothing available int eh UI that is not available via the API. For more details see the Getting Started Guide at https://communities.vmware.com/t5/VMware-NSX-Documents/NSX-Policy-API-Getting-Started-Guide-v1-0-pdf/ta-p/2775137

The following examples walks you through the declarative API examples for two of the customer scenarios:

3.3.2.1 API Usage Example 1- Templatize and deploy 3-Tier Application Topology

This example provides how the Declarative API helps user to create the reusable code template for deploying a 3-Tier APP shown in figure 2-3, which includes Networking, Security & Services needed for the application.

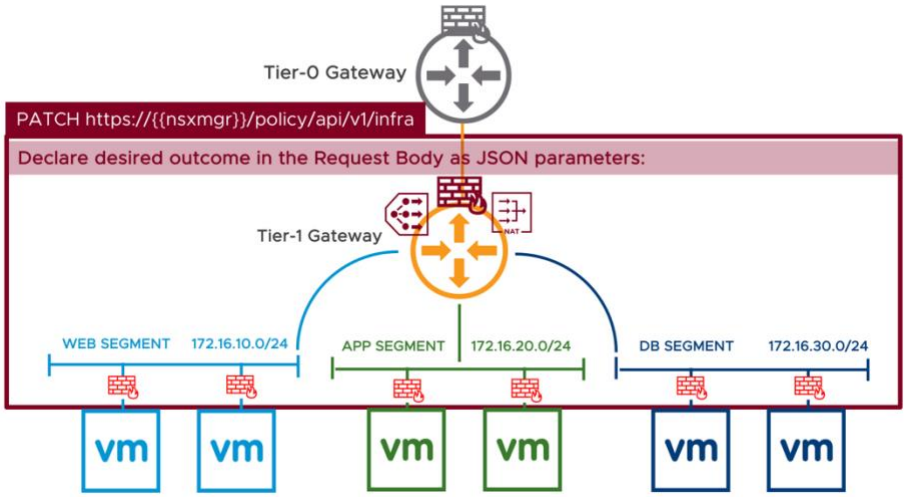

Figure 3- 9 Three Tier App

The desired outcome for deploying the application, as shown in the figure above, can be defined using JSON. Once JSON request body is defined to reflect the desired outcome, then API & JSON request body can be leveraged to automate following operational workflows:

- Deploy entire topology with single API and JSON request body.
- The same API/JSON can be further leveraged to templatize and reuse to deploy same application in different environment (PROD, TEST and DEV).
- Handle life cycle management of entire application topology by toggling the "marked_for_delete" flag in the JSON body to true or false.

3.3.2.2 API Usage Example 2- Application Security Policy Lifecycle Management

This example demonstrates how a security admin can leverage declarative API to manage the life cycle of security configuration, grouping, and micro-segmentation policy for a given 3-tier application. The following figure depicts the entire application topology and the desired outcome to provide zero-trust security model for an application.

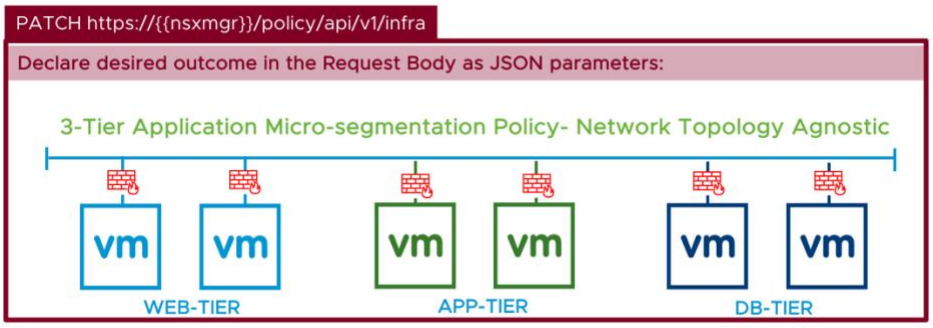

Figure 3-10 JSON Declarative statement

To define the desired outcome for defining grouping and micro-segmentation polices using JSON and use single API with JSON request body to automate following operational workflows:

- Deploy allow-list security policy with single API and JSON request body.
- The same API/JSON can further leveraged to templatize and reuse to secure same application in different environment (PROD, TEST and DEV).
- Handle life cycle management of entire application topology by toggling the "marked_for_delete" flag in the JSON body to true or false.

The details of both sample examples are fully described in the NSX-T Data Center Reference Design Guide.

4 Virtual Firewalling

The practice of firewalling goes back to the early days of the Internet when there was a leased line connecting the "Inside" to the "Outside". First, the router which provided that connection was configured with an access list or filter which define which traffic types were allowed in which direction. As time went on, there was a recognition that simple router access lists did not suffice to secure these connections because a greater level of intelligence was needed and firewalls were born. As corporate networks grew and changed, more firewalls were added at the access points: Remote Access entries, Partner connections, etc. As traffic needed firewall servicing, it would be directed to these central appliances. With the proliferation of these firewalls, the concept of a firewall manager was born. This manager would provide a central point for the administrator to configure firewalls. This configuration would later be pushed to the firewalls themselves. This architecture of a manager controlling a network of choke point appliances has remained unchanged for decades. This is the architecture of legacy firewalls today.

4.1 *NSX Firewalling: A New Approach*

NSX-T brings a new paradigm to the firewall strategy with the Distributed Firewall. As described in the previous chapter, NSX-T provides a central management and control plane for a distributed data plane. From a security perspective, this means centralized control and policy with ubiquitous distributed enforcement. Whereas legacy firewalls are discrete chokepoints which need to have traffic directed to them (and were thus easily bypassed), NSX-T Distributed Firewalls (DFWs) operate on every virtual NIC (VNIC) of every VM, seeing every single packet entering or exiting the VM without the need to reroute those packets, and without the need to change any IP addressing. When a vMotion takes place and a VM is moved from one host to another, the legacy firewalls which were designed for static infrastructure put a greater burden on the infrastructure to direct traffic to them. Because the DFW is tied to the vNIC of the VM, it is impossible to bypass the DFW as it moves with the VM as part of the VM in the vMotion event. This also means that only the firewall administrator can disable its functionality to allow traffic to bypass the DFW.

When comparing the Distributed Firewall to legacy firewall architecture, it is important to note certain limitations which were part of the legacy model. For example, legacy chokepoint firewalls cannot secure endpoints on the same VLAN, unless they are deployed in Layer 2 mode (in which case one instance is deployed per application.). Most importantly, legacy firewalls still bear the markings of their birth: they are built around IP address constructs. Although legacy firewalls have central managers, those central managers are merely aggregators. The policy definitions in those managers are still built around IP addresses or groups of IP addresses. NSX, on the other hand, is built around a software defined policy constrict. Although policy definition in NSX can be around IP addresses, it does not have to be around IP addresses. With NSX, the grouping criteria may be logic encompassing the OS, a substring of the vm name, and perhaps a tag or two. In the case of container environments, the labels used in containers can be leveraged for the grouping. The point is that a software born firewall architecture has a software defined means for identifying the groups.

In short, as opposed to hardware-based firewalls which are wrapped in a software wrapper to become a VM, NSX firewalling is software born and software architected. It is ubiquitous and pervasive in its data plane for enforcement, while being diverse and agile in its central management place. NSX firewalling is a firewall architecture that supports the diverse and expansive needs of modern infrastructure.

There are three types of firewalls in the NSX-T architecture: Gateway Firewalls and the Distributed Firewall is an element of firewalling attached to the data plane source/destination (be it a pod in a container, a VM on prem or in a public cloud, or a physical server. The third type of firewall is the Bridge Firewall. Gateway firewalls are designed to run in the periphery or boundaries; they are North-South Firewalls. Two examples of these peripheries or boundaries might be between the physical to virtual boundary or between tenants. The Distributed Firewall runs in every hypervisor, container, or physical server and is an East-West Firewall. A key characteristic of the DFW is that it is network agnostic and pervasive. The Bridge Firewall is used only in NSX bridges (which are used to adjoin two L2 domains at layer 2 – without routing- such as a VLAN and a geneve overlay segment). The Bridge Firewall is a layer 2 firewall and beyond the scope of this document. For more details on the Bridge Firewall, see the NSX documentation.

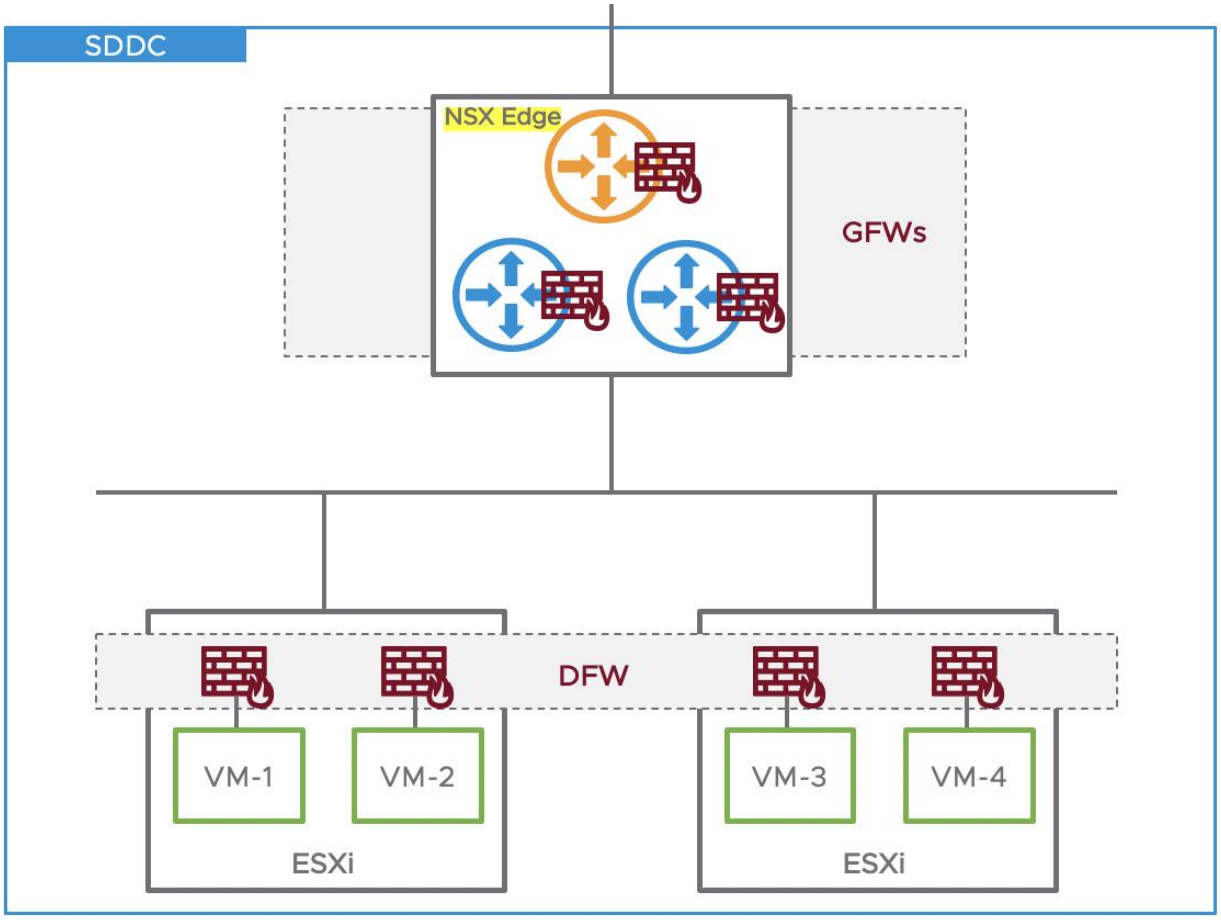

Figure 4 - 1 NSX-T Firewalls

Figure 3-1 shows the Gateway firewalls running on NSX Edge nodes. NSX Edge nodes are virtual appliances or physical servers managed by NSX. They provide networking and security services for north-south traffic, interfacing with top of rack switches. One NSX Edge node can contain multiple Gateway firewalls. These Gateway firewalls have their own firewall or security rule table; they operate individually while being centrally managed by NSX. Gateway Firewalls make it easy to create security zones according to network boundary and manage firewall rules per zone or organization. Also, in figure 3-1, the NSX Distributed firewall is shown. The DFW is built into the hypervisor as if each VM has its own firewall. Distributed Firewall is managed as one universal firewall. It is agnostic to network topology and enables micro segmentation without requiring network segmentation. Combined together, NSX Gateway Firewall and Distributed Firewall can secure north-south traffic and East-West traffic of Data Center.

4.2 *Gateway Firewall*

As mentioned above, the Gateway Firewall provides firewalling services at boundaries or perimeters. The Gateway Firewall is supported on both Tier 0 and Tier 1 routers (for more information about Tier 0 and Tier 1 routers, see the NSX Design Document). Note that although the Gateway Firewall is instantiated in the same software as the Tier 0 and Tier 1 routers, its functionality IS NOT equivalent to an access list in traditional routers. Even if routing is performed elsewhere (ie, disabled on the T1 or T0), the Gateway Firewall will still function. The Gateway firewall provides firewalling services and services that cannot be distributed such as NAT, DHCP, VPN, and Load Balancing, and as such need the Services Router component of the router. This means that the Gateway Firewall is implemented in the NSX Edge Transport Nodes, which are dedicated DPDK appliances. Further, the Gateway Firewall provides functionality such as Service Insertion which will be described in Chapter 7.

4.2.1 Zone Firewalling with the Gateway Firewall

As the Gateway Firewall is designed to work at boundaries, it is ideal for designing zones.

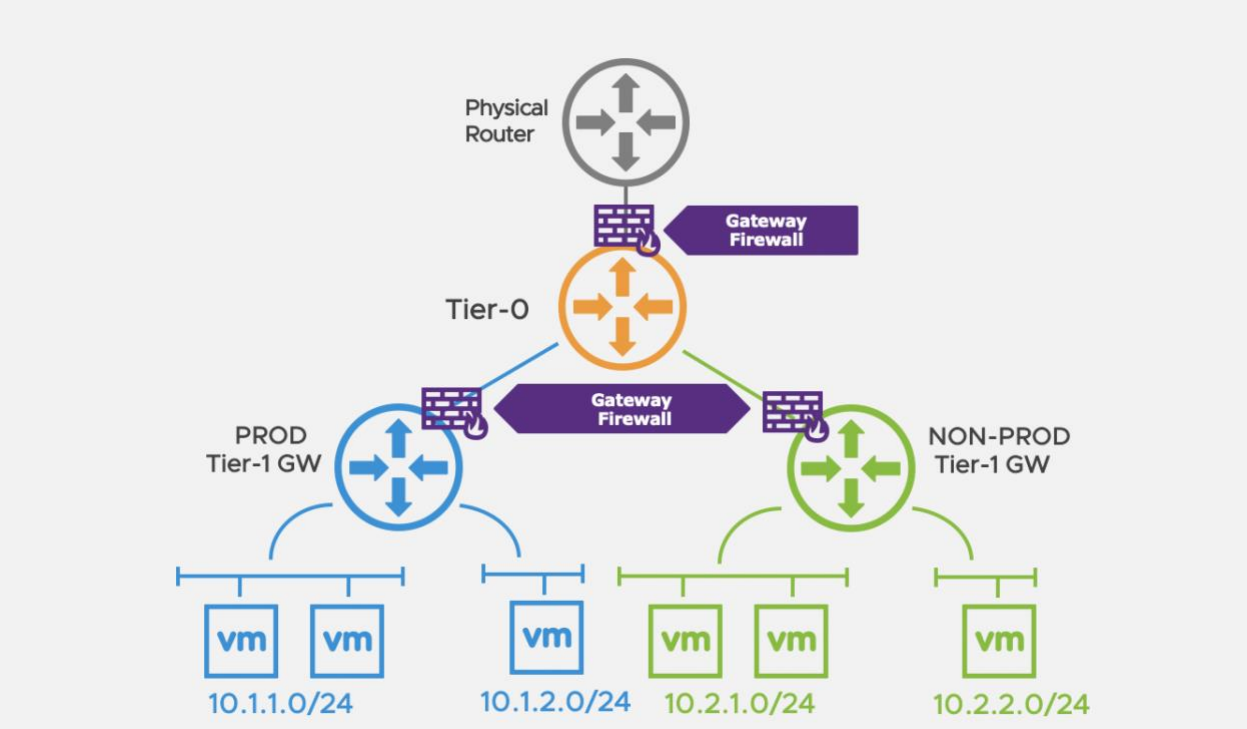

Figure 4 - 2 NSX-T Gateway Firewalls implementing Zones

As figure 3-3 shows, the Gateway Firewall is applied to both the Gateway Uplinks and Services Interfaces. In this figure, the Prod and Non-Prod zones can have policy defined for each zone independently. Although this figure does not depict it, the two zones could even have overlapping or duplicate address space, with NAT at the T0, or each T1. In either case, security policy is implemented at the gateway level for all traffic entering or exiting the respective zones. The T0 gateway is where policy securing the NSX environment is applied. This policy is applied on the northbound interface. As is shown in this scenario, the Tier 0 gateway is also an inter-tenant connector. For firewalling between tenants, the policy is applied on the northbound interface of the T1s. At this level, for example, one may define the policy that the Prod can talk to the Non-Prod, but not vice versa. The T1 gateway firewalls are ideal for implementing zone or tenant specific policy. The

T1 would be the ideal place to define which services are available within that zone – say web services can go to the 10.1.1.0/24 segment only. This hierarchical definition of policy provides a means to minimize policy clutter.

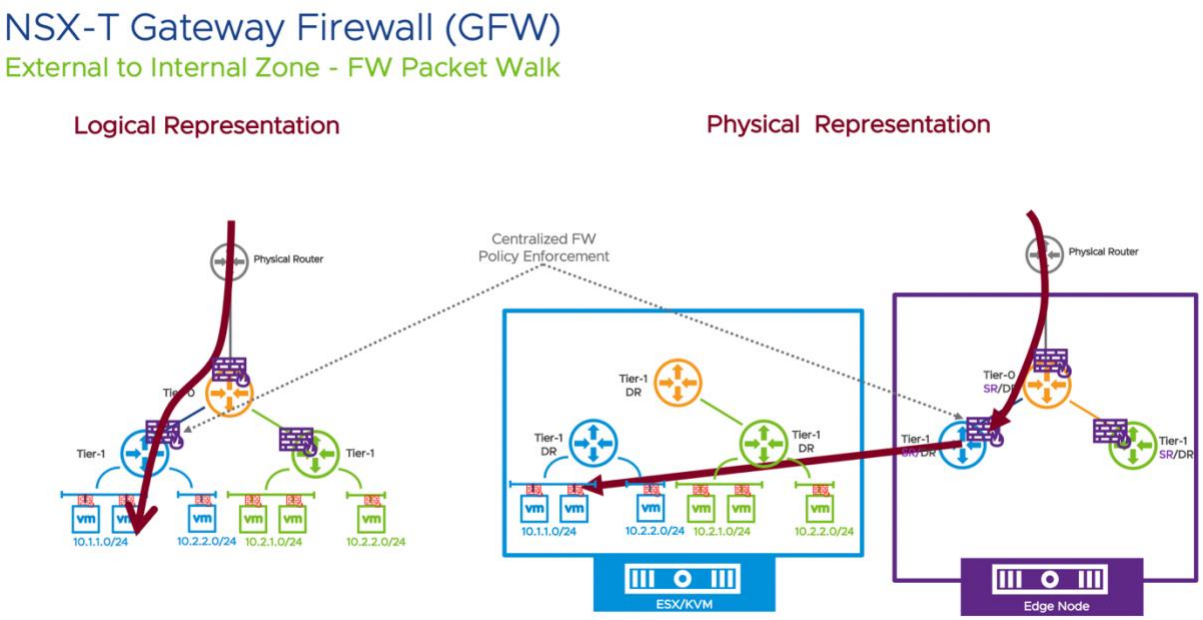

Figure 4 - 3 NSX-T Gateway Firewalls, External to Internal Traffic

Figure 3.4 depicts the path of a packet though the above-described zone configuration. In this scenario, a packet originates in the outside and is destined to the right VM on the 10.1.1.0/24 segment. The left half of Figure 3-4 shows the logical representation of this flow. The right half shows the physical representation of the flow. In the physical manifestation of this environment, the Edge Node hosts the Gateway Firewall itself (as indicated above). Given that the Edge Node is also where routing connecting the virtual world to the outside world would happen, this places the security at the outermost boundary.

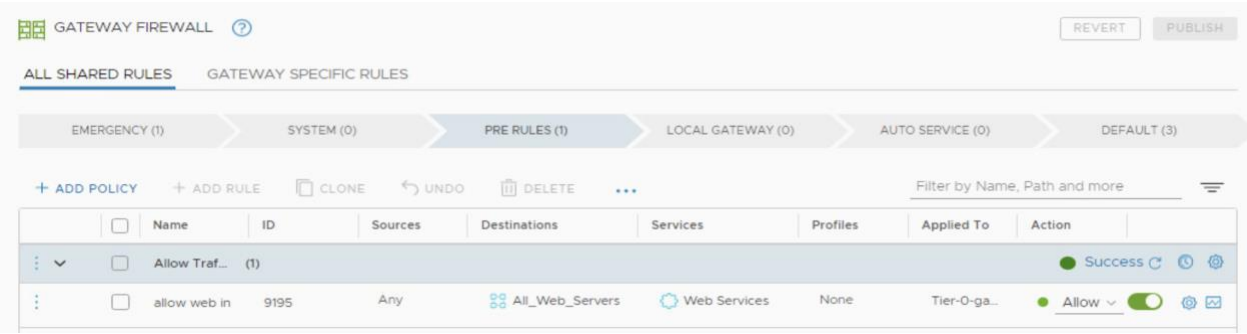

Figure 4 - 4 T0 Gateway Firewall Rule

In this case (as is seen in figure 3.5), there is a gateway policy on the T0 Gateway firewall that allows all http traffic to any VM with *web* in the name. (Note that this is where following a naming convention pays off!) This will allow the packet through the perimeter. Now, just because the traffic is allowed through the gateway, that does not mean it is allowed into the zone.

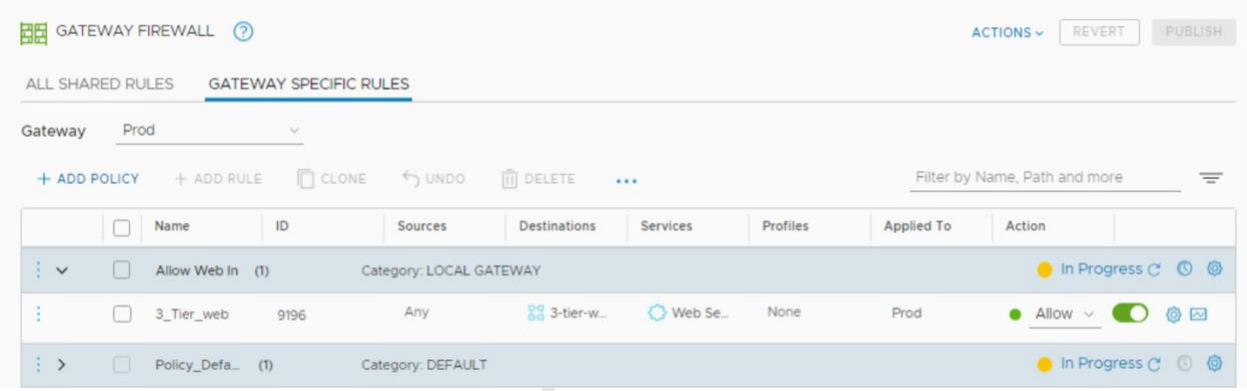

Figure 4 - 5 T1 Gateway Firewall Rule

In this case (as is seen in Figure 3.6), there is a policy on the T1 Gateway firewall that allows all http to that application VM. This is a layered gateway firewall security approach. The T0 gateway firewall has a general policy - what is allowed in, which tenant can talk to which other tenant - and the T1 gateway firewall has a more specific policy, regarding its own tenancy. This distributed, hierarchical model allows for optimal efficiency where the T0 Gateway firewall is not cluttered with details about each of the zone specifics.

In the physical representation, both the T0 and the T1 firewalls are on the Edge transport Node. Thus, the packet does not leave the Edge host until it has passed through the T1 Gateway Firewall. At this point, the packet is sent to the host with the destination VM, encapsulated in any overlay headers that may be required. (The network details of this are included in the NSX Design document.). Upon arriving at the destination host, the packet will then be examined by the Distributed Firewall for that VM, as described in the following section.

Next, for consideration is inter-tenant traffic, or traffic between tenants.

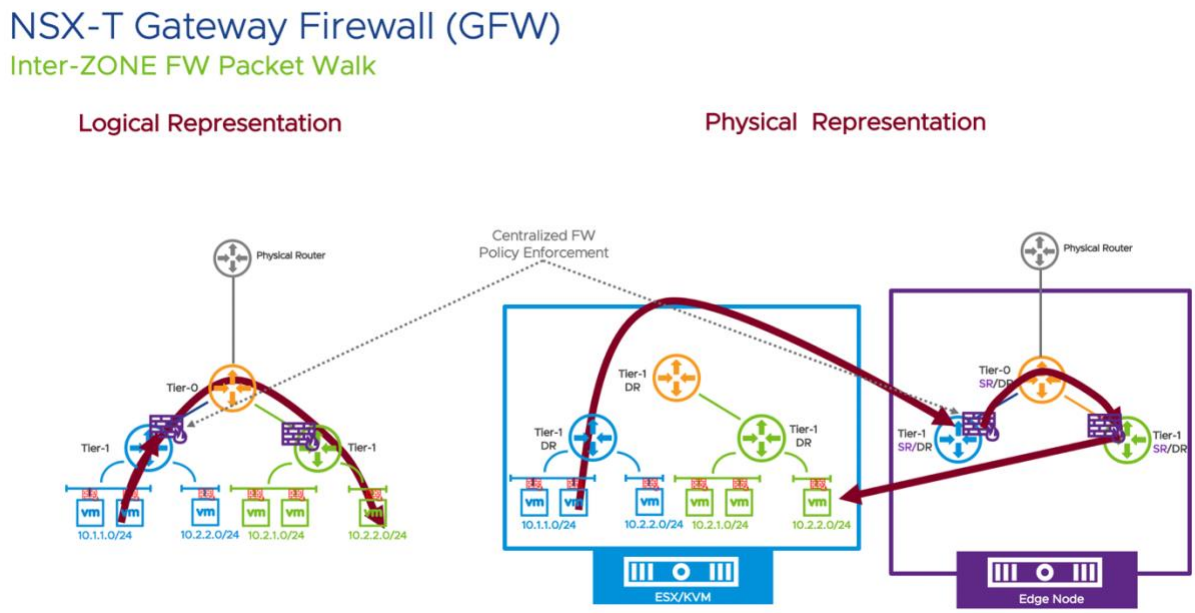

Figure 4 - 6 NSX-T Gateway Firewalls, Inter-tenant Traffic

In figure 3.7, traffic originating inter-tenant traffic from the Prod zone is going to the Non-Prod zone. Again, this is depicted both logically and physically. The traffic originates at the VM on the

10.1.1.0/24 segment of the Prod (blue) zone and is destined to the VM on the 10.2.2.0/24 segment in the NonProd (green) zone. Assuming the packet is allowed out the DFW on the VM, it then goes to the Prod T1 Gateway which resides on the Edge Node. At the Prod T1 Gateway Firewall, it hits a rule that allows the web_prod to talk to the Non_Prod Dev_Test segment. From there, the packet goes to the To Gateway Firewall which allows Prod to talk to NonProd. Finally, it will hit the NonProd T1 gateway firewall which allows the in with a rule that says web _servers can talk to the Dev_Test Segment. Once again, each Gateway firewall has rules relevant to its scope.

4.2.2 Gateway Firewall Functions

The Gateway Firewall is where state-sensitive services such as NAT, DHCP, VPN, and LB are implemented. One of the differentiating services which is available with NSX security is the full security suite of services functionality available from our Advanced Load Balancer.

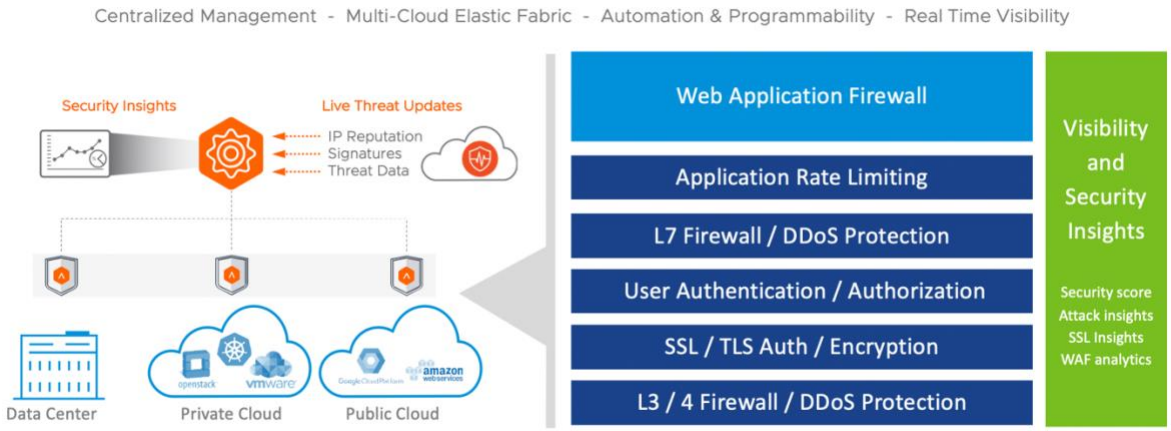

Figure 4 - 7 Advanced Load Balancer Security Service Suite

WAF (Web Application Firewalling) is one part of the security stack within the Advanced Load Balancer (ALB). Obviously, the ALB provides load balancing services, global load balancing. On top of that, though, there is a security stack that can be applied to applications ranging from the basic layer three/four firewalling all the way up to SSL termination. The Advanced LB also offers authentication and authorization via integration with SAML. Next there is layer seven firewalling - the ability to have firewall rules on HTTP headers, url and so on. There is also DDoS protection at layer seven for application attacks like Slow Loris, built into the platform as well. To complement the L7 security, there is comprehensive rate limiting. This provides the ability to rate limit both connections and requests in a fairly granular way all the way down - if you need to - to individual clients or per URL. Finally, on top of all of that security, there is the web application firewall which is part of that LB Service Engine. It is not a separate component and not a separate feature or license. It is literally a policy that you assign to an application when you deploy it and that application is then protected by the WAF. As you change the LB URLs for that application, the WAF is automatically learning those changes.

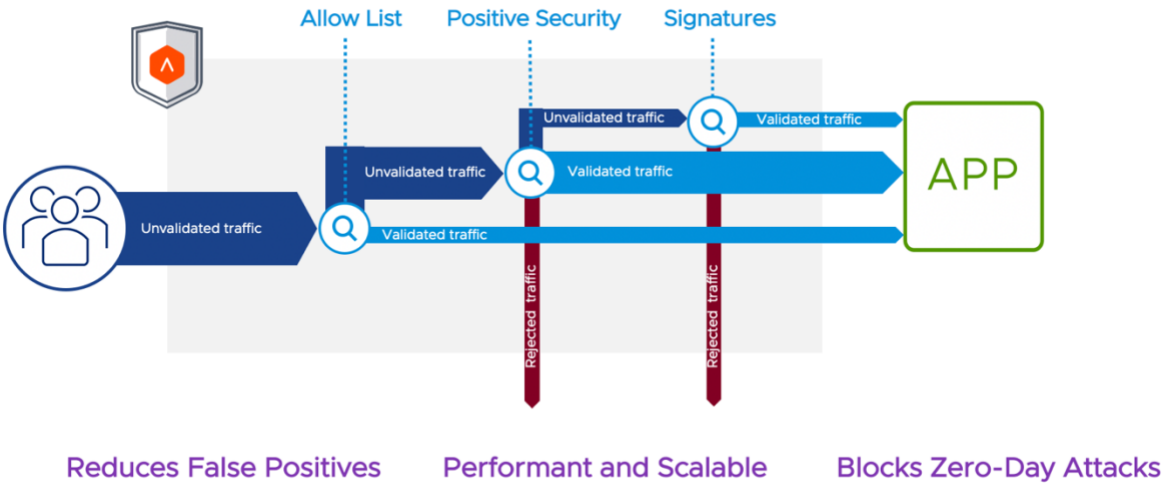

Figure 4 - 8 WAF Security Pipeline

Error! Reference source not found. shows the full WAF security pipeline, which has been designed with optimal security and efficiency in mind. WAF checks include HTTP checks (enforcing the HTTP standard), encoding bypass checks (multiple encoding attempts), and even restricted files or extensions (such as forgotten *.bak* files, for example). Walking through this pipeline, the first pass is an allow list of things which are known good. The next step is Positive Security with its learning input which checks a high percentage of all parameters, therefore reducing the impact of the last step: signature checking. Each step is designed to cull traffic for the following, more computationally expensive step. All learned and enforced tragic by the positive security engine reduces traffic for the signature checks, which are the most expensive. Since generic signature checks are the most common for false positives, reducing the traffic on which they operate also reduces the false positive rate. The result of this inspection waterfall is that zero-day attacks are blocked, false positives are reduced, and WAF performance is optimized.

4.3 *Distributed Firewall*

As was mentioned above, the Distributed Firewall is an East-West Firewall. The DFW inspects every packet going in and out of a VM, as a function of a packet travelling along the virtual NIC (VNIC) of the VM. The DFW exists in the kernel of the hypervisor, which means it delivers line rate performance. Moreover, since it exists in the hypervisor, the DFW scales linearly with added compute. Most importantly, the DFW rules move with the VM during vMotion events. This means that traffic flow state is preserved, regardless of which host a VM moves to. (vMotion events will typically disrupt legacy firewalls deployed in a VM form factor.) Another key aspect of the distributed firewall is that it provides a central policy, enforced in a distributed manner. Chapter 4 will dive into the details of Distributed Firewall policy design.

One consideration for DFW optimization is IPv6. By default, the DFW has both IPv4 and IPv6 enabled for every firewall rule. For resource optimization, it is recommended to only enable IPv4 in firewall rules where IPv4 is the only protocol in use. This is done by clicking on the gear icon to the right of the rule, which brings up the configuration screen shown in figure 3.10. Note that this screen will also allow the enabling of logging and rule directionality definition. When IPv6 is selected, it is

important to note that NSX-T IPv6 resolution is enabled by default and IPv6 learning is disabled. (The opposite was true with NSX-V.). The IPv6 settings are adjusted in the Networking section.

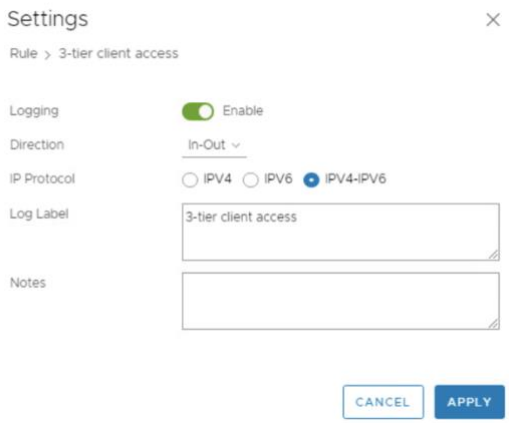

Figure 4 - 9 Rule Settings for IPv6

In Figure 4 - 10 NSX-T Distributed Firewall, the DFW being used to create zones. The red VMs and containers are the DMZ zone; the green VMs and containers are the Prod zone; and the blue VMs are the NonProd zone. Note that all of these VMs can comingle on the same segments in the same host and be secure with DFW policy, without the need to change the underlying network infrastructure. The gray services zone happens to be all on the same segment (because luck occasionally shines). This design allows the creation what is called a "DMZ Anywhere" design. A DMZ no longer means

stranded compute capacity, nor does it require the backhaul of DMZ traffic across the DC for security treatment.

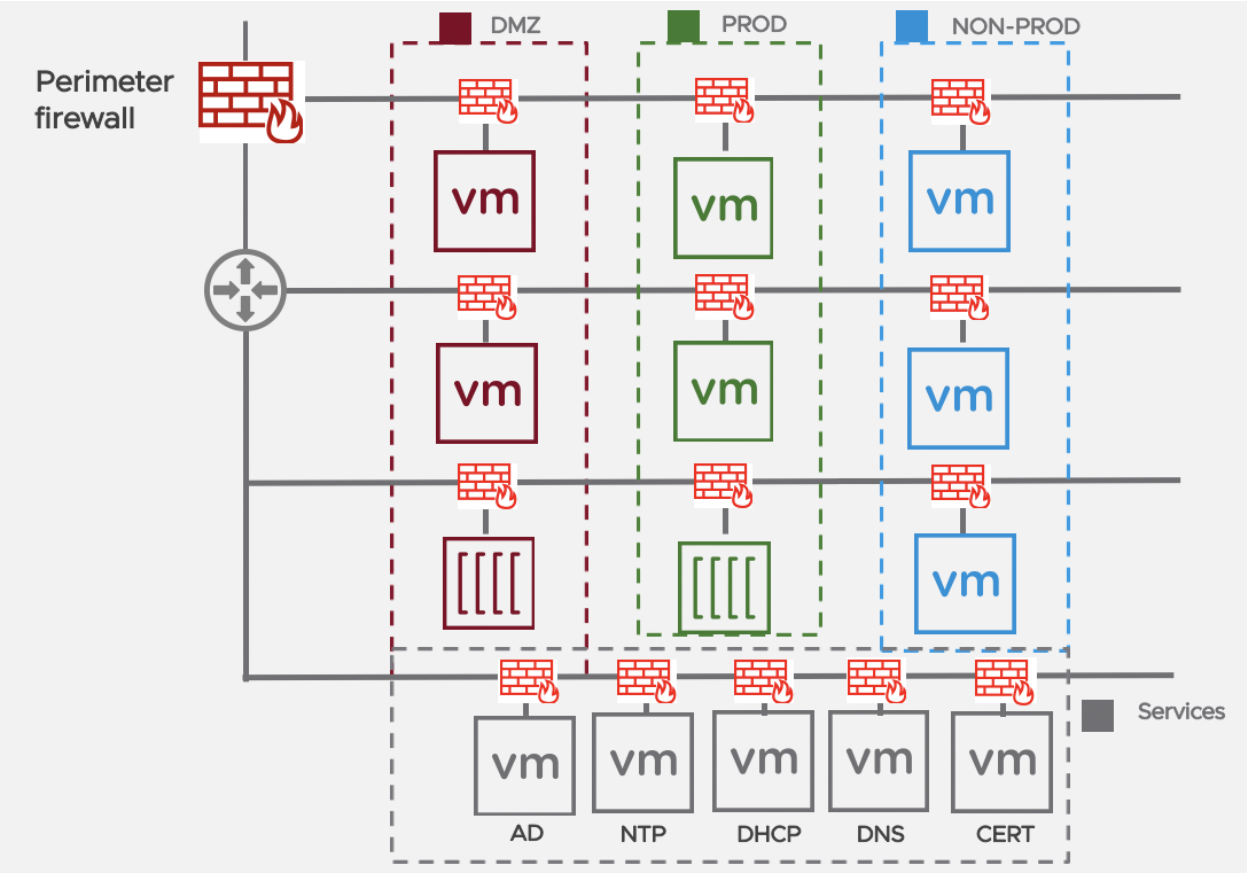

Figure 4 - 10 NSX-T Distributed Firewall

4.3.1 Zone Firewalling with the Distributed Firewall

Revisiting the interzone communication above where the VM on the blue zone communicates to the VM on the green zone, as shown in Figure 4 - 10 NSX-T Distributed Firewall. But, this time, one

examines that flow without any Gateway firewall rules in place. In this example, the T0 and T1 routers exist only for the purposes of routing.

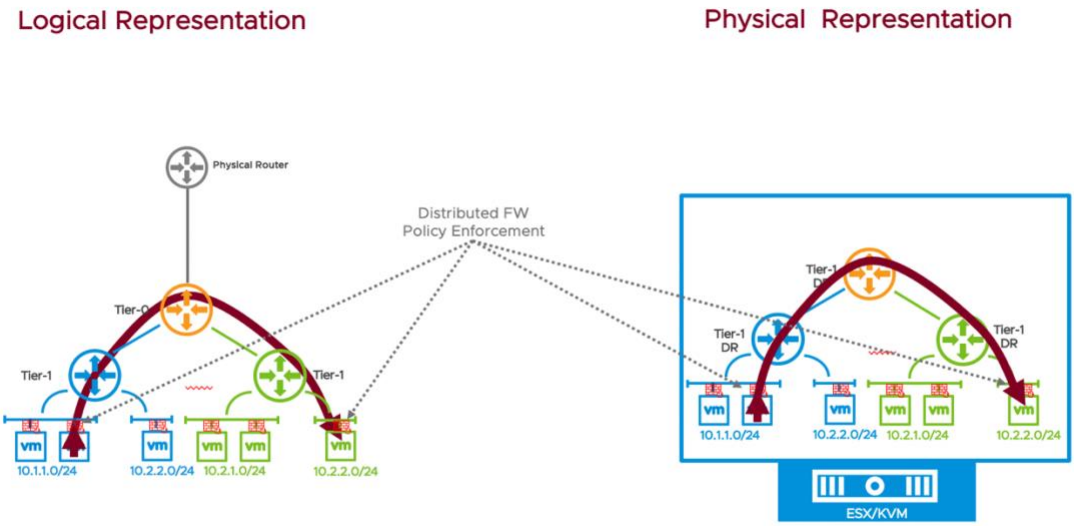

Figure 4 - 11 NSX-T Distributed Firewall Physical and Logical Representation

As explained earlier, the packet leaving the VM must traverse the DFW on the VM. This means that the DFW must allow that protocol out. In this case, there is a rule that allowing that. Because of the magic of distributed routing (described in detail in the NSX Design document), the packet never leaves the host but appears at the destination VM which coincidentally lives on the same host. The packet now arrives that the destination VM's VNIC, but it must go through the destination VM's DFW. Here, again, a rule that allows the packet in.

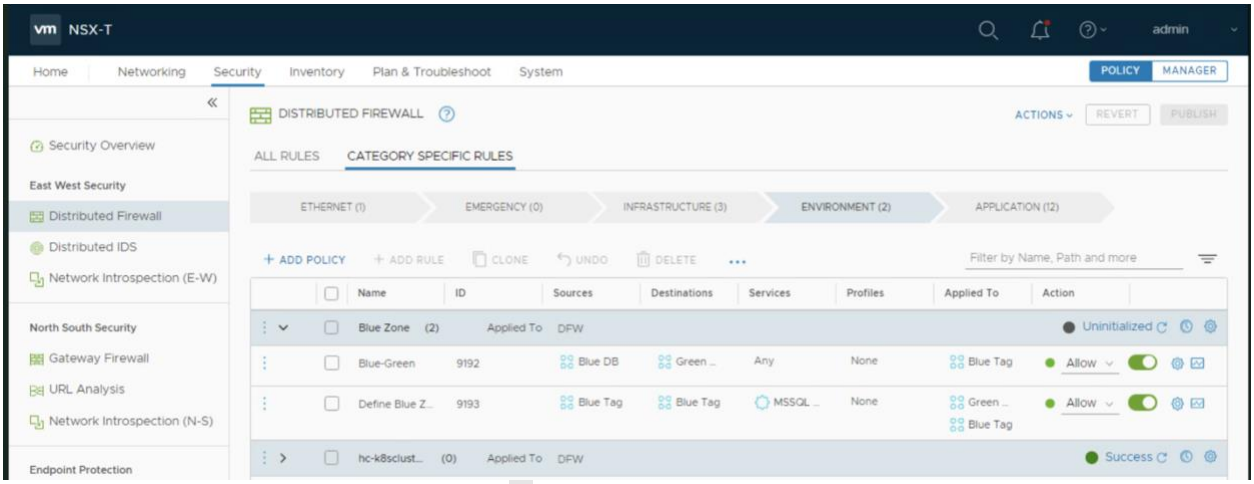

Figure 4 - 12 NSX-T Distributed Firewall GUI

Figure 4 - 12 NSX-T Distributed Firewall GUI shows a sample policy that would define a blue Zone then add a rule for exceptions allowed out of the zone.

5 NSX Firewall Policy Building

While one CAN build NSX policy in the same manner that legacy firewall policy has been built for years, the history of VMware support cases shows that not to be the best idea as one get to large scale environments. One of the most common problems seen by support is temporary measures which last far beyond their intended period, only to cause massive problems down the road. Moving to an NSX firewall model is an opportunity to start fresh, with all the lessons of the past, to build a better policy. It is advised against porting legacy firewall policies to NSX. Can it be done? Sure. It can. And the policy will work. SHOULD it be done? Not if a solid, long-term solution is the goal. VMware professional services have worked with many customers to migrate their policy, but the key to the success of those engagements has been the translations and optimizations that took place to make the resulting policy optimized for NSX. Importing a legacy firewall config into NSX without translation is like putting a gas engine into a Tesla. It can be done, and it will work for transportation, but the differentiating value is lost.

The NSX-T transport nodes make a distributed data plane with DFW enforcement done at the hypervisor's kernel level. Each of the transport nodes, at any given time, connects to only one of the Central Control Plane (CCP) controller based on mastership for that node. Once the Local Control Plane (LCP) has the policy configuration from the CCP, the LCP pushes the firewall policy and rules to the data plane filters (in kernel) for each of the virtual NIC's. The LCP programs rules only on virtual NICs based on the contents of the Applied To field. instead of every rule everywhere (optimizing the use of hypervisor resources). In policies ported from legacy firewalls, the Applied To field is a concept that does not exist with any greater granularity than a whole firewall; in NSX, the Applied To field can limit policy down to a cluster, host, VM, or even an individual vNIC (each greatly reducing the size of the ruleset applied). Thus, the ported policy is substandard right off the bat.

NSX Distributed Firewall Policy is shipped with a permit in the default rule. This is because of the potential Denial of Service that a default deny would imply in an East-West environment. The action of the default rule can be modified to drop or reject (ICMP Unreachable is sent).

Although some of the same concepts in building legacy firewall policy apply, there are new constructs available in building NSX firewall policy which can make the resulting implementation run more efficiently. This chapter examines the new constructs of building virtual firewall policy.

5.1 *Rule Lookup*

NSX firewalls implement a top down rule search order. When a packet matches, it pops out of the search based on the processing indicated in the matched rule.

By default, the DFW implements the rule table and flow table model that most firewalls use. However, this behavior can be overwritten for troubleshooting or other corner cases as described later.

In figure 4.1, the processing of a packet takes place as follows:
An IP packet identified as pkt1 that matches rule number 2. The order of operation is the following:

1. A lookup is performed in the connection tracker table to determine if an entry for the flow already exists.
2. As flow 3 is not present in the connection tracker table, a lookup is performed in the rule table to identify which rule is applicable to flow 3. The first rule that matches the flow will be enforced.
3. Rule 2 matches for flow 3. The action is set to 'Allow'.

4. Because the action is set to ‘Allow’ for flow 3, a new entry will be created inside the connection tracker table. The packet is then transmitted out of DFW.

Subsequent packets are processed in this order:

1. A lookup is performed in the connection tracker table to check if an entry for the flow already exists.
2. An entry for flow 3 exists in the connection tracker table. The packet is transmitted out of DFW

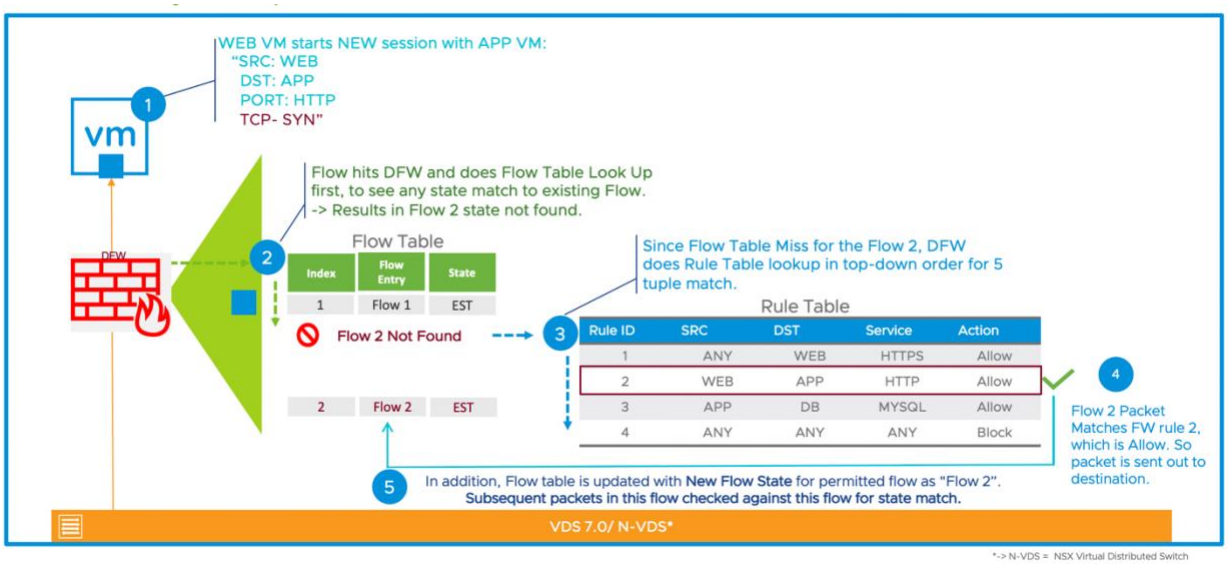

Figure 5 - 1 NSX-T Rule Processing

5.2 *Applied To Field*

THE most important best practice, the one that addresses the majority of calls into VMware support due to policy suboptimization, is the use of the Applied To field. To reiterate: USE THE APPLIED TO FIELD!! So, what is this magical applied to field and how can it help? Applied To is the filed that indicates which vnics will receive the rule in question. It limits the scope of a given rule. By default (DFW in the Applied To field), every rule is applied to every vnic. This means that if there are 1,000 VMs in the environment and a rule allows VM A to talk to VM B, all 1,000 VMs will receive that rule instead of just A and B, or 998 VMs too many.

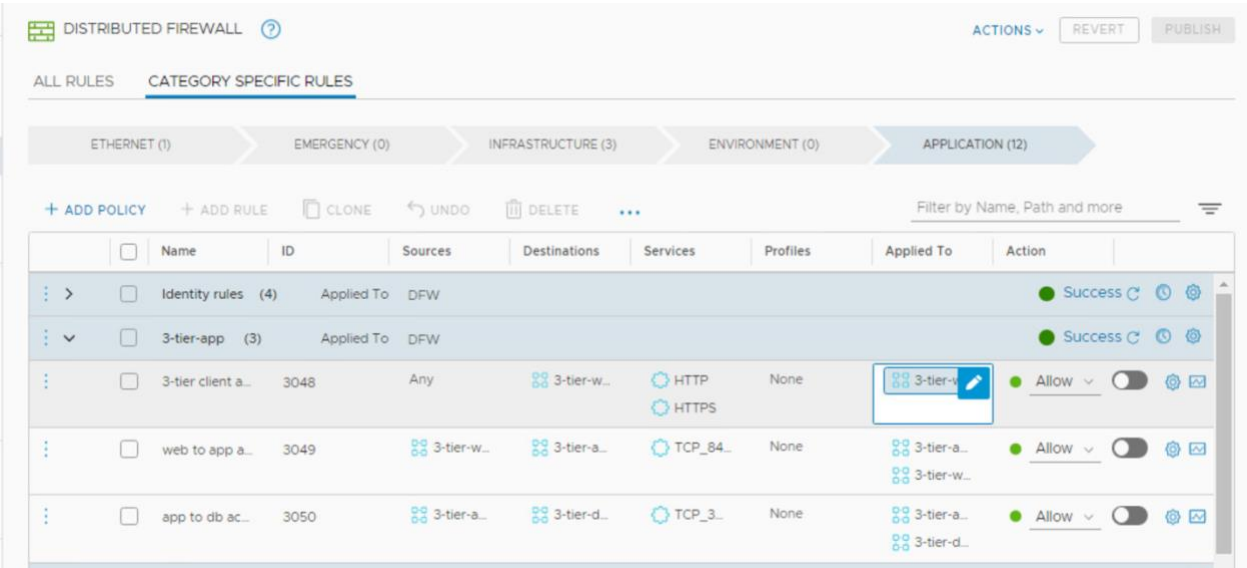

Figure 5 - 2 The Applied To Field

Figure 5 - 2 The Applied To Field, shows a three tier application, "3-tier", which has a web, app, and DB tier. The first rule is applied only to the web servers. The second rule is applied both to the web and app servers. The third rule is applied to both the app and db servers. The following rule of thumb clarifies what to put in the applied to field:

- If the source is any (external to the NSX environment), apply the rule to the destinations only. (This is what we saw above.)
- If the source is any, and can include sources within the NSX environment, apply the rule to everything (DFW).
- If the source and destinations are clearly defined in the rule, apply the rule to BOTH the source and the destination.

That simple step will set your NSX policy off on the right foot.

It is important to note that when there is a multitenant environment (especially with overlapping IP addresses), the use of the Applied To field is critical. In this case, typically assets are tagged with their tenancy.

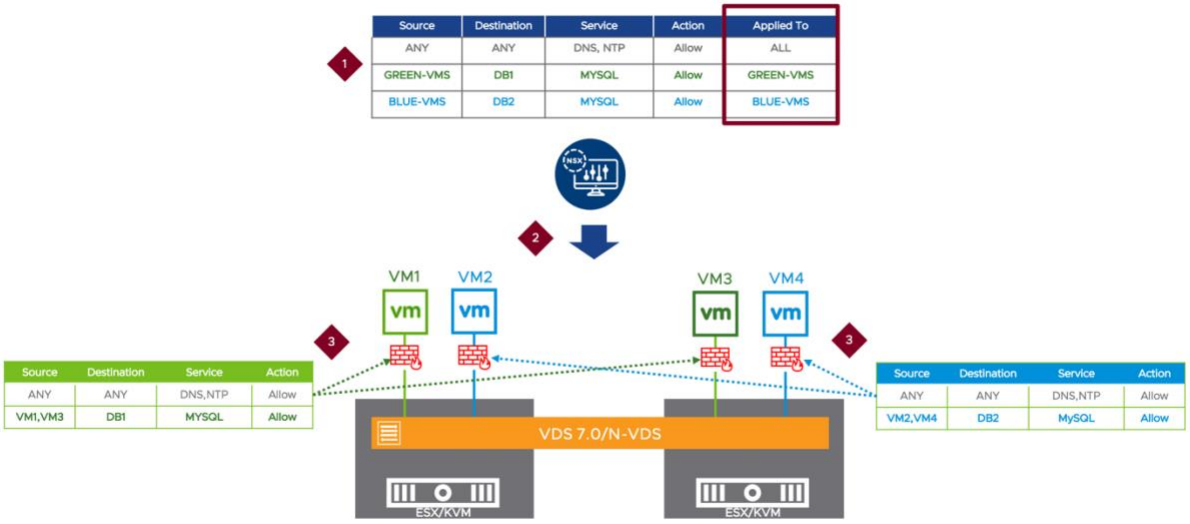

Figure 5 - 3 Applied To Field in Action

In Figure 5 - 3 Applied To Field in Action, the reward of using the Applied To field is evident. The policy allows the green VMs to talk to each other and the blue VMs to talk to each other. The applied to field is used in both rules. This full policy is built in the NSX management appliance and sent to the LCP on the hosts. The VSIP module will instantiate only the green policies on the green VNICs and the blue policies on the blue VNICs, based on the contents of the Applied To field. Note that the default behavior of the Applied To field, *DFW*, means that the rule will be implemented in everything. As policies grow to thousands of entries, the Applied To field becomes critical for scale. However, retrofitting the Applied to field is extremely challenging, so the use of the Applied To field is critical from the outset.

It is important to note that there is a Policy Level Applied To field and a Rule Level Applied To field. The Policy Level Applied To field will OVERRIDE the Rule level Applied TO field, except in the case where the Policy Level Applied To field is DFW (the default). In that case, the Rule level Applied To field is honored. Any other entry in the Policy Level Applied To field will override the Rule Level Applied To field:

Figure 5 - 4 Applied To Field - DFW (Default)

Figure 5 - 4 Applied To Field - DFW (Default) above shows the default and least desired configuration of the Applied To field.

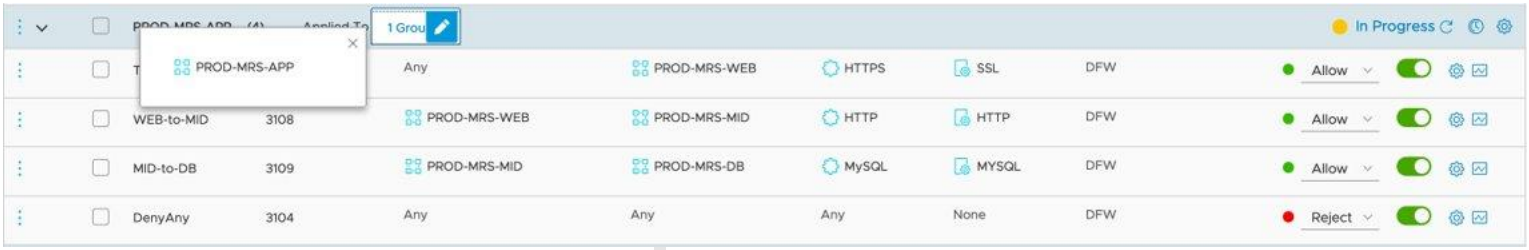

Figure 5 - 5 Policy Applied To Field

Figure 5 - 5 Policy Applied To Field above shows the rules applied only to the PROD-MRS-APP group.

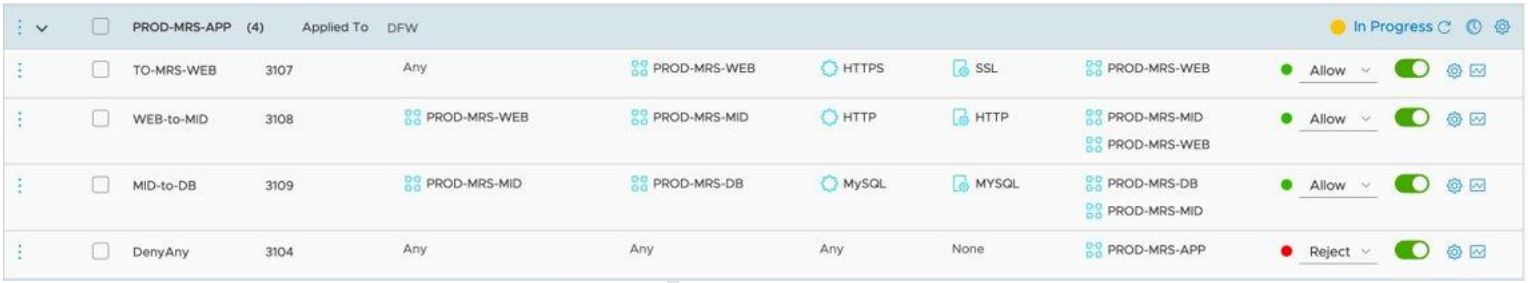

Figure 5 - 6 Rule Applied To field

Figure 5 - 6 Rule Applied To field above shows the Rule Applied TO field being used to further limit rule sprawl. In this case, the first rule applies to the PROD-MRS-WEB group, the second rule applies to both the PROD-MRS-MID and the PROD-MRS-WEB groups, the third rule applies to the PROD-MRS-DB and the PROD-MRS-MID groups and the last rule only to the PROD-MRS-APP Group.

In *Figure 5 - 7 Policy Applied To Overriding Rule Applied To*, all rules apply to the PROD-MRS-APP group for all rules, overriding the Rule Applied To fields – as stated above.

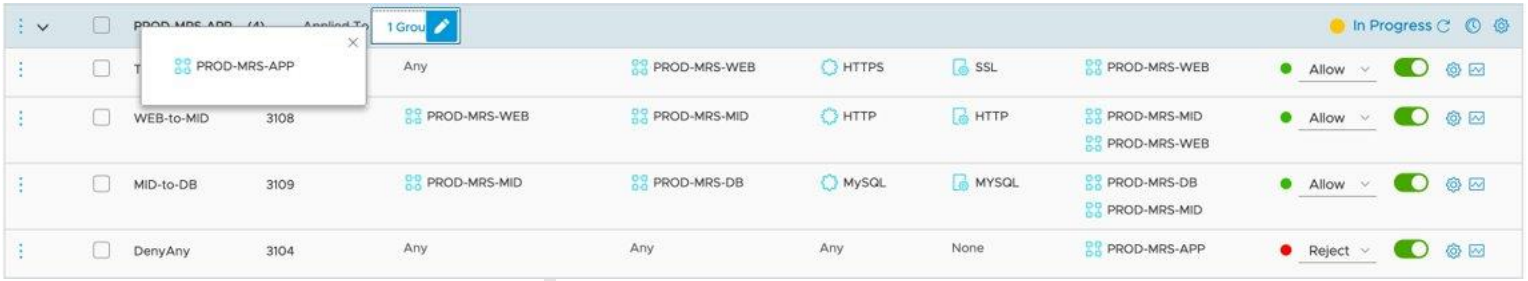

Figure 5 - 7 Policy Applied To Overriding Rule Applied To

One last note about the Applied To field with IP groups that have IP Addresses. NSX allows for overlapping IP addressing in different tenants (say, tenant A uses IP address 10.1.1.1. and Tenant B uses that same IP address 10.1.1.1 but referring to a different endpoint). This means that when the IP address is entered into the Applied To field, it is impossible for the NSX LCP to know which instance is referenced. So, to use the Applied To field in this case, it is necessary to create a group with the relevant segment(s) for use in the Applied To field. Granted, that may be larger than the anticipated scope if there is only one or 2 relevant IP addresses in the segment in question, but that is still a smaller scope than the entire environment.

Note:
Groups consisting of only IP addresses, MAC Addresses, or Active Directory groups should not be used in the Applied To field. If used, the rule/policy will be ignored and not be applied to any of the workloads, as that group doesn't have any segment-port members. The group used in Applied-to should result in one or more segment-port members. The group using dynamic criteria Tag/VM name, Segment, Segment-port, VM, etc. would result in segment-port members.

5.3 *Scale*

NSX Scale is greatly enhanced by the Applied To field described above. The current release scale limits are defined at https://configmax.vmware.com/. There are 3 aspects of rule counts that affect system scale: System Wide Rule Counts, Per VM VNIC Rule Count, and ESX Host Rule Count. The three are defined as follows:

1. **NSX System Wide Rule Count:** This refers to the total rules configures on the NSX Manager, across all different Tenants/Zones/Apps (6.5K rules, in the figure below).
2. **Per VM VNIC Rule Count:** This is what is optimized by the Applied To fields, as described above. (Blue = 1.5K, Green = 2.5K, and Blue = 3.5K below)
3. **ESX Host Rule Count:** This is the sum of all VNIC rules across all VMs on a given host. (ESX-1 = 15K, ESX-2 = 21K below)

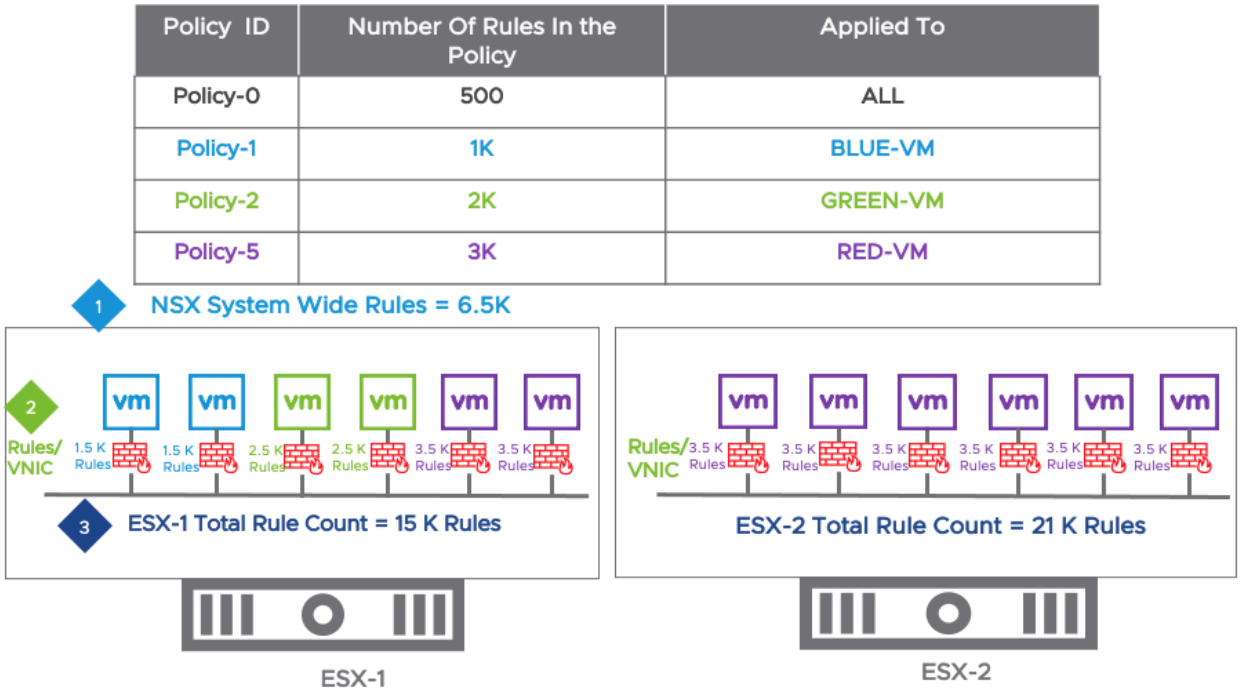

Figure 5 - 8 System, VNIC, and Host Rule Counts

There is a script on Github (https://github.com/vmware-samples/nsx-t/blob/master/helper-scripts/DFW/nsx-get-dfw-rules-per-vm.py) that can be used to get per vnic rule counts. This script can be run with the --aboverulelimitonly yes and --fwrulelimit N option where N is the number of rules desired.

5.4 *Grouping*

Groups are a very useful tool for defining the source or destination in a rule. While the grouping concept is trivial (one term used to describe many objects), the use of groups can be made optimal if best practices are known at the outset.

The following concepts apply in deciding the proper grouping construct:

- IP Block / CIDR / Infrastructure constructs per environment are typically static.
    - Most organizations have different CIDR blocks for their prod and non-prod environments, for example. When that is the case, it is optimal to use the CIDR block as a grouping construct.
    - When adding IP Addresses to a group, you can import a *txt* file. (This allows for noncontiguous IP Address ranges.)
- Have broader groups like Environment/Zone more statically using IP Subnets/ Segments
- Application/Application Tier level grouping should use dynamic grouping with VM Tags or VM name or combination
- Nested groups should limit to 3 levels of nesting for manageability and resource optimization
- When using dynamic grouping with multiple AND/OR criteria, limit the complexity of the criteria for the same reasons as well as to limit the number of unexpected members.
- When possible, use tag/name "Equals-to" to limit the number of unexpected members.

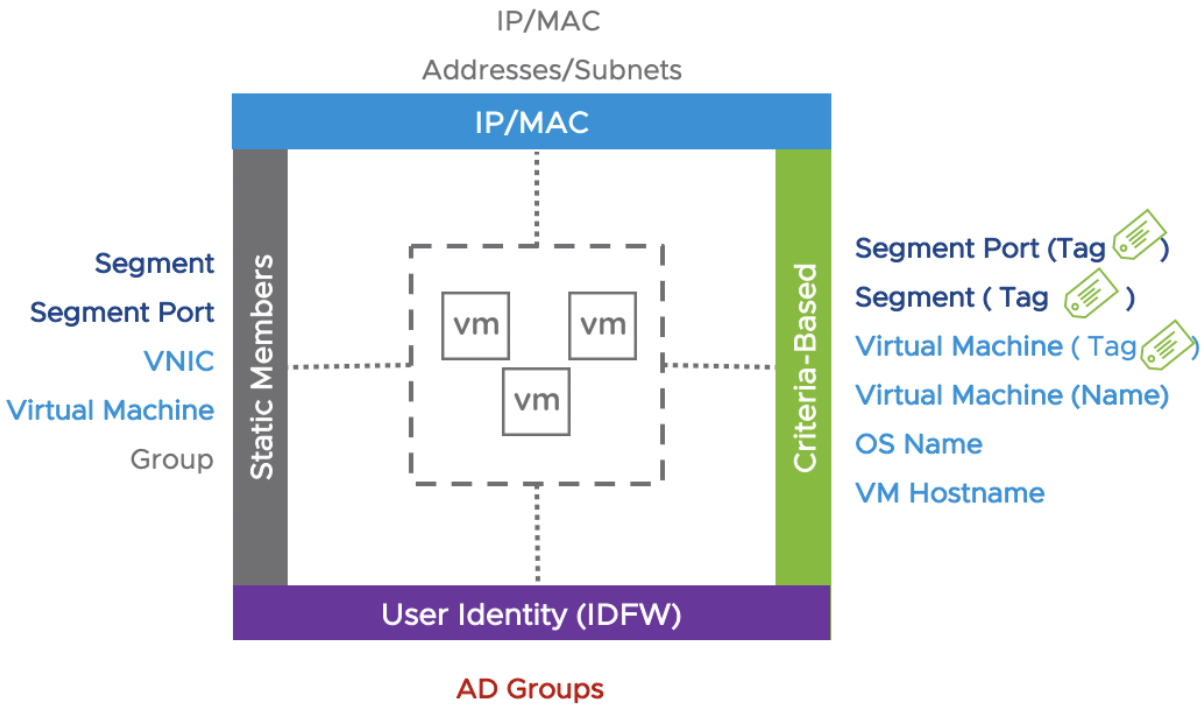

Figure 5 - 9 NSX-T Groups

With security, there is a balance between agility and dynamic membership and security. Many new installations like to use regex to create groups. Although this is supported, it is highly recommended (from a security perspective) that this be done to create initial groupings which can be reviewed for accuracy, then static groups be created at least for sensitive groupings. When there is a desire to have automated security, tags are a much better way to go than groupings with complex membership.

Few things to remember with dynamic criteria usage with Tags:

- Match criteria “Scope & Tag” is an “and” operation within a criteria.
- Blank value in scope or tag within a criteria – will ignore the empty field for computation.
- Both scope & tag field cannot be blank in a criteria.
- Match criteria with operator other than Equals to (contains, starts-with etc) - Scope is always uses equals to, however Tag value will be evaluated with the used operator.
- Matching criteria needs either Scope: Tag with some value. You can group only based on tag or group based on Scope only.
- Uses 5 AND or OR operations to group workloads within a Group.

Using Nested Groups

Groups can be nested. A Group may contain multiple groups or a combination of groups and other grouping objects. A security rule applied to the parent Group is automatically applied to the child Groups. Nesting should be limited to 3 levels, although more are supported. This is to ease troubleshooting, minimize unintentional policy results, and to optimize the computational burden of publishing policy. Nothing prolongs downtime like trying to follow the logic of a grouping nested 5 levels deep.

In the example shown in Figure, three Groups have been defined with different inclusion criteria to demonstrate the flexibility and the power of grouping construct.

- Using dynamic inclusion criteria, all VMs with name starting by "WEB" are included in Group named "SG-WEB".
- Using dynamic inclusion criteria, all VMs containing the name "APP" and having a tag "Scope=PCI" are included in Group named "SG-PCI-APP".
- Using static inclusion criteria, all VMs that are connected to a segment "SEG-DB" are included in Group named "SG-DB".

Nesting of Group is also possible; all three of the Groups in the list above could be children of a parent Group named "SG-APP-1-AllTier". This organization is also shown in Figure.

Figure 5-10: Group and Nested Group Example

5.5 *Tags*

Tags is what all the cool kids are doing in security. Why? Well because tags accelerate automation, apply policy when the workloads are provisioned, allow for policy definition apart from application, AND they prevent rule sprawl (when used properly). What more could you ask from a nifty software construct?

(It is important to call out here that this document refers to NSX-T security architecture. The tagging approach described below is an example of the differing architecture between the two platforms. Should this approach be applied to an NSX-V implementation, serious performance penalties may be experienced due to architectural differences between the two platforms.)

Here are key benefits of Tag based policies:

- Dynamic Group/Policies based on Tags
- Not tied to IP (v4/6) or physical or logical network topology
- Automated Security Policy enforcement and lifecycle for new applications being provisioned
- Granular dynamic policies, specific to individual applications tier, application, Zone or Tenant
- Easily replicate and automate security for different environment.

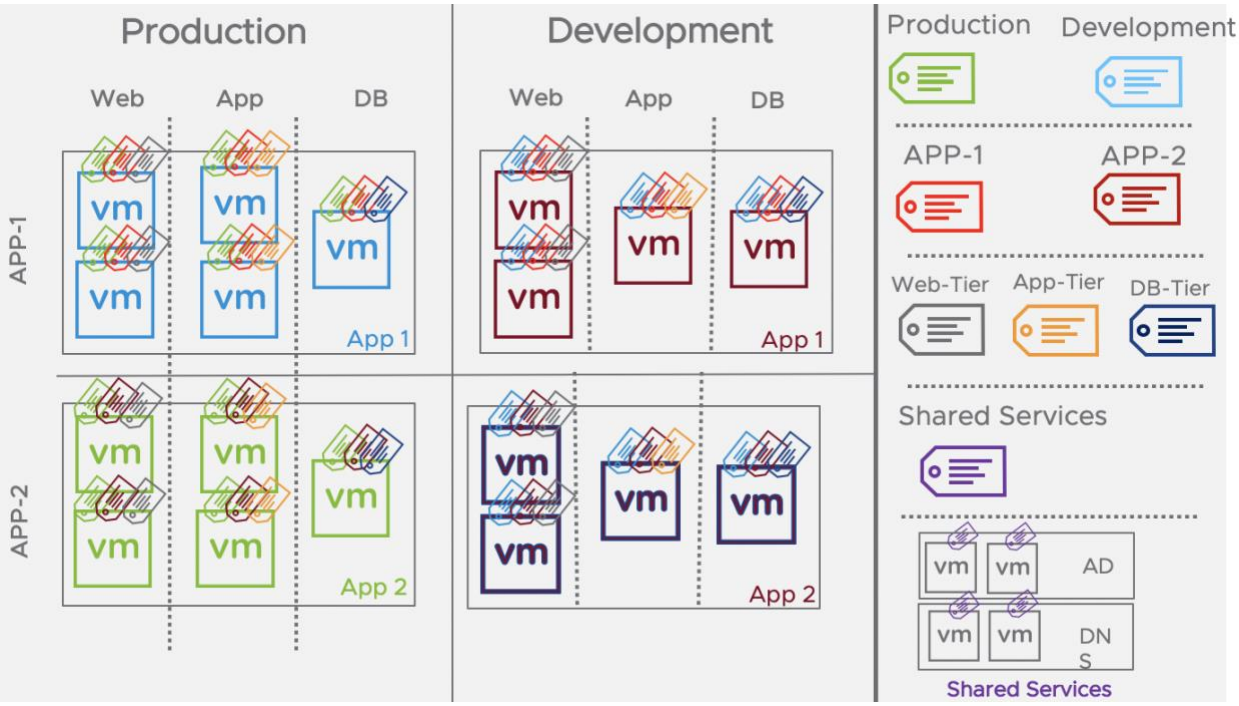

Figure 5 - 11 NSX-T Tags

Tags are a security wonder because security is ***automated!*** This means that *if* one service finds something, ***then*** another service can do something about it. Tagging also provides a security posture of a workload of VM. This can be either an intended posture or runtime posture. You can create custom tags to tag VMs. Third Party Services are required to tag on specific events. This helps to create automated workflows. For example, antivirus can tag a VM when it is found to be infected. By having predefined rules based on this tag, this allows for automated remediation.

Security Tag are applied to Physical Servers, Virtual Machines, Logical Ports, and Logical Segments and can be used for dynamic Security Group membership. NSX allows multiple tags per VM allowed, up to 30 to identify environment, zone, tenant, application, tier, OS etc. They can apply differentiated policy based on OS, Environment, or a myriad of other attributes. Tags are used to automate policy definition for new applications being provisioned. The tag scope is analogous to a key and the tag name is analogous to a value. For example, let us say, you want to label all virtual machines based on their operating system (Windows, Mac, Linux). You can create three tags, such as Windows, Linux, and Mac, and set the scope of each tag to OS. Other examples of tag scope can be tenant, owner, name, and so on.

5.5.1 Tags With/Without Scope

NSX tag is a Key:Value pair which uses Scope: Tag field to define a tag. Scope is an optional field. From grouping perspective customer can use either or both of them to group the workloads.
Tag with SCOPE helps to:

- Have proper Tag inventory management with both Key and value.
- Indirectly one can insert more tags/metadata than 30 NSX Tag, which is the tag limit per object
- Similarly, can have more than 5 AND/OR GROUP criteria indirectly, which is the limit otherwise

The best practice is, if the number of Tag and Group criteria requirements are within the NSX supported limit (true for most customers), then keep it simple, have multiple individual Tags with optional Scope. Scope can be

used to represent the Key for that tag, for example scope:tag can be defined like “region:us-west”, “enviornment:prod”, “app_name:hr_app” or “app_tier:web” or “os_type:windows”.

5.5.2 Multiple Tags vs Combined Tags

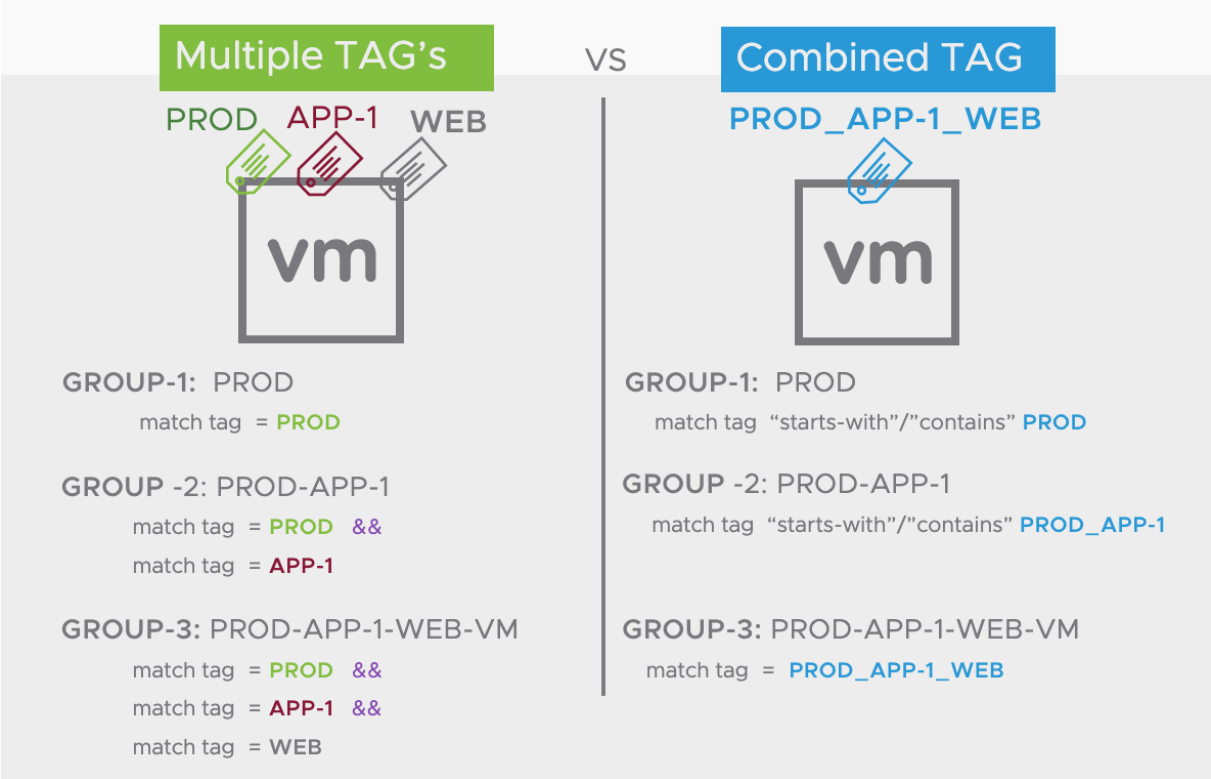

Figure 5 - 12 Compound or Combined Tag

Combined Tag helps to:

- Indirectly one can insert more tags than 30 NSX Tag, which is the tag limit per object.
- Similarly, it can have more than 5 AND/OR GROUP criteria indirectly, which is the limit otherwise.

However,

- Need to be planned well & less flexible if Tag needs to be updated
- Also need to use group regex (contains/starts-with/end-with) match criteria to build broader group, unlike with multiple tags where group definition always use && operator.
- Increases the number of Tags that need to be configured. For example, if a customer has 3,000 3-tiered applications in 3 environments (Dev, Test, and Prod). If this customer takes a compound tagging approach only (defining a tag for each combination), they will have 3,000 (applications) x 3 (environments) x 3 (tiers) = 27,000 tags requiring 27,000 or more rules. However, if they use three tags (one for environment, one for application, and one for tier), they would end up with 3,006 tags

The best practice is, if the number of Tag and Group criteria requirements are within the NSX supported limit (true for most customers), then keep it simple, have Multiple individual Tags without Scope, do not have combined Tag.

5.5.3 Tagging VM vs Segment vs Segment-port

NSX Tagging can be done on Virtual Machine or Physical Server, Segment or Segment-Port depending on the use case. The table below compares each options a customer needs to be aware of with respect to use case, grouping options, tag retention, and other tagging operations. This helps in understanding overall implementation and helps in having a better tagging strategy.

	VM Tag	Segment Tag	Segment port Tag
Grouping Option	Grouping criteria has: • Equals, • Contains, • StartsWith, • EndsWith	Grouping criteria is limited to one option: "Tag : Equals"	
Bulk Tagging **(tag multiple objects in workflow)**	Possible from UI & API	Not Available Today	
Inventory/Operations	Centralized- Simplifies operation. VM Tags have dedicated UI page- to tag all VM's	Non-Centralized- Little inconvenient to tag Segment/Segment-port, as it needs to be done from Segment/Port page/context	
When Tag can be Applied	As VM is created, NSX VM inventory would have that available for Tagging	Independent of VM creation or connecting. Part of Segment creation workflow	VM needs to be connected to a segment to assign a tag.
Use case	All Use Case with VM level grouping and Policy	• Broader grouping- ZONE/Tenant- With dedicated segments per ZONE/Tenant • VM with Multi-Home use case. Need separate policy per segment level	• Container plugin uses container-port tagging(label) for group/policy • VM with Multi-Home use case. Need separate policy grouping per VM VNIC.
Tag Retention	Retains Tag - Until VM is removed from the Inventory	• Retains Tag - Until Segment is removed from Inventory	• Retains Tag- until VM/Containers is removed or moved to another segment
Dynamic Grouping Members	Group would have all • VNIC/Segment-Port of the VM & • IPs of the VM	Group would have all • VNIC/Segment-Port connected to the segment • IPs of all workloads connected to this segment	Group would have **only** • VNIC/Segment-Port which is used • IP(s) of only that VNIC/Segment-Port

Figure 5 - 13 Tagging VM vs Segment vs Segment-port

5.5.4 Tags in Automation

In vRealize Automation, upon a blueprint deployment, all VMs part of an application are placed into a new Security Group. Also, every VM is tagged with multiple tags identifying Function, Zone, OS, Environment and Tenant.Tanzu also uses tags to define policy.

5.6 *NSX-T Policy Structure*

The NSX-T Manager UI has two different areas, one for the Distributed Firewall, and one for the Gateway Firewall, as shown in Figure 5 - 14 NSX-T Policy UI. The top portion, shown below, is for the Distributed Firewall, in the East West section. The gateway Firewall section is just below that, in the North South section. This layout reflects the findings that most customers spend the majority of their time in the East West section, as opposed to the North South section.

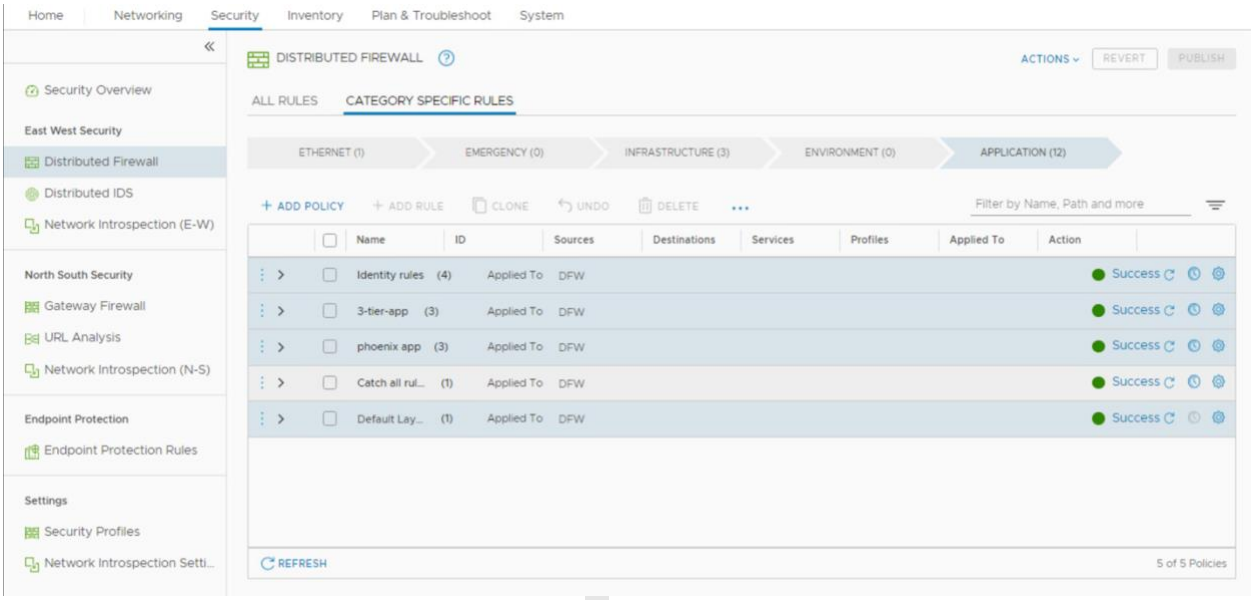

Figure 5 - 14 NSX-T Policy UI

Within each of these areas, there are categories which provide a means for organizing your security policy. Each Category is evaluated top to bottom, with the order of the categories being right to left as per the UI display. The categories of the Gateway and Distributed Firewalls will be examined below.

5.6.1 Gateway Firewall Policy Categories

NSX Firewall simplifies policy definition by having pre-defined categories. To match with common security policy best practices used by our customers like you. This helps in organizing the rules. As stated above, rules are evaluated top down within a category and left to right across categories. Category names can be changed using the API.

First look at the NSX gateway firewall and its predefined categories.

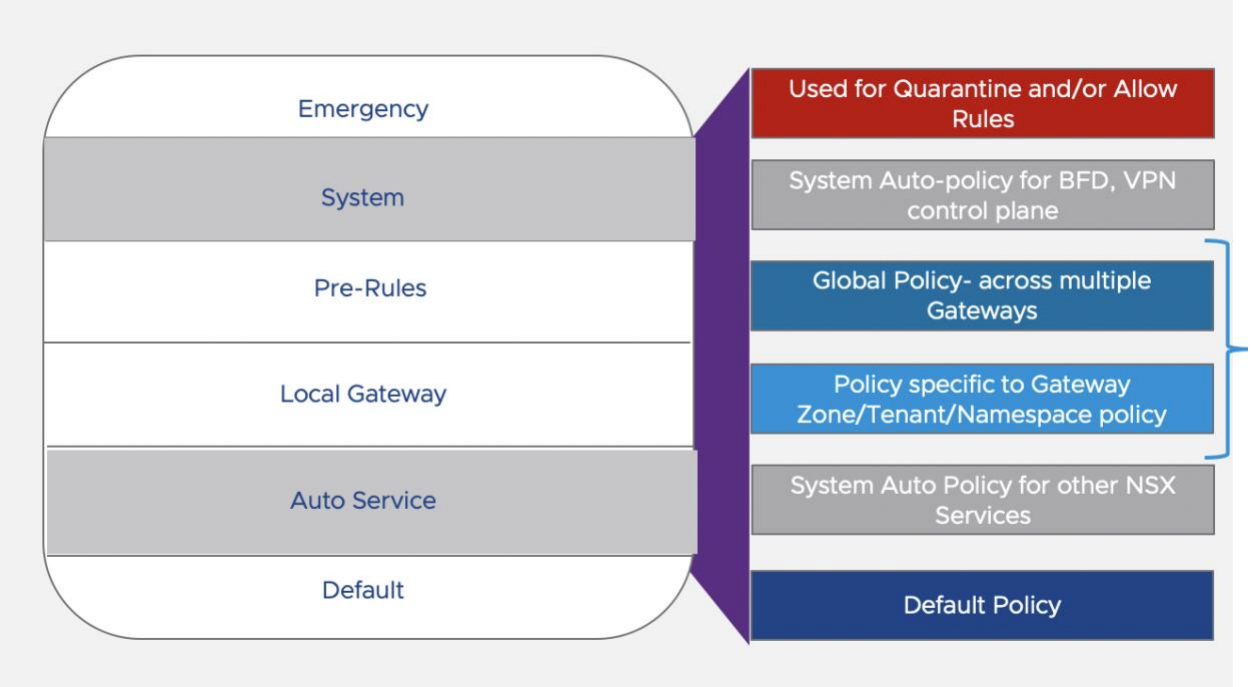

Figure 5 - 15 NSX-T Gateway Firewalls, Policy Structure

Emergency – This is used for Quarantine. It can also be used for Allow rules.

System – These rules are automatically generated by NSX and are specific to the internal control plane traffic (such as BFD rules, VPN rules, etc.) DO NOT EDIT SYSTEM RULES.

Shared Pre Rules – These rules are applied globally across all of the gateways.

Local Gateway – These rules are specific to a particular gateway.

Auto Service Rules – These are auto-plumbed rules applied to the data plane. These rules can be edited as required.

Default – These rules define the default gateway firewall behavior.

Most Gateway Firewall configuration will be done in the Pre-Shared and Local Categories. A good rule of thumb for the two categories would be that corporate policy lives in the Pre-Shared Rules while tenant/application policy lives in the Local rules.

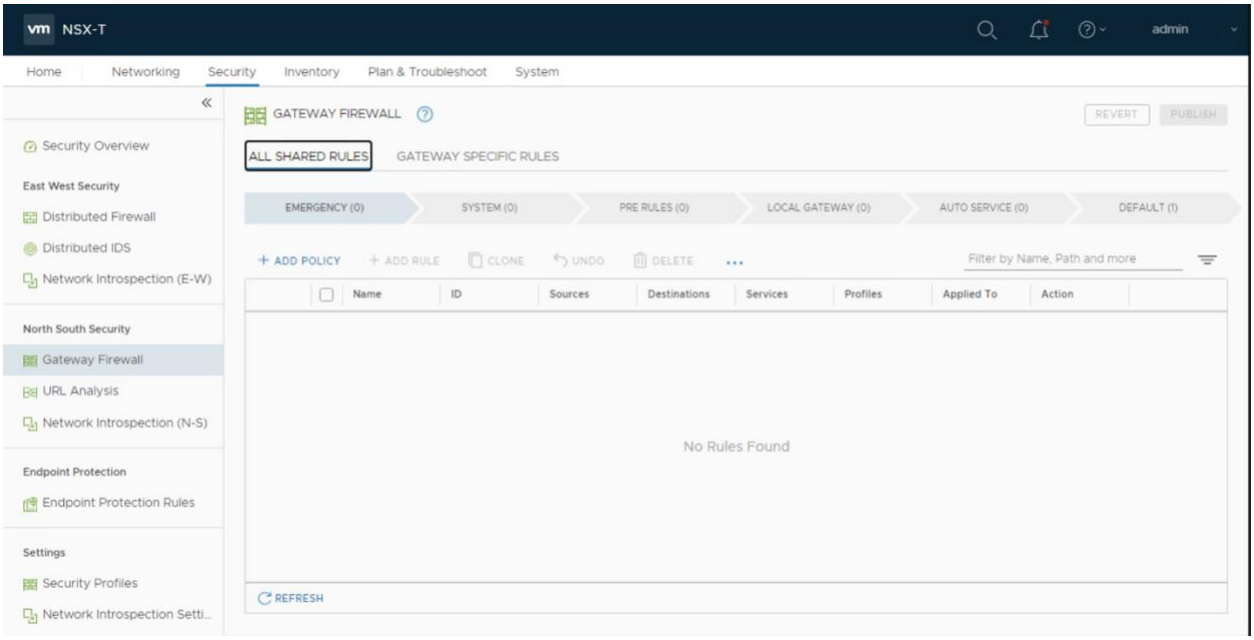

Figure 5 - 16 NSX-T Gateway Firewalls UI

5.6.2 Distributed Firewall Policy Categories

As with the Gateway Firewall rules, the rules in the Distributed Firewall are processed top down and left to right. Again, the category names can be changed via that API. As you can see, the categories are quite different from the Gateway Firewall. Those will be examined in detail.

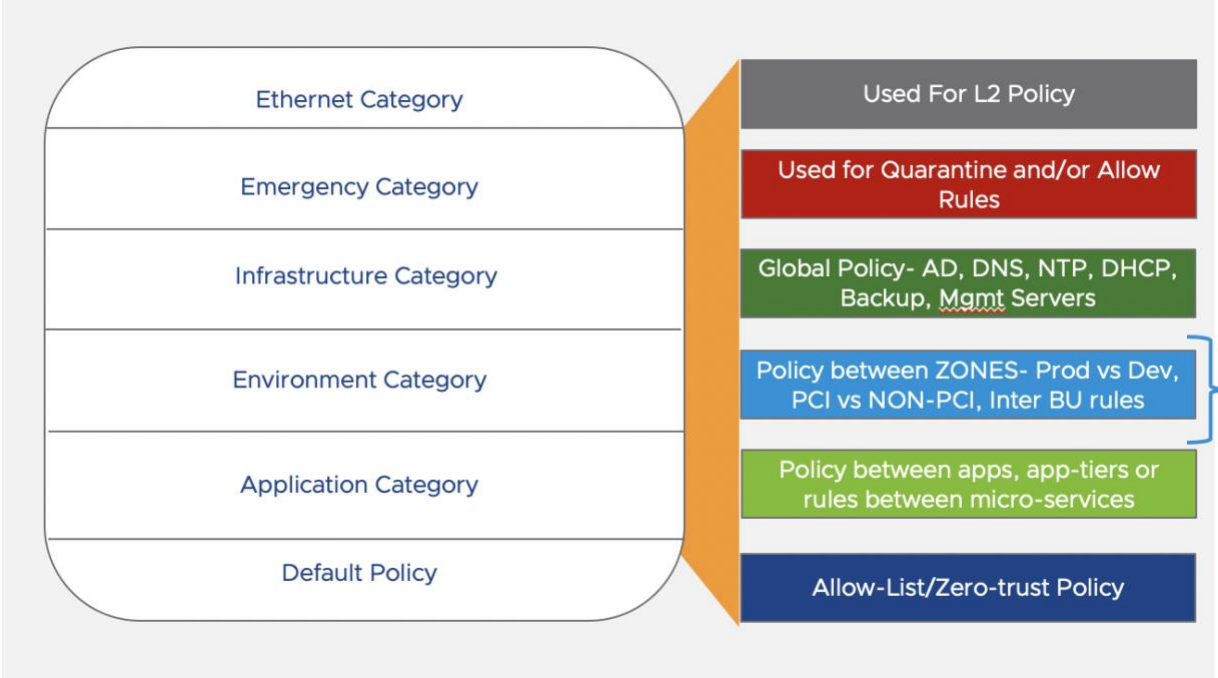

Figure 5 - 17 NSX-T Distributed Firewall Policy Structure

Ethernet – These are layer 2 rules based on MAC addresses

Emergency – This is the ideal place to put quarantine and allow rules for troubleshooting.

Infrastructure – These rules define access to shared services. Examples of rules in this category would be to allow AD, DNS, NTP, DHCP, Backup, Management access.

Environment – These are rules between zones. For example, allowing Prod to talk to Non Prod, or inter business unit rules. This is also a means to define zones.

Application – These are rules between applications, application tiers, or defining micro services.

Ideally, the top categories are less dynamic than the bottom categories.

In using the DFW for zoning, the environment can be used by creating ring-fencing policies. These are policies that create a ring around an environment. For example, the following policy creates rings around the Prod, Dev, and Test environments such that nothing is allowed out of those environments:

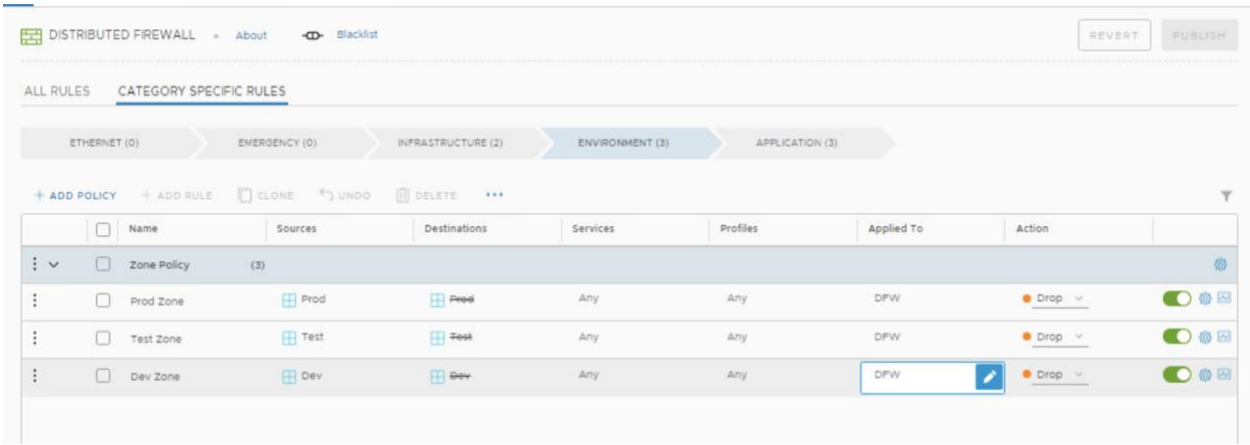

Figure 5 - 18 NSX-T Zone Policy

To create the rules, the group negation has been leveraged as shown below:

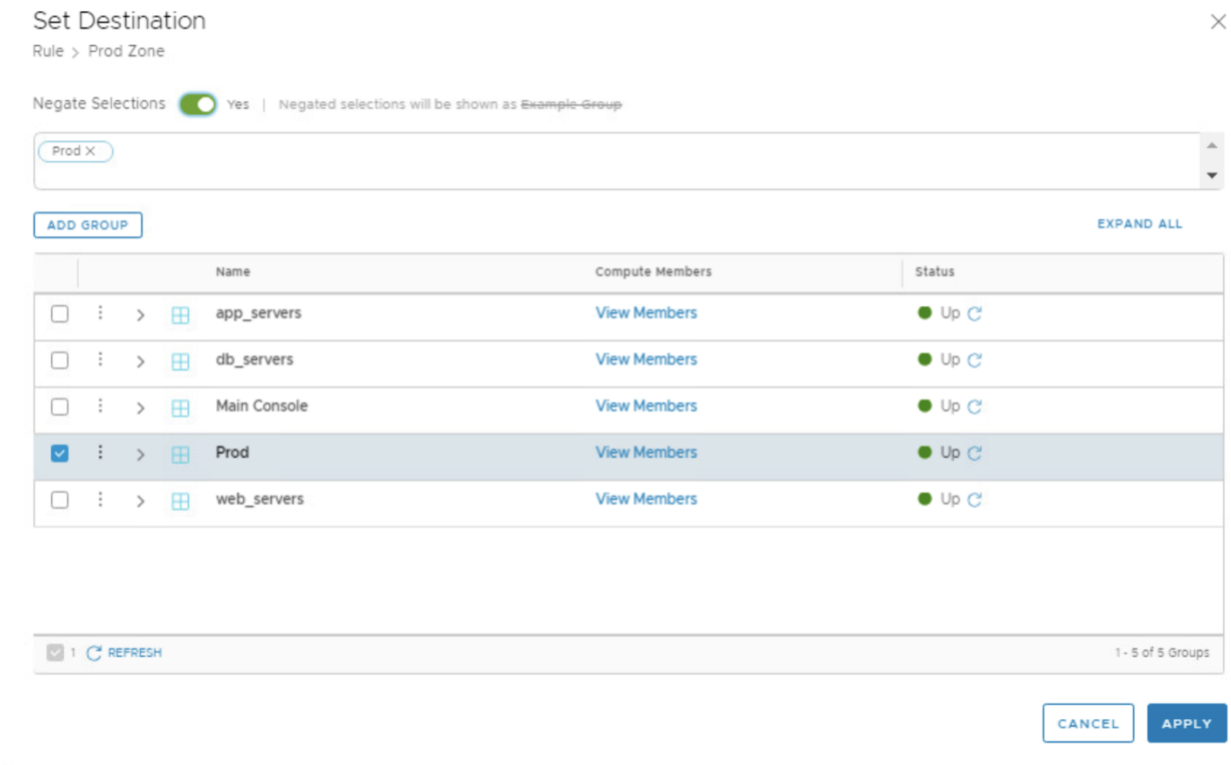

Figure 5 - 19 NSX-T Group

The only traffic to leave the environment section will be Prod traffic traveling within Prod, test within test, or Dev within Dev. Thus, the Zones have been established. As indicated above, the infrastructure section has already caught traffic that was DNS, LDAP, or other common traffic that would cross the zone boundary. If there are Zone exceptions, it is common to see a Zone exception Section before the zone policy as shown below.

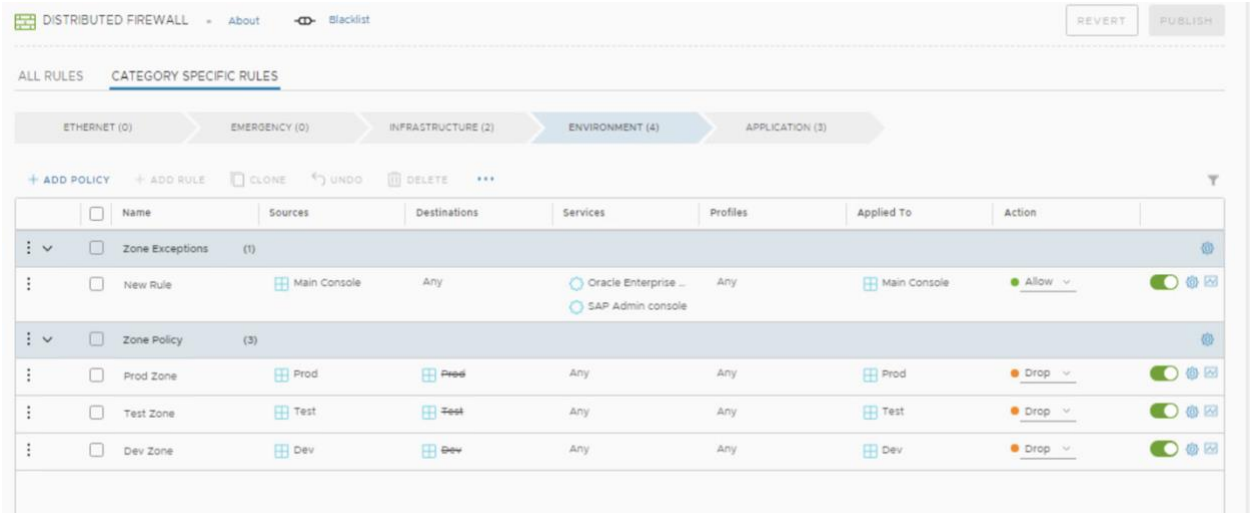

Figure 5 - 20 NSX-T Gateway Zone Policy Exception

5.6.3 Firewall Policy Drafts

The DFW allows for firewall drafts. Firewall drafts are complete firewall configurations with policy section and rules which can be immediately published or saved for publishing at a later time. Auto drafts (enabled by default) means any config change results in a system generated draft. A maximum of 100 auto drafts can be saved. These auto drafts are useful for reverting to a previously known good config. Manual firewall drafts (of which there can be 10) can be useful for having (for example) different security level policies in predefined policy for easy implementation. It is worth noting that when updates are made to the active policy (for example a new application is added), that change is not updated on previously saved drafts.

5.6.4 Exclusion List

The Distributed Firewall provides an exclusion list which allows for it to be removed from certain entities. From a security practitioner's perspective, this is a useful tool to be used very rarely, if at all. (For example, in troubleshooting, it may be useful to place a VM in the exclusion list to rule out the security policy being an issue in communication – if a problem exists with the VM in the exclusion list, the policy is clearly not the problem.) Placing a logical port, logical switch, or Group in the exclusion list means that DFW will not be applied to that/those entities at all. Even if a VM is referred to in the rules or the Applied To field, it will not receive any policy if it is in the exclusion list. Upon installation, NSX places the NSX Manager and NSX Edge node VMs into this list. This prevents novice users from locking themselves out of those entities. For a secure installation, it is recommended that a policy allowing the communication ports defined at ports.vmware.com be added and those entities be removed from the exclusion list. Figure 4-13 shows how to access the exclusion list for DFW:

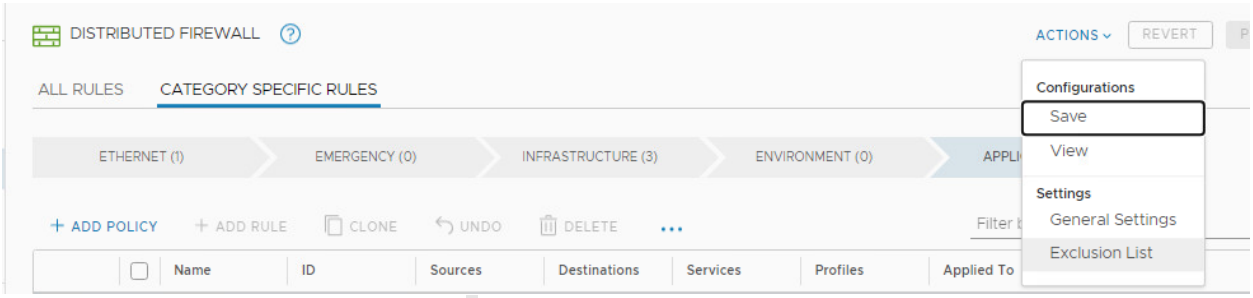

Figure 5 - 1 NSX-T Distributed Firewall Exclusion List

The exclusion list is handy for troubleshooting to remove the DFW so that it can be determined if DFW policy can be causing connectivity issues. Other than as a troubleshooting tool, its use is not recommended in secure environments.

5.6.5 Statistics

NSX Rules provides statistics for the rules, as depicted below. While traffic is flowing, the byte, packet and hit count will increase.

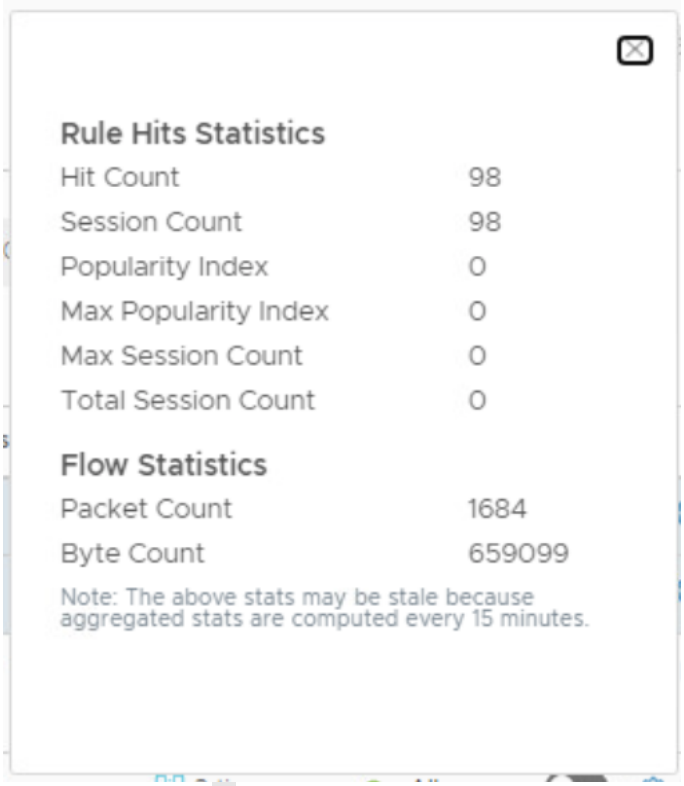

Figure 5 - 22 Distributed Firewall Rule Statistics

5.6.6 Logs

Logging is another tool which is handy for troubleshooting. NSX-T DFW logs are found in the /var/log/dfwpktlogs.log for both KVM and ESXi hosts. The log format is space delimited and contains the following information:

- timestamp:
- last eight digits of the VIF ID of the interface
- INET type (v4 or v6)
- reason (match)
- action (PASS, DROP, REJECT)
- rule set name/ rule ID
- packet direction (IN/OUT)

- packet size
- protocol (TCP, UDP, or PROTO #)
- SVM direction for next rule hit
- source IP address/source port>destination IP address/destination port
- TCP flags (SEW)

5.7 *Security Profiles*

One of the very useful tools within NSX for defining security policies is Profiles. Security Profiles are used to tune Firewall Operations features such as Session Timers, Flood Protection, and DNS Security. Each of those will be examined in this section.

5.7.1 Session Timers

Session Timers define how long the session is kept after inactivity on the session. When this timer expires, the session closes. The distributed firewall and gateway firewalls have separate independent firewall session timers by default. This is configurable per Tier-0/Tier-1 gateways or to group of VM's for DFW using Groups. In other words, default session values can be defined depending on your network or server needs. While setting the value too low can cause frequent timeouts, setting it too high will consume resources needlessly. Ideally, these timers are set in coordination with the timers

on the servers to which traffic is destined. The figures below provide the default values for the Session Timers:

Timer Property	Default (secs)	Minimum (secs)	Maximum (secs)
ICMP Error Reply	10	10	4320000
ICMP First Packet	20	10	4320000
TCP Closed	20	10	4320000
TCP Closing	120	10	4320000
TCP Established	43200	120	4320000
TCP Fin-wait	45	10	4320000
TCP First Packet	120	10	4320000
TCP Opening	30	10	4320000
UDP First Packet	60	10	4320000
UDP Multiple	60	10	4320000
UDP Single	30	10	4320000

Figure 5 - 23 Default DFW Session Timers

Timer Property	Edge Default (secs)	Minimum (secs)	Maximum (secs)
ICMP Error Reply	6	10	4320000
ICMP First Packet	6	10	4320000
TCP Closed	2	10	4320000
TCP Closing	900	10	4320000
TCP Established	7200	120	4320000
TCP Fin-wait	4	10	4320000
TCP First Packet	120	10	4320000
TCP Opening	30	10	4320000
UDP First Packet	30	10	4320000
UDP Multiple	30	10	4320000
UDP Single	30	10	4320000

Figure 5 - 24 Default Gateway Session Timers

5.7.2 Flood Protection

Flood Protection helps protect against Distributed Denial of Service (DDoS) attacks. DDoS attacks aim to make a server unavailable to legitimate traffic by consuming all the available server resources through flooding the server with requests. Creating a flood protection profile imposes active session limits for ICMP, UDP, and half-open TCP flows. The distributed firewalling can cache flow entries which are in SYN-SENT and SYN-RECEIVED state and promote each entry to a TCP state after and ACK is received from the initiator, completing the three-way handshake. Note that due to its

distributed nature, the DFW is far better able to protect against DDoS attacks than a legacy centralized firewall which may need to protect many servers at once.

The following table provides details around the Flood Protection parameters, their limits, and their suggested use:

Parameter	Minimum and maximum values	Default	
TCP Half Open Connection Limit - TCP SYN flood attacks are prevented by limiting the number of active, not-fully-established TCP flows which are allowed by the firewall.	1-1,000,000	Firewall - None Edge Gateway - 1,000,000	Set this text box to limit the number of active TCP half open connections. If this text box is empty, this limit is disabled on ESX nodes and set to the default on value of Edge Gateways.
UDP Active Flow Limit -UDP flood attacks are prevented by limiting the number of active UDP flows which are allowed by the firewall. Once the set UDP flow limit is reached, subsequent UDP packets which can establish a new flow are dropped.	1-1,000,000	Firewall - None Edge Gateway - 1,000,000	Set this text box to limit the number of active UDP connections. If this text box is empty, this limit is disabled on ESX nodes and set to the default on value of Edge Gateways.
ICMP Active Flow Limit - ICMP flood attacks are prevented by limiting the number of active ICMP flows which are allowed by the firewall. After the set flow limit is reached, subsequent ICMP packets which can establish a new flow are dropped.	1-1,000,000	Firewall - None Edge Gateway - 10,000	Set this text box to limit the number of active ICMP open connections. If this text box is empty, this limit is disabled on ESX nodes and set to the default on value of Edge Gateways.
Other Active Connection Limit	1-1,000,000	Firewall - None Edge Gateway - 10,000	Set this text box to limit the number of active connections other than ICMP, TCP, and UDP half open connections. If this text box is empty, this limit is disabled on ESX nodes, and set to the default on value of Edge Gateways.
SYN Cache - Syn Cache is used when a TCP half open connection limit has also been configured. The number of active half-open connections are enforced by maintaining a syncache of the not-fully-established TCP sessions. This cache maintains the flow entries which are in SYN_SENT and SYN_RECEIVED states. Each syncache entry will be promoted to a full TCP state entry after an ACK is received from the initiator, completing the three-way handshake.		Only available for firewall profiles.	Toggle on and off. Enabling SYN cache is effective only when a TCP half open connection limit is configured.
RST Spoofing		Only available for firewall profiles.	Toggle on and off. SYN Cache must be selected for this option to be available

Figure 5 - 25 Flood Protection Parameters

5.7.3 DNS Security

DNS Security guards against DNS-related attacks. DNS security controls include the ability to snoop on DNS responses for a VM or group of VMs to associated FQDNs with IP addresses and adding global and default DNS server information for select VMs. Only one DNS server profile can be applied to any given VM. Tags are supported so that profiles can be applied associated with a given group.

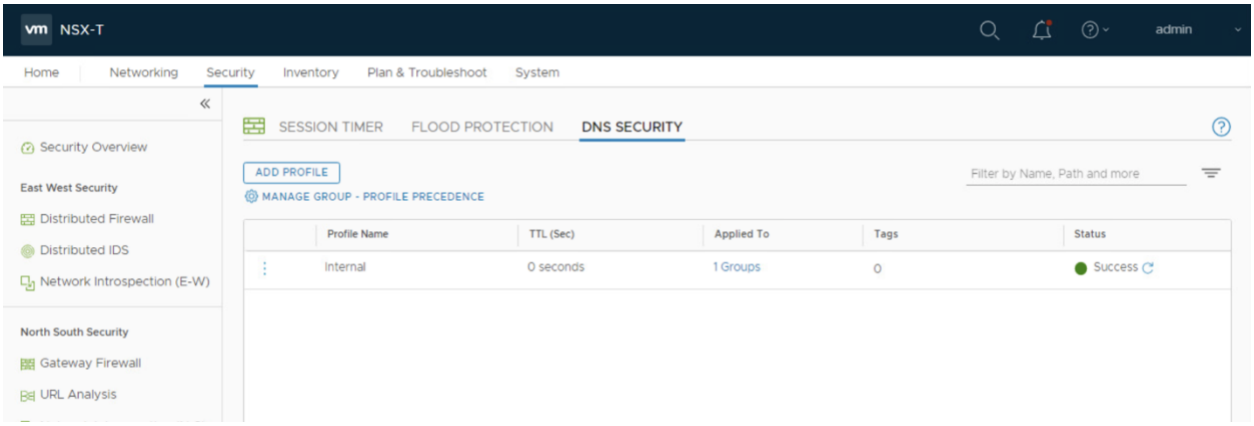

Figure 5 - 26 DNS Security UI

5.8 *Policy Automation with vRNI*

The policy journey is one which requires constant revisiting and reviewing of policy as the infrastructure changes, as the compliance requires change, and as the business needs change. The following figure depicts the basics of the security journey:

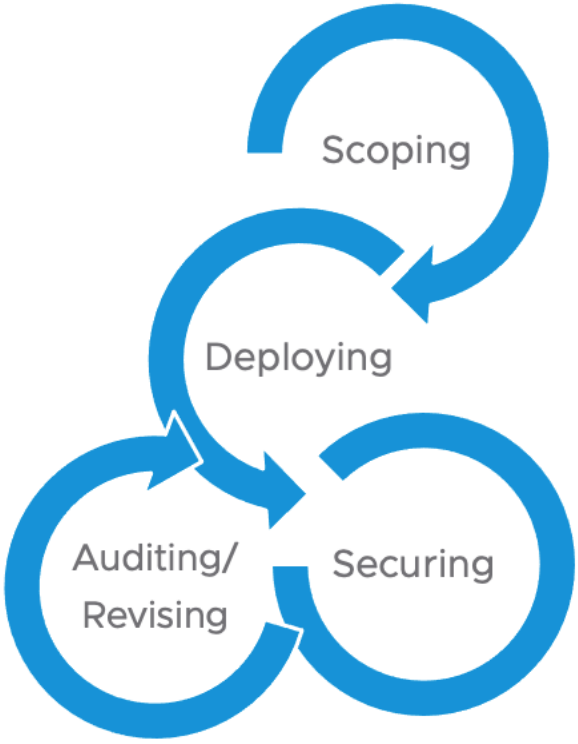

Figure 5 - 27 Security Policy Journey

The first step of the policy journey is defining the scope. Although Scope is specifically used in the context of PCI, it is a concept which is applicable to every environment. Scope defines the breadth of the security zone. The scoping exercise in a typical enterprise environment will be to define the production and non-production areas, at a minimum. The production area would include any assets that are mission critical. This is the area of greatest security and least risk. The non-production assets are those asserts where some risk is tolerable. This would be where new code gets deployed before the production area. Communication across the prod-non prod boundary is tightly controlled.

After the scope has been defined, the next step of the journey is deployment. In the case of NSX, this is something that does not require a change of IP address scheme, nor a rearchitect of the network. This means that NSX firewalling may be deployed alongside or even in concert with existing legacy firewalls. Unlike replacing a Checkpoint Firewall with Palo Alto firewall where there is a switch that must be made, NSX is deployed as part of the data center hypervisor fabric and can run alongside the legacy firewalls, either offloading traffic or working in conjunction with legacy firewalls using service insertion as described in chapter <>

5.8.1 Discovery with vRNI

In order to understand the east west traffic patterns of the scoped area, VMware provides vRealize Network Insight as a tool. This tool can discover traffic patterns before NSX is installed. Most

importantly, it can discovery underlying health problems in applications which may be exacerbated by a change of infrastructure. Ideally, only healthy applications are secured. However, the world is not always running at our behest so if there a need to secure an unhealthy application, vRNI offers the means to review the sequence of events for later troubleshooting.

Figure 5 - 28 vRNI Application Health Summary

If there is an existing CMDB such as ServiceNow, vRNI can leverage that information for expedient application definitions. vRNI will discover the flows of an application and capture the source and destination IP addresses, ports, and protocols. vRNI also has the intelligence to recognize groupings. Clicking on a tier of an application in the vRNI Plan Security wheel, will provide details including the number of flows (which helps understand the popularity of the tier) and also the numbe4r of services in a tier (a measure of the complexity of the tier). vRNI will also provide a suggested policy recommendation for the given tier for the application.

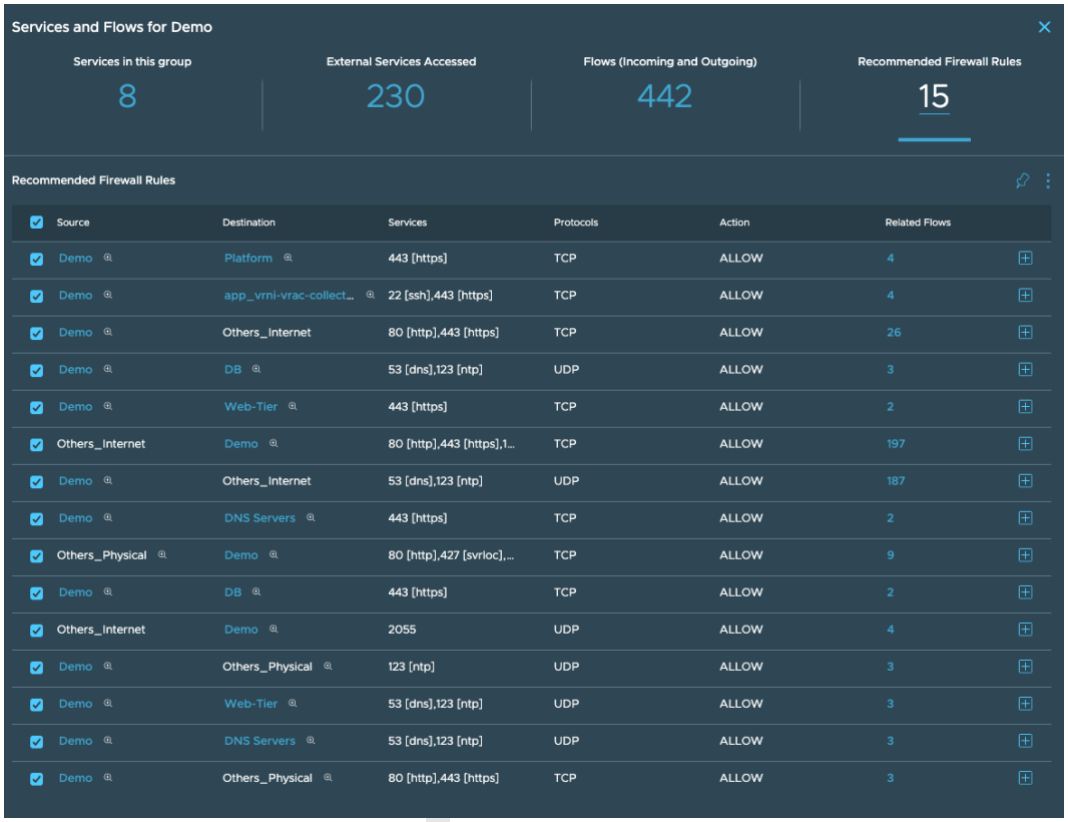

Figure 5 - 29 vRNI Policy Recommendation

Once vRNI has discovered the application flows within an application, a security policy can be exported from vRNI:

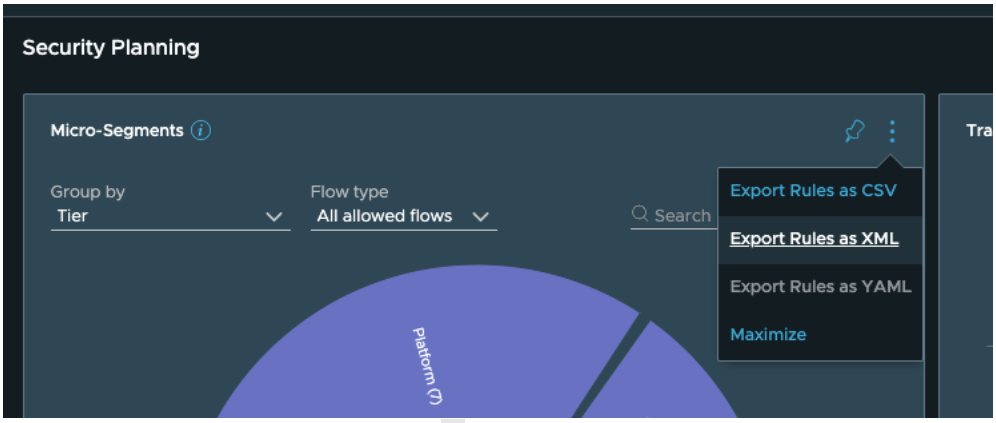

Figure 5 - 30 vRNI export options

The NSX API provides a means for importing said rules into NSX. The policy nature of the NSX REST API makes creating rule quite simple as many objects can be created in one call. More details of the NSX 3.1.1. API can be found here: https://code.vmware.com/apis/1124/nsx-t.

At this point, the policy can be examined by the security team by reviewing the CSV export. This review can happen prior to the actual NSX deployment so that the day that NSX is installed and enabled on the hosts, the approved policy can be imported into the NSX environment, providing immediate protection.

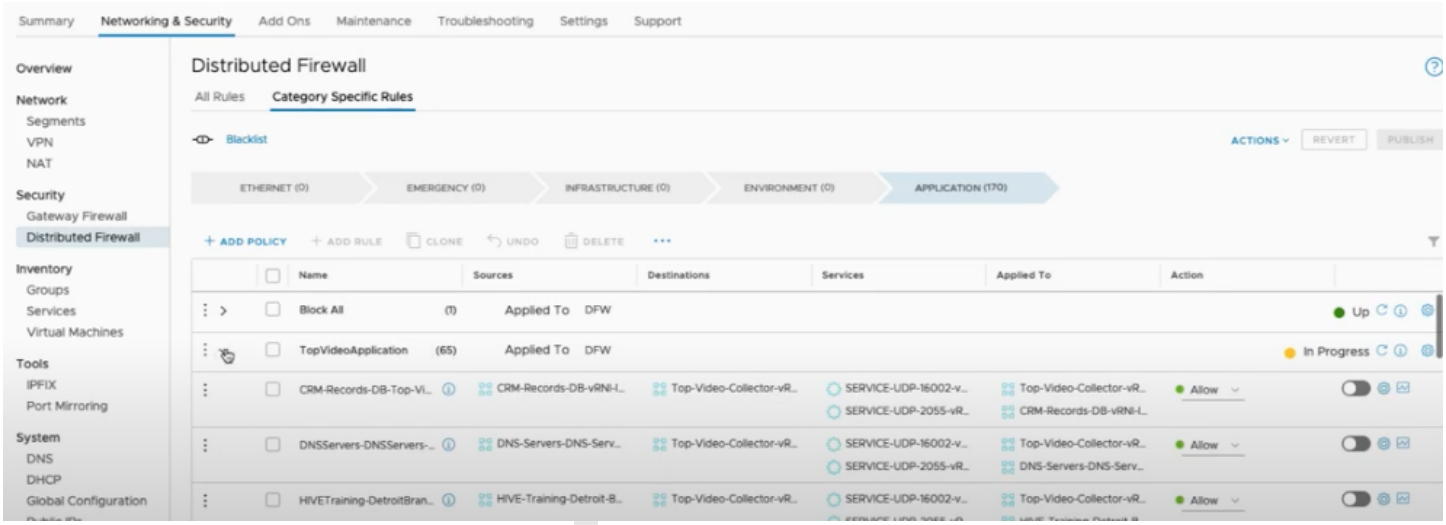

Figure 5 - 31 vRNI Imported Policy in NSX Manager

It should be noted that the policy imported in the figure above was done with the rules are disabled. This is an example of a policy import that can be done during production hours with the enabling of the rules to be done during a defined maintenance window. If this is not necessary, the rules could have been imported enabled by default for immediate protection.

6 Container Security

The programmable nature of NSX makes it the ideal networking and security infrastructure for containers. With NSX, the developer can deploy apps with the security built in from the get-go. While security is traditionally seen as an impediment among the developers, the visibility which security requires can be leveraged by developers to ease their troubleshooting. Moreover, NCP security can be quite extensive providing firewalling, LB (including WAF), and IDS. This section dives deeply into the NSX Container Plug-in, a software component provided by VMware in the form of a container image meant to be run as a Kubernetes pod.

NSX Container Plug-in (NCP) provides integration between NSX-T Data Center and container orchestrators such as Kubernetes, as well as integration between NSX-T Data Center and container-based PaaS (platform as a service) products such as OpenShift and Pivotal Cloud Foundry or CaaS (Container as a Service) platforms such as EKS (Amazon Elastic Kubernetes Service), AKS (Azure Kubernetes Service), and GKE (Google Kubernetes Engine). The NCP has a modular design, allowing for additional platform support in the future.

The main component of NCP runs in a container and communicates with NSX Manager and with the Kubernetes control plane. The NCP monitors changes to containers and other resources and manages networking resources such as logical ports, switches, routers, and security groups for the containers by calling the NSX API.

The NSX Container Plug-in is a software component provided by VMware in form of a container image, e.g. to be run as a K8s/OCP Pod.

There are four key design goals of the NSX OCP/K8S integration:

1. Don't stand in the way of the developer!
2. Provide solutions to map the Kubernetes constructs to enterprise networking constructs
3. Secure Containers, VMs and any other endpoints with overarching Firewall Policies and IDS
4. Provide visibility & troubleshooting tools to ease the container adoption in the enterprise

The NSX CNI plug-in runs on each Kubernetes node. It monitors container life cycle events, connects a container interface to the guest vSwitch, and programs the guest vSwitch to tag and forward container traffic between the container interfaces and the VNIC.

The NCP automatically creates an NSX-T Data Center logical topology for a Kubernetes cluster, and creates a separate logical network for each Kubernetes namespace. It also connects Kubernetes pods to the logical network, allocates IP and MAC addresses. Finally, the NCP supports network address translation (NAT) and allocates a separate SNAT IP for each Kubernetes namespace. These separate SNAT IP addresses allow each Kubernetes names space to be uniquely addressable.

The NCP implements the following in Kubernetes:

- Security policies with the NSX-T Data Center distributed firewall.
    - Support for ingress and egress network policies.
    - Support for **IPBlock** selector in network policies.

- Support for **matchLabels** and **matchExpression** when specifying label selectors for network policies.
- Support for selecting pods in another namespace.

- **ClusterIP** and **LoadBalancer** service types.
- Ingress with NSX-T layer 7 load balancer.
  - Support for HTTP Ingress and HTTPS Ingress with TLS edge termination.
  - Support for Ingress default backend configuration.
  - Support for redirect to HTTPS, path rewrite, and path pattern matching.
- Creates tags on the NSX-T Data Center logical switch port for the namespace, pod name, and labels of a pod, and allows the administrator to define NSX-T security groups and policies based on the tags.

NCP 3.0.1 supports a single Kubernetes cluster. You can have multiple Kubernetes clusters, each with its distinct NCP instance, using the same NSX-T Data Center deployment.

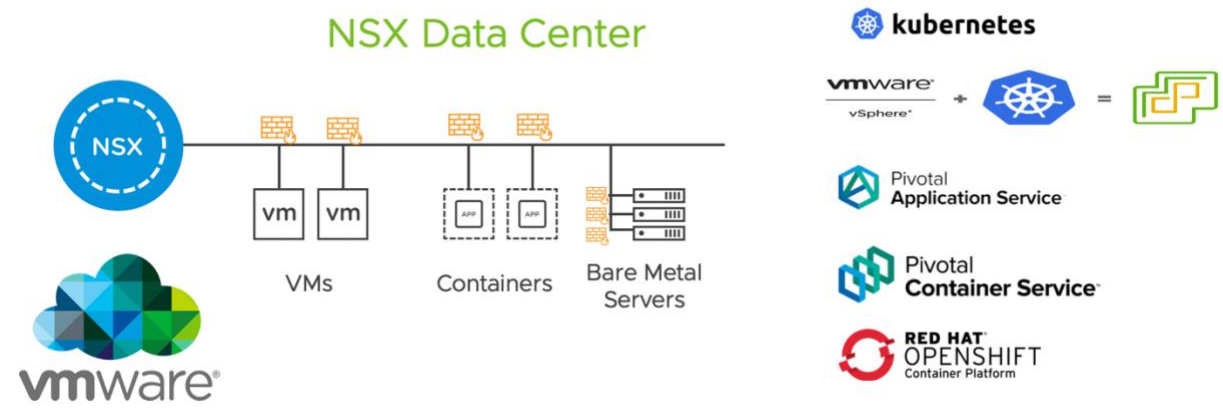

Figure 6 - 1 NSX-T Broad platform support

6.1 *NCP Components*

NCP is built in a modular way, so that individual adapters can be added for different CaaS and PaaS systems. The current NCP supports K8S, Tanzu, and OpenShift, but more can be easily added. Figure 6.2 shows this modular architecture.

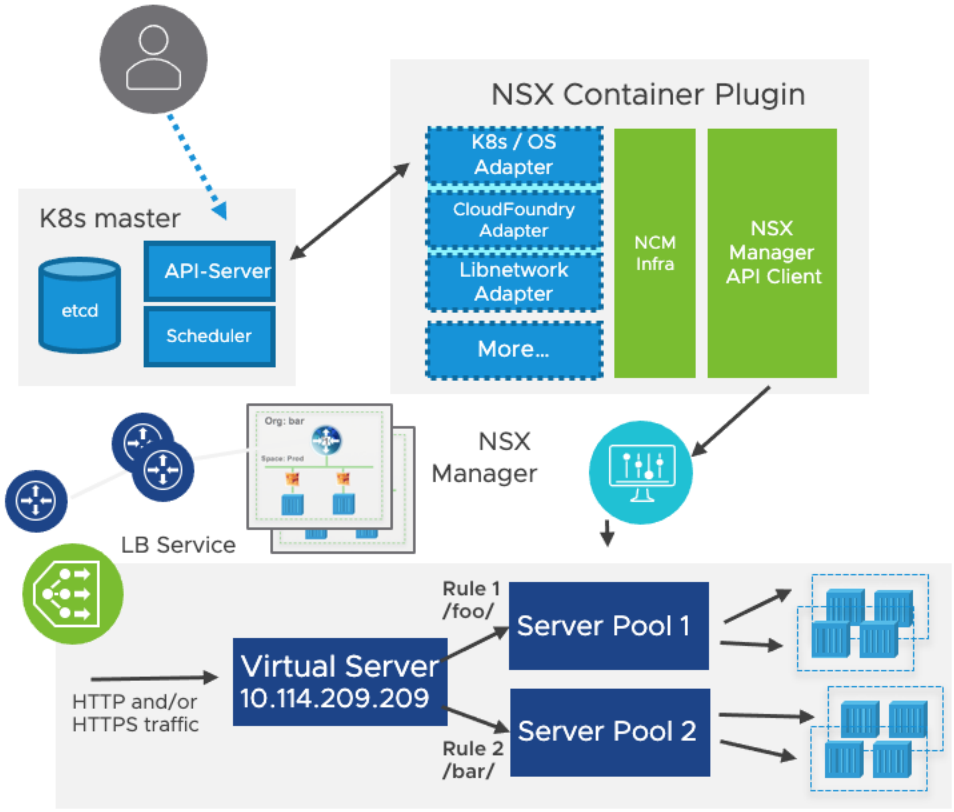

Figure 6 - 2 NCP Components

The heart of the NCP is the NCM Infra. It talks to both the NSX Manager via the NSX Manager API client and the container environments via a container-specific adapter as shown above. In a K8s environment, the NCP communicates with the K8s control plane and monitors changes to containers and other resources. It monitors containers life cycle events and connects the container interface to the vSwitch. In doing so, the NCP will program the vSwitch to tag and forward container traffic between the container interfaces and the vnic. The NCP also manages resources such as logical ports, switches, and security groups by calling the NSX API. This allows the NCP to extend all NSX services, even Distributed IDS (discussed in chapter 7) to the container, as seen in figure 5-3, below:

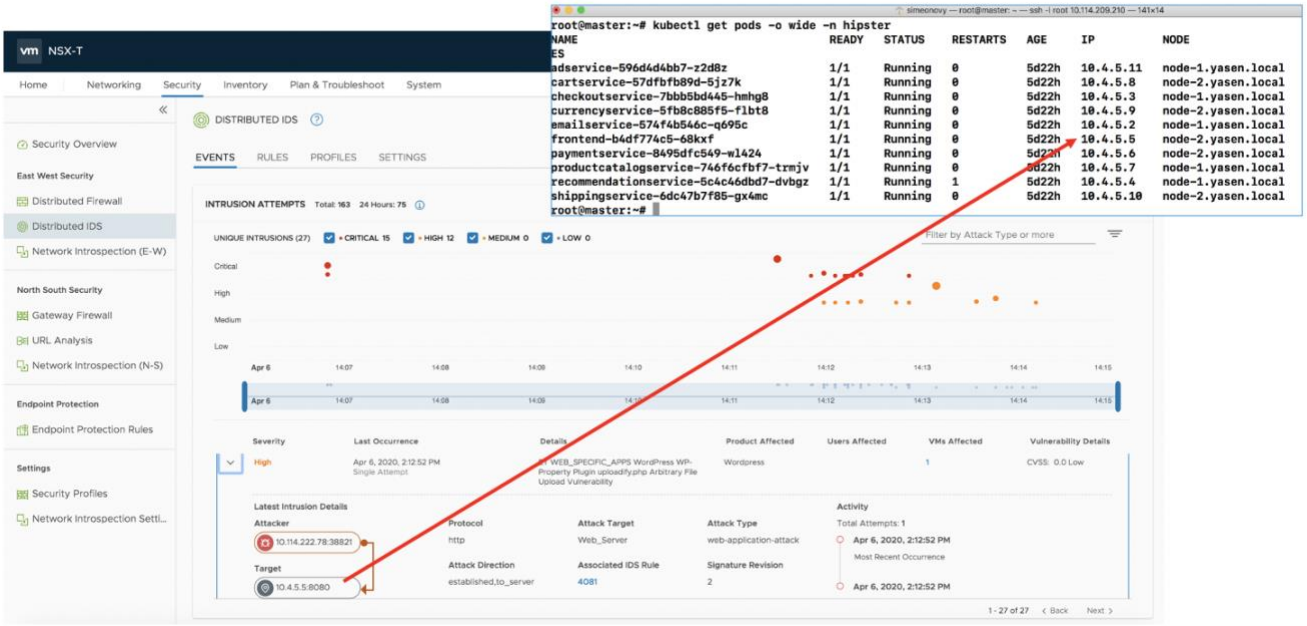

Figure 6 - 3 Distributed IDS for Containers

Because NSX infrastructure exists solely in software, it is entirely programmable. The next section will look at how the NCP calls to the NSX Manager when instantiating K8s clusters.

As described above, the NCP provides per namespace topology upon creation. This is shown in figure 5.4 below in which two namespaces are created: foo and bar, each with its own topology.

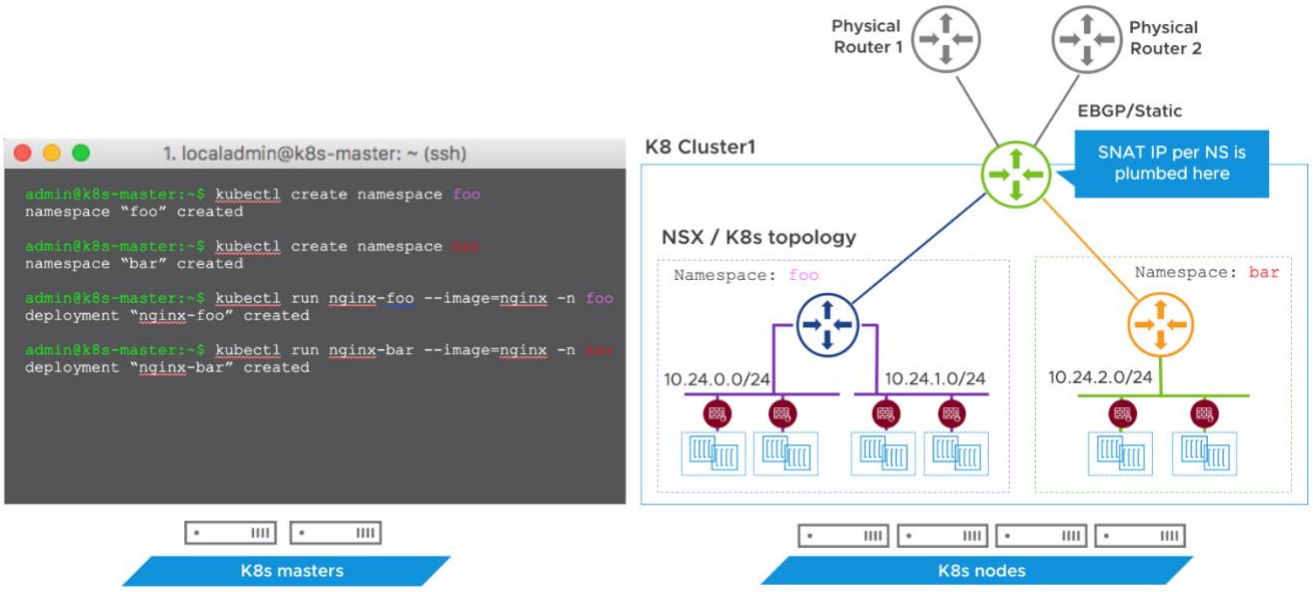

Figure 6 - 4 NSX-T Per Namespace Topology

Walking through the four commands above provides an understanding of this environment's instantiation. The first thing the NCP does is request a subnet for each namespace from the block which is pre-configured in NSX. (This block is defined when the NCP is set up in NSX.) Next, the NCP will create a logical switch and T1 router (which it will attach to the pre-configured T0 router). Finally,

the NCP will create a router port on the T1 which it will attach to the logical switch (to which it has assigned the subnet it received). This is how the commands result in the topology on the right. Note that smaller environments, may wish to have a shared T1 for all namespaces. This is also supported. On the other end of the spectrum, where there may be a requirement for massive throughput, Equal Cost Multi Path (ECMP) routing may be enabled on the T0s above the T1s, providing up to 8 parallel paths in an out of each environment. (for more details on NSX network design, please see the NSX Design document.)

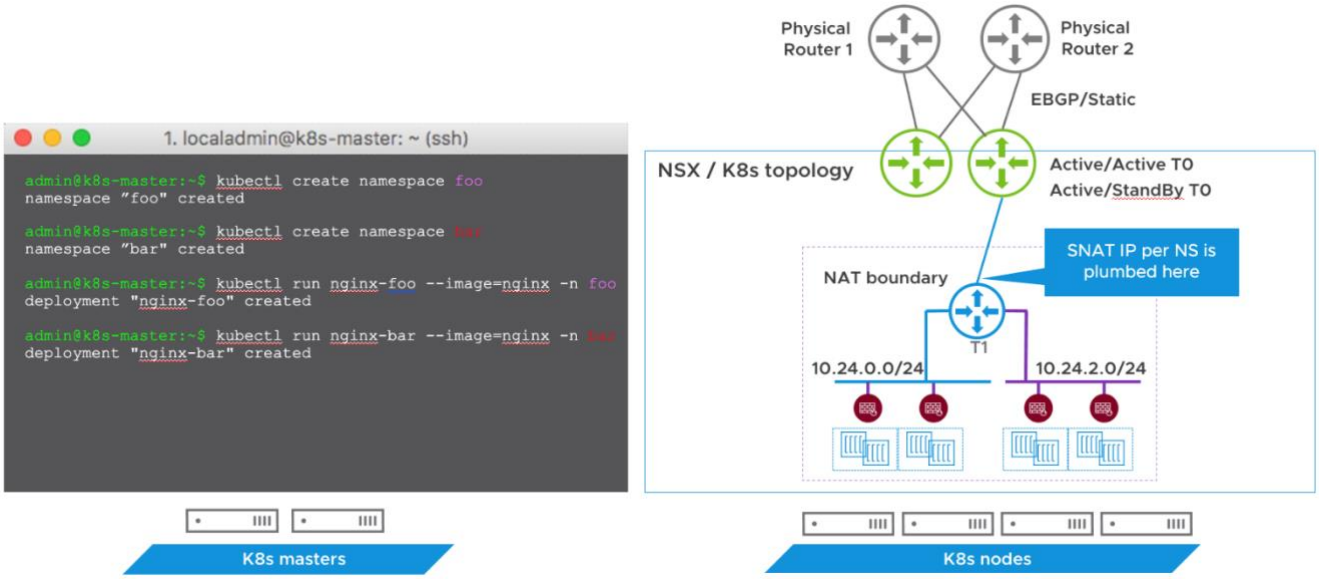

Figure 6 - 5 NSX-T Namespace Scalability

One of the critical pieces of a secure infrastructure design is the reliability of IP addressing. This is necessary for forensic purposes. It is critical that when there is an endpoint with a given IP address, it be assigned to that endpoint throughout the life of that endpoint and that it will not change, making it harder to track that endpoint's history. This leads to the requirement for persistent SNAT in the world of containers. NSX allows for persistent SNAT IP per K8S service. With this feature, a set of Kubernetes Workloads (Pods) can be assigned to use a specific IP or group of SNAT IPs from which to source their traffic. Persistent SNAT also allows the creation of rules in legacy firewalls and other IP-addressed based infrastructure.

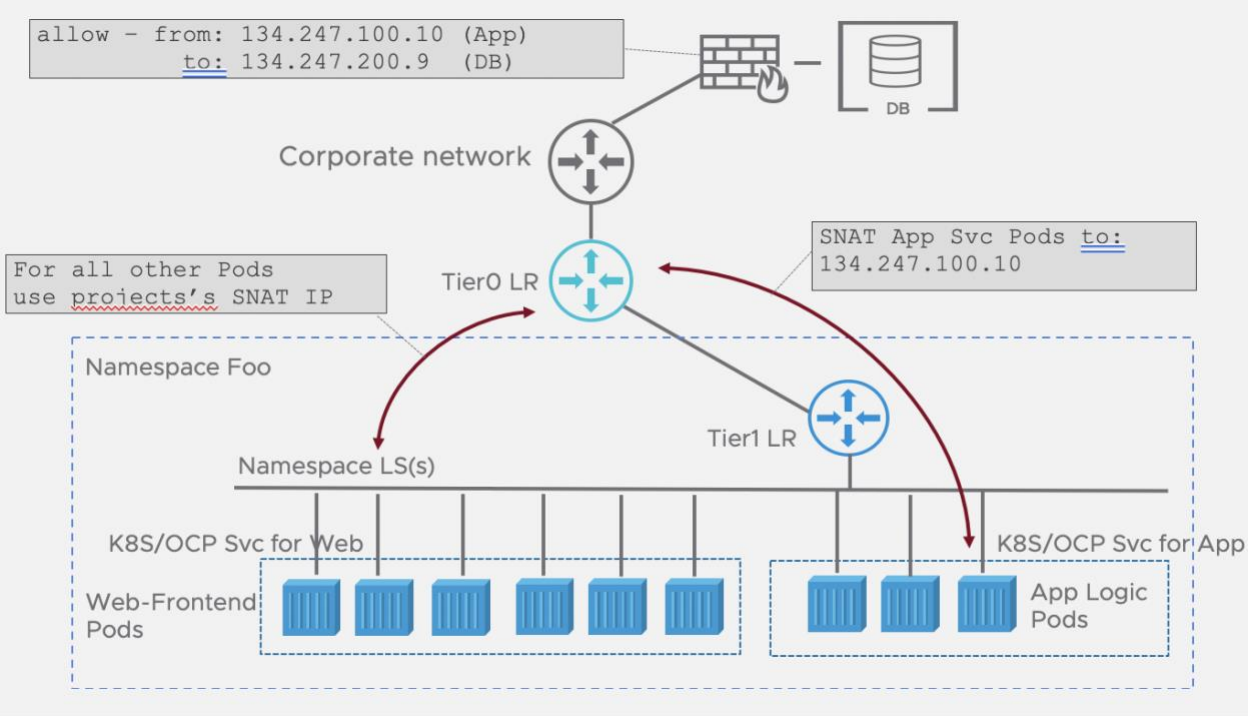

Figure 6 - 6 NSX-T Persistent SNAT IP per K8S Service

To further help with security, metadata within Kubernetes (like namespace, pod names, and labels) all get copied to the NSX Logical Port as Port Tags, as shown in Figure 6 - 7 K8S Metadata mapping below.

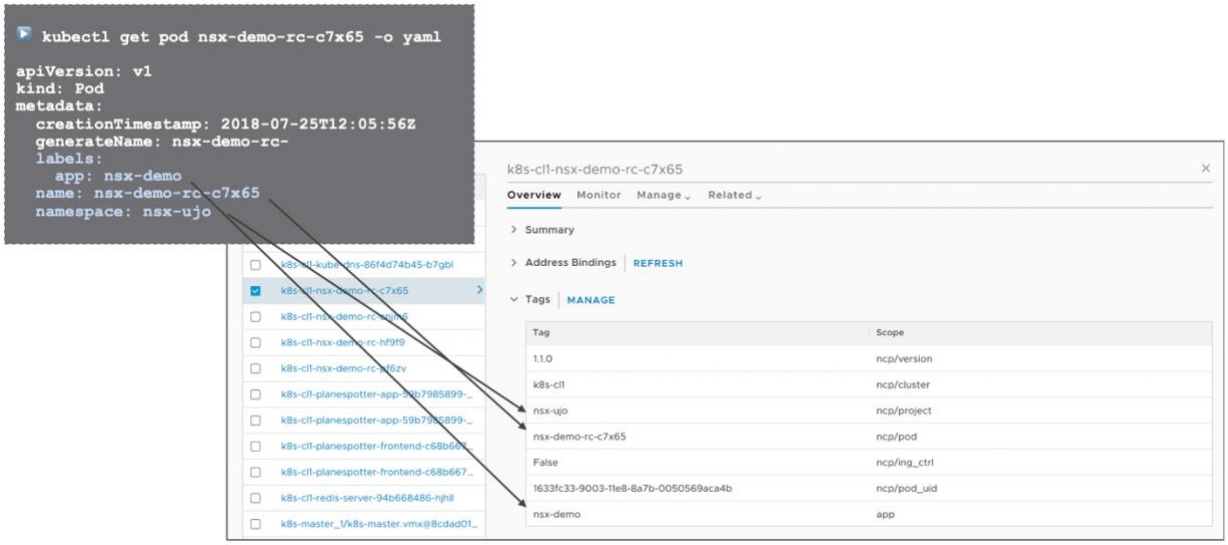

Figure 6 - 7 K8S Metadata mapping

Although this may seem like merely an administrative convenience, it has significant security implications as well. NSX can be configured to collect ports and switches in dynamic security groups based on Tags (derived from Kubernetes Metadata). Those same groups can be referenced in firewall rules, as Figure 6 - 9 NSX-T DFW Category Support shows.

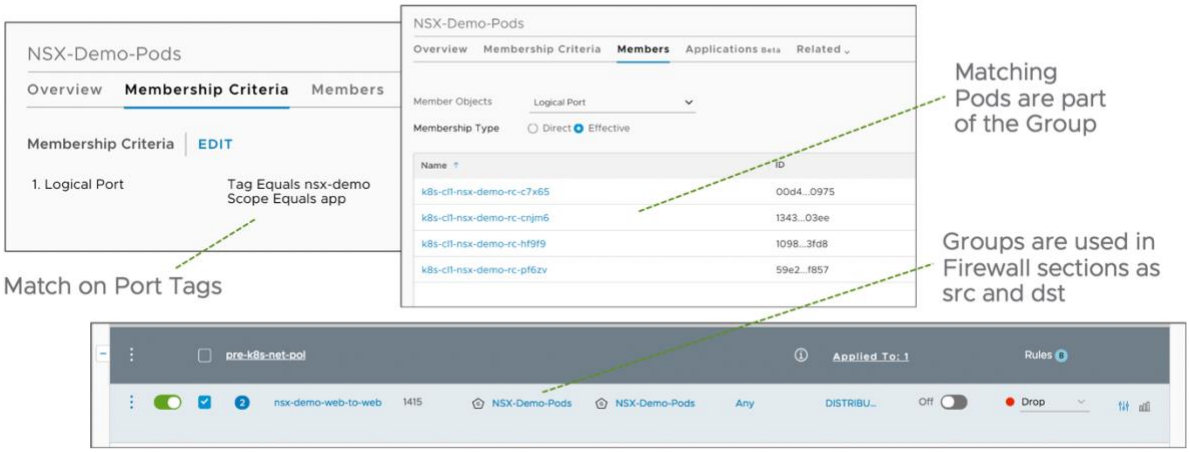

Figure 6 - 8 K8S Pre-Created Firewall Rules with Pre-Created Groups

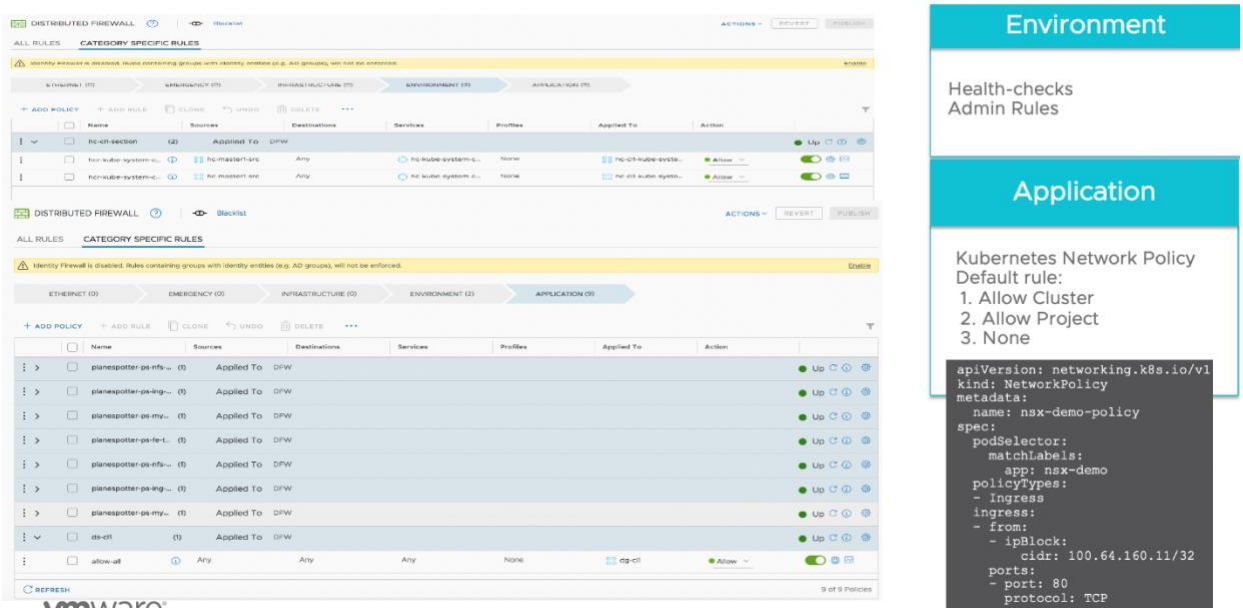

Figure 6 - 9 NSX-T DFW Category Support

6.2 *Tanzu Application Service*

NCP functionality in Tanzu environments is similar to the one described in the K8s section above. The NMC Infra component lies between the Cloud Foundry Adapter and the NSX API Client to orchestrate the two environments.

In Tanzu application service environments, CF orgs (typically a company, department, or applications suite) are assigned a separate network topology in NSX so that each CF org gets its own Tier 1 router (as seen in the K8S section above). For each CF space, NSX creates one or more logical switches, which are then attached to the Org's T1 router. Each Tanzu AI (container) has its own logical port on an NSX logical switch (so NAT is not needed). Every cell can have AIs from different orgs and spaces. Every AI has DFW rules applied on its Interface, with policies defined in the new cf-networking policy server. ASGs (Application Security Groups) are also mapped to the DFW. For North/South routing, NSX infrastructure (Tos) provide connectivity to the outside world. During installation, one can

select direct Gorouter to container networking (with or without NAT). NSX also provides IP Address Management (IPAM) by supplying subnets (from the IP Block provided at install) to Namespaces. NSX also provides the individual IP addresses and MACs to the AIs (containers).

6.3 *OpenShift*

The NSX Container Plugin for OpenShift is designed for OpenShift4 (and for OpenShift3 in the case of NCP 2.5). As described above, the main component of the NCP runs in a container, communicating with the NSX Manger via the Client API. It also communicates with the OpenShift control plane via the OpenShift Adapter. Through this interaction, the NCP will create an NSX-T logical topology for each OpenShift cluster, creating a separate logical network for each OpenShift namespace. The NCP will connect the OpenShift pods to the logical network, allocating IP and MAC addresses. As the NCP creates the logical switch port, it will assign labels for the namespace, pod name, and labels of a pod which will can be referenced in firewall policies. Each OpenShift namespace will also be allocated an SNAT. Through the DFW, the NCP will also support ingress and egress network policies with *IPBlock* selector, as well as *matchLabels* and *matchExpressions* when specifying label selectors for policies. Using the NSX LB, the NCP can implement the OpenShift route, including support for HTTP route and HTTPS route with TLS edge termination, as well as routes with alternate backends and wildcard subdomains. The Advanced LB available with NSX allows for a whole security suite to be applied to the HTTP traffic, including rate limiting and WAF. For more details on the ALB security suite, see Chapter XX.

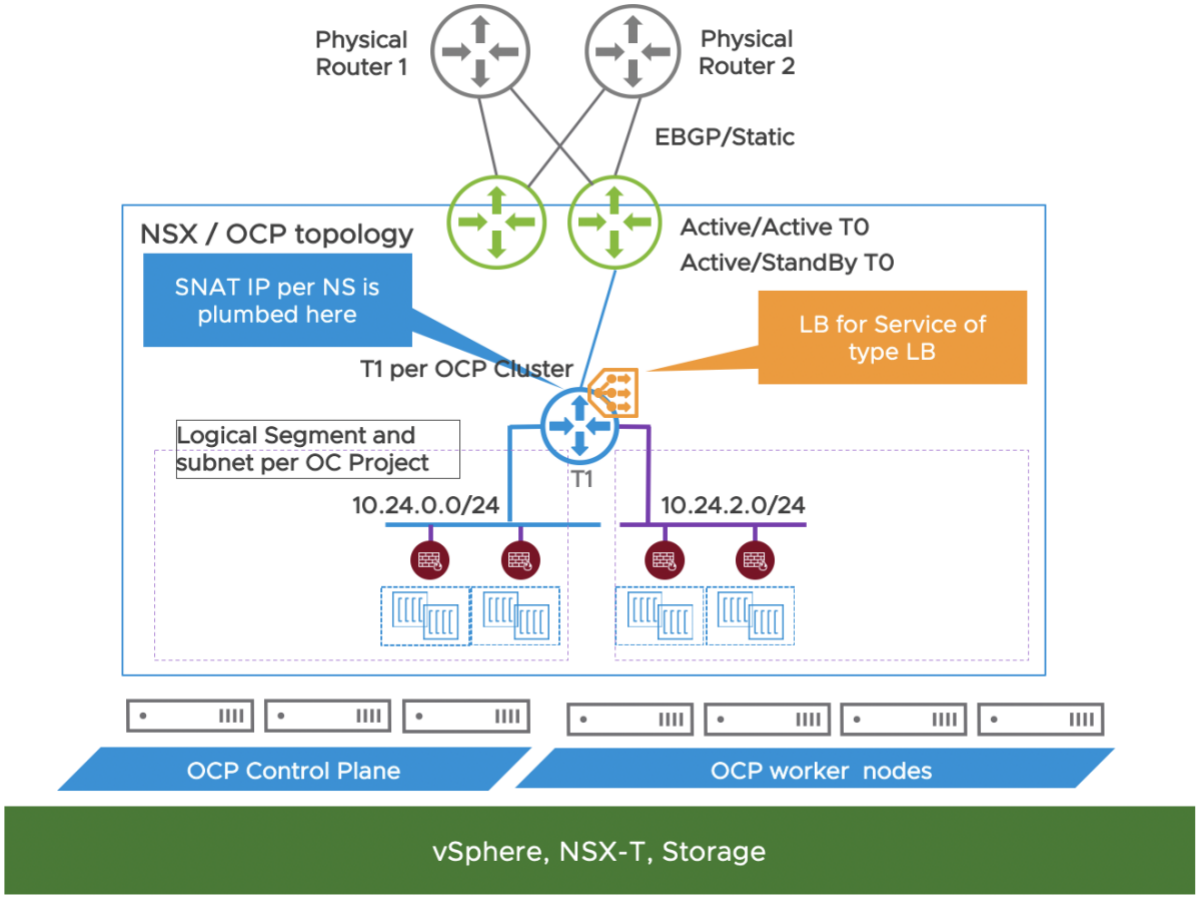

Figure 6 - 10 OpenShift NCP

To trigger an NCP deployment, the networkType field in the CRD in the RedHat UBI (Universal Base Image) must be “ncp”. Both the NCP and the Network Cluster Operator are packaged with the Red Hat UBI. Operators apply the equivalent of the K8s controller model at the level of the application.

6.4 *NCP Features*

The previous sections discussed the NCP architecture and functionality in support of K8S, OpenShift, and Tanzu Application Services. This section will look at the additional functionality the NCP brings to these environments that makes them more secure and easier to operate.

6.4.1 Visibility

NSX ends the black hole that is the container environments. NSX Topology mapper provides a dynamic topology map of the environment.

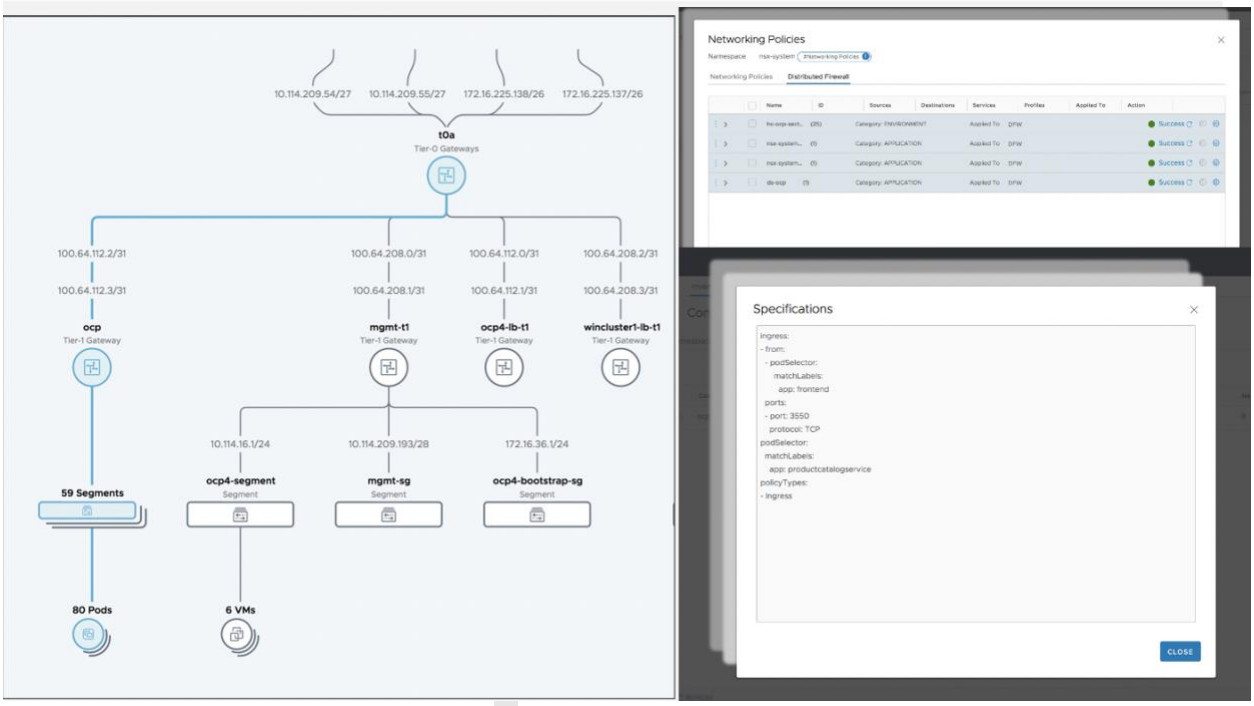

Figure 6 - 11 NCP Topology and Policy Visibility

Tools such as traceflow not only extend visibility, but they also aid in troubleshooting connectivity across the entire flow, from VM to container, or even between pods.

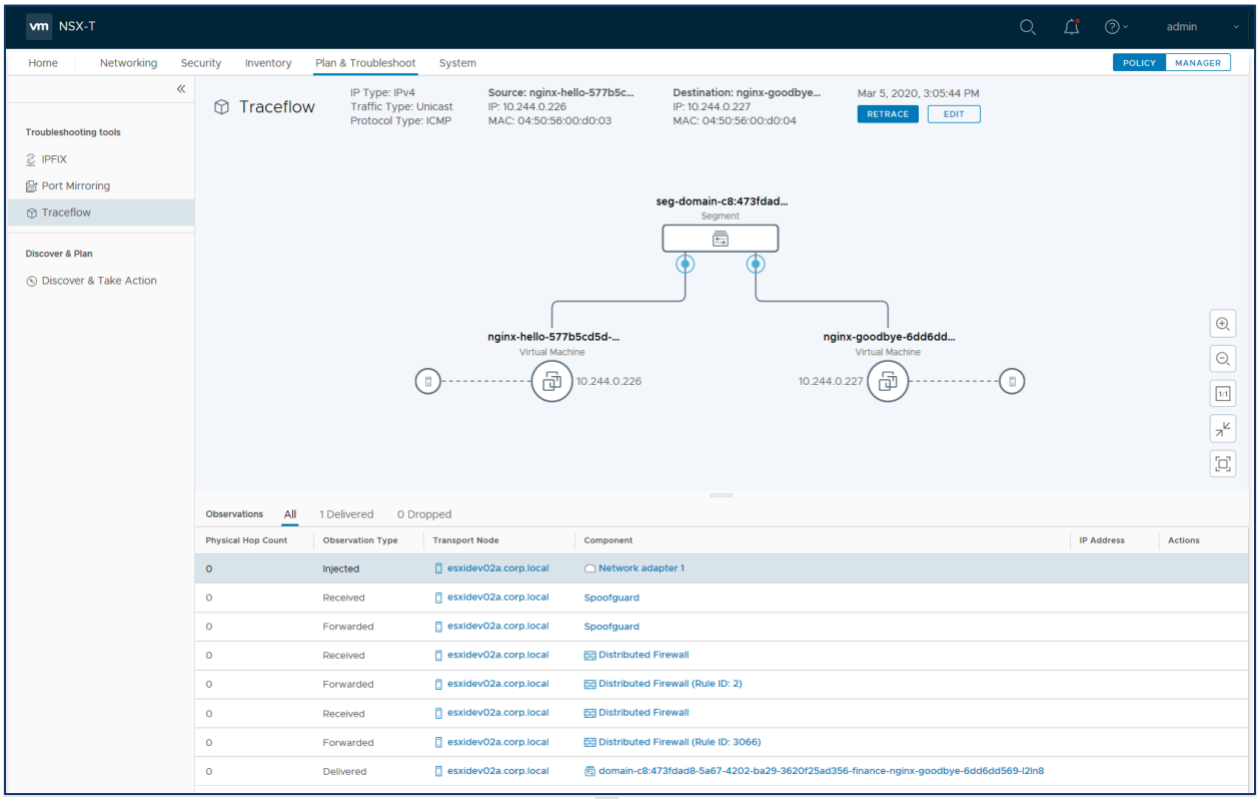

Figure 6 - 12 NCP Traceflow

6.4.2 IPv6

IPv6 is supported on the NSX Container Plug-in, for both single and two-tier topologies. IPv6 and IPv4 IP blocks cannot be mixed in the NCP configuration. Dual stacks are not supported, so if a container has an IPv6 address, it cannot have IPv4 addressing. For north-south traffic to work properly, the Tier-0 gateway must have an IPv6 address and spoofguard must be disabled. The Kubernetes cluster must be created with an IPv6 service cluster CIDR with a maximum 16 bit subnet mask. All namespaces will be in no_SNAT mode. Kubernetes nodes must have an IPv6 address for connectivity between the nodes and pods, and TCP and HTTP liveliness and readiness probes to work. Either SLAAC or static IPs can be used. The Kubernetes nodes can also be in dual-stack mode, in which case you must register the node with an IPv6 address by specifying the IPv6 address with the *-node-ip* option as one of the kubelet's startup parameters.

6.5 *Project Antrea*

No discussion of Container Networking would be complete without the mention of Project Antrea. Project Antrea is an open source Container Network Interface (CNI) plug in providing pod connectivity and network policy enforcement with Open vSwitch in K8s. It is available on https://antrea.io. Being an open source project, Antrea is extensible and scalable. Antrea simplifies networking across different clouds and operating systems. Its installation is quite simple, requiring only one yaml file. An Antrea CNI is installed per K8s cluster, allowing for better scale in environments with many K8s clusters. In the future, these CNIs will be able to managed by the NSX manager for global policy distribution. This document will be updated with details when that functionality comes available.

7 Firewall features

The NSX Firewall provides many features which are useful for securing the environment. Although there are a myriad of firewall features including time of day rules and so on this chapter will only highlight a few of the ones most commonly used: URL Analysis, Service Insertion, and Endpoint Protection (also known as Guest Introspection). The focus on these features is highlighted due to the impact these features has on system architecture and design. For an exhaustive look at firewall features, see the NSX product documentation.

7.1 *URL Analysis*

URL Analysis allows administrators to gain insight into the type of external websites accessed from within the organization and understand the reputation and risk of the accessed websites. URL Analysis is available on the gateway firewall and is enabled on a per cluster basis. After it is enabled, you can add a context profile with a URL category attribute. URL Analysis Profiles specify the categories of traffic to be analyzed. If no profiles are created, all traffic is analyzed. To analyze domain information, you must configure a Later 7 gateway firewall rule on all Tier-1 gateways backing the NSX Edge cluster for which you want to analyze traffic. The DNS traffic is analyzed to extract the hostname and IP information. The extracted information is then used to categorize and score traffic.

To download the category and reputation database, the management interface of the edge nodes on which URL Analysis is enabled must have internet access.

This is depicted in Figure 7 - 1 NSX-T URL Analysis below.

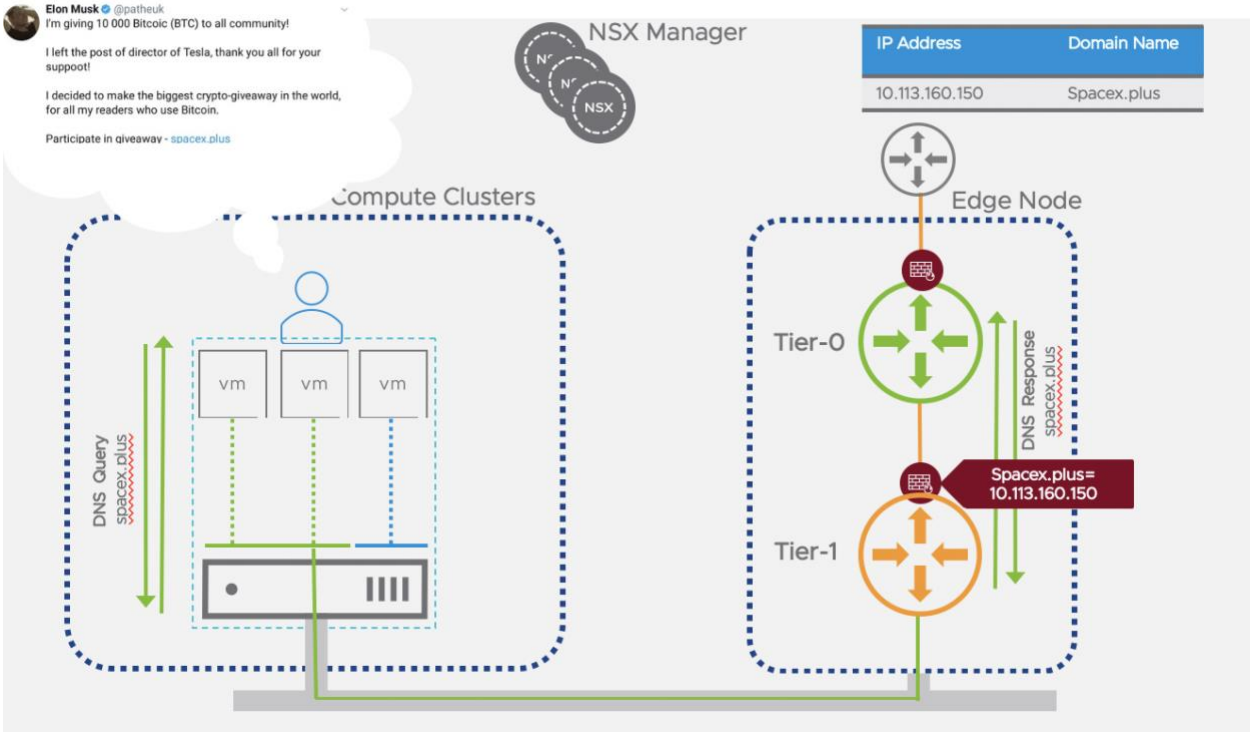

Figure 7 - 1 NSX-T URL Analysis

URL categories are used to classify websites into different types. There are more than 80 predefined categories in the system. Currently, categories cannot be customized. A website or domain can belong

to multiple categories. Based on their reputation score, URLs are classified into the following severities:

- High Risk (1-20)
- Suspicious (21-40)
- Moderate Risk (41-60)
- Low Risk (61-80)
- Trustworthy (81-100)

The Webroot BrightCloud® Web Classification and Web Reputation Services provide the most effective way to block access to unwanted content and protect users against web-based threats. For these services, Webroot:

- Uses patented machine learning that enables single classifiers to work at a rate of 20K classifications per second; with 500+ classifiers running in parallel, site classification is extremely fast and accurate
- Categorizes the largest URL database of its kind across 82 categories
- Observes and protect users in real time from the risks of connecting to any URL, regardless of reputation
- Provides details as to why a site classification was made, empowering admins to make better-informed security decisions

7.2 Service Insertion and Service Chaining

The value of NSX security extends beyond NSX to your pre-existing security infrastructure; NSX is the mortar that ties your security bricks to build a stronger wall. Legacy security strategies were intolerant of pre-existing security infrastructure. Anyone who had a Checkpoint firewall and wanted to move to a Palo Alto Networks firewall would run the 2 managers, side by side until the transition was complete. Troubleshooting during this transition period required a lot of chair swiveling. NSX brings a new model, complementing pre-existing infrastructure. Service Insertion is the feature which allows NSX firewalls (both gateway and DFW) to send traffic to legacy firewall infrastructure for processing. This can be done as granularly as a port level, without any modification to existing network architecture.

Service Insertion not only sends the traffic to other services for processing, Service Insertion offers a deep integration which allows the exchange of NSX Manager objects to SI service managers. So, a group in NSX which is comprised on VMs which a substring of "web" (for example) would get shared to the SI service manager. Thus, when a new VM is spun up which becomes a member of the new group, the NSX Manager will send that update to the SI Service Manager so that policy can be consistently applied across platforms.

This section examines Service Insertion, which provides the functionality to insert third-party services at the Tier-0 or Teir-1 gateways.

Figure 7 - 2 shows Service Insertion at the gateway firewall (north south service insertion) and at the distributed firewall (east west service insertion). Notice that east west service insertion means it can be applied to traffic destined to physical servers, VMs, or containers. In other words: if you decide that you want your sql traffic to be directed to a Fortinet firewall (a viable security policy), that policy will apply to all sql traffic destined to physical servers, VMs, or containers as the actual instantiation of the server is an implementation detail which should not dilute the security policy.

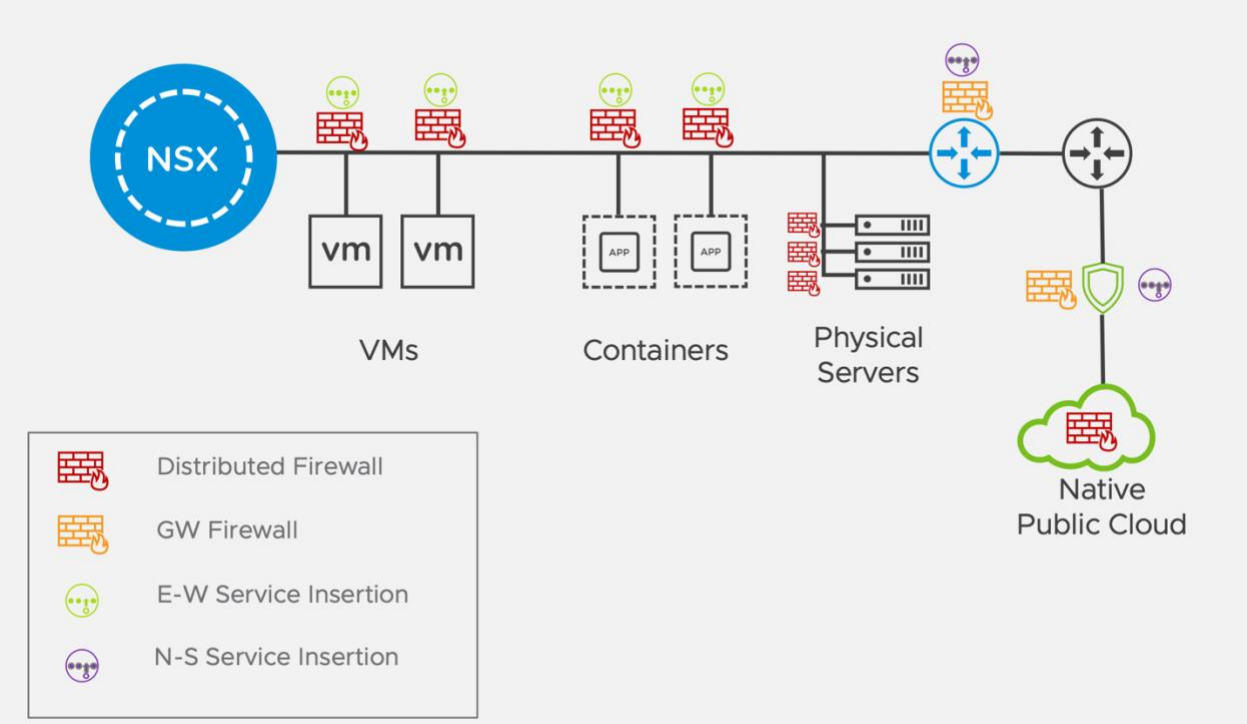

Figure 7 - 2 NSX-T Service Insertion

For a complete list of the currently supported vendors for Service Insertion, see the VMware Compatibility Guide.

7.2.1 North-South Service Insertion

The first step in integrating NSX with your existing firewall vendor is to determine which deployments are supported. In the case of North-South service insertion this is fairly straightforward as the gateway firewall are central data planes which are very much in line with legacy firewalling models. North-South Service Insertion is available at both the Tier-0 and Tier 1 routers. Figure 7 - 3 depicts the typical supported deployment model for North-South Insertion. In this figure, the Service Insertion rule is applied at the Tier 0 gateway.

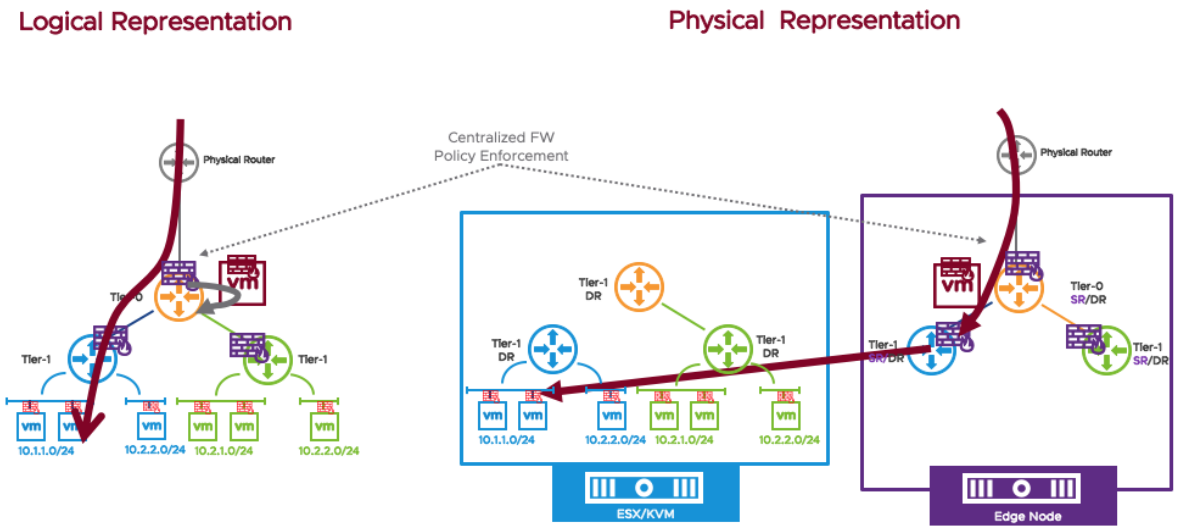

Figure 7 - 3 NSX-T North South Service Insertion

This model suggests the deployment of the VM form factor of the legacy firewall alongside the Gateway firewalls on the Edge Nodes. This suggestion would minimize the need for traffic to exit the host for processing by the virtualized legacy firewall. Note that when the NSX firewall and the gateway firewall are coresident, this means that the additional delay in traffic processing by the additional security element is a matter of microseconds as nothing is traversing wires, contending with network traffic. Traffic sent from the NSX gateway firewall to the VM firewall arrives in a matter of microseconds, dependent solely on CPU load. Upon successful processing by the VM, traffic returns to the NSX gateway to be routed on its path. Again, this processing required no modification to routing or any network infrastructure.

Once the supported deployment is verified, the configuration of service insertion involves just three simple steps:

1. Register the Service with the NSX Manager.
2. Deploy a Service for North South Introspection.
3. Add Redirection Rules for North South Traffic.

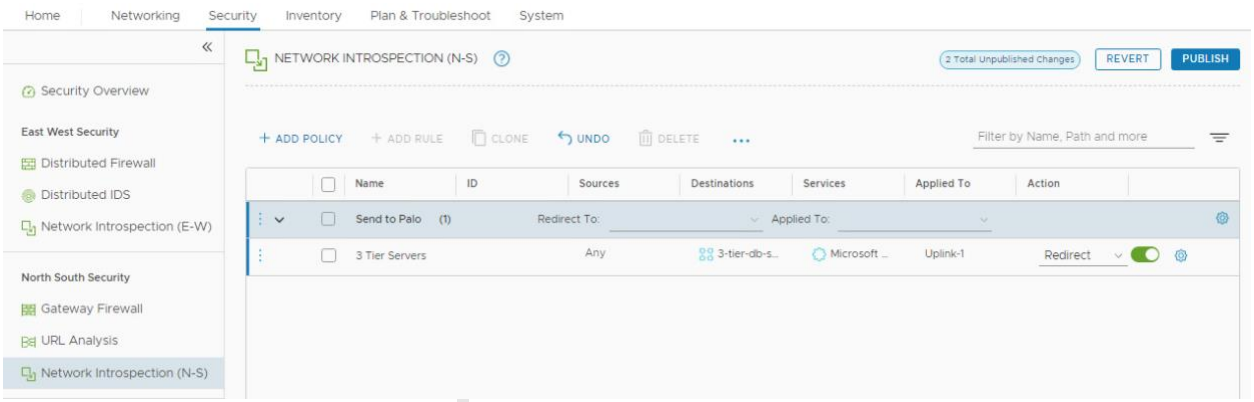

Figure 7 - 4 NSX-T North South Service Redirection Rule

Figure 7 - 4 shows a service redirection policy. You will notice that this policy has sections defined by which SVM the traffic is redirected to. It is entirely possible to have more than one entity or vendor to which traffic is redirected. Under each section, rules are defined for the traffic that will be redirected or NOT redirected. Note that if your Edges are running in HA mode, you need to create a redirection rule for each Edge Node. NSX does not automatically apply the redirection rule to the standby node in the event of a failover as not all vendors support failing over the service VM.

As part of the SI integration, the NSX Manager will update the partner manager with changes to group membership. In other words, the state is automatically synchronized to ensure consistent processing. For some customers, this provides a great way to start NSX and legacy firewall integration. This extends the inventory and dynamic grouping constructs into their legacy firewall environment. The next step of the adoption would be to use the North-South insertion where the Gateway firewall becomes a means to reduce the processing burned on their legacy firewalls.

7.2.2 East West Service Insertion and Service Chaining

East West Service Insertion gets a bit more complicated as the DFW is a distributed firewall. Legacy firewalls have no equivalent model. Because of this, understanding the supported deployment models for your firewall vendor is especially important. Here are a few concepts which are important to keep in mind:

- **Service**: Partners register services with the NSX Manager. A service represents the security functionality offered by the partner, service deployment details such as OVF URL of service VMs, point to attach the service, state of the service.
- **Vendor Template**: It defines the functionality that a service can perform on a network traffic. Partners define vendor templates. For example, a vendor template can provide a network operation service such as tunneling with IPSec service.
- **Service Profile**: It is an instance of a vendor template. An NSX-T Data Center administrator can create a service profile to be consumed by service VMs.
- **Guest VM**: It is a source or destination of traffic in the network – where the packets originated or destined. The incoming or outgoing traffic is introspected by a service chain defined for a rule running East-West network services.
- **Service VM**: A VM that runs the OVA or OVF appliance specified by a service. It is connected over the service plane to receive redirected traffic.
- **Service Instance**: It is created when a service is deployed on a host. Each service instance has a corresponding service VM.
- **Service Segment**: A segment (overlay or VLAN backed) of a service plane that is associated to a transport zone. Each service attachment is segregated from other service attachments and from the regular L2 or L3 network segments provided by NSX-T. The service plane manages service attachments.
- **Service Manager**: It is the partner manager that points to a set of services.
- **Service Chain**: Is a logical sequence of service profiles defined by an administrator. Service profiles introspect network traffic in the order defined in the service chain. For example, the first service profile is firewall, second service profile is monitor, and so on. Service chains can specify different sequence of service profiles for different directions of traffic (egress/ingress).
- **Redirection Policy**: It ensures that traffic classified for a specific service chain is redirected to that service chain. It is based on traffic patterns that match NSX-T Data Center security group and a service chain. All traffic matching the pattern is redirected along the service chain.
- **Service Path**: It is a sequence of service VMs that implement the service profiles of a service chain. An administrator defines the service chain, which consists of a pre-defined order of service profiles. NSX generates multiple service paths from a service chain based on the number of locations of the guest VMs and service VMs. It selects the optimum path for the traffic flow to be introspected. Each service path is identified by a Service Path Index (SPI) and each hop along the path has a unique Service Index (SI).

For east west service insertion, one has typically two options: a Service Cluster or a Host-Based model. These two options are shown in Figure 7 - 5 and Figure 7 - 6, below both depicting the same flow between tenants in DFW that were examined in chapter 4.

In a per host deployment (as shown in Figure 7 - 5), an instance of the SVM is installed on each host in the ESXi Cluster. Traffic between guestVMs on the same host is inspected without ever having to leave the host. This clearly offers a significant processing advantage to the clustered model, with a greater licensing cost.

Figure 7 - 6 shows a Service Cluster model. In a clustered deployment, the service VMs are installed on one single cluster. Traffic between the VMs is redirected to the service cluster for policy inspection and enforcement before reaching its final destination. When configuring a cluster deployment, you can specify which particular host within the cluster the traffic should be redirected to (if there is a desire to segregate traffic while undergoing security policies), or you can select any and NSX will select the optimal host.

It is important the note that the two models may coexist in different clusters of the same installation. For example, one may have a cluster of DB VMs where every VM will require processing and may go with a host model for that cluster. Another cluster may have a mixture of general population VMs and only a small portion of traffic or even traffic which is not very delay sensitive is being inspected. In this cluster, the service model may the preferred architecture.

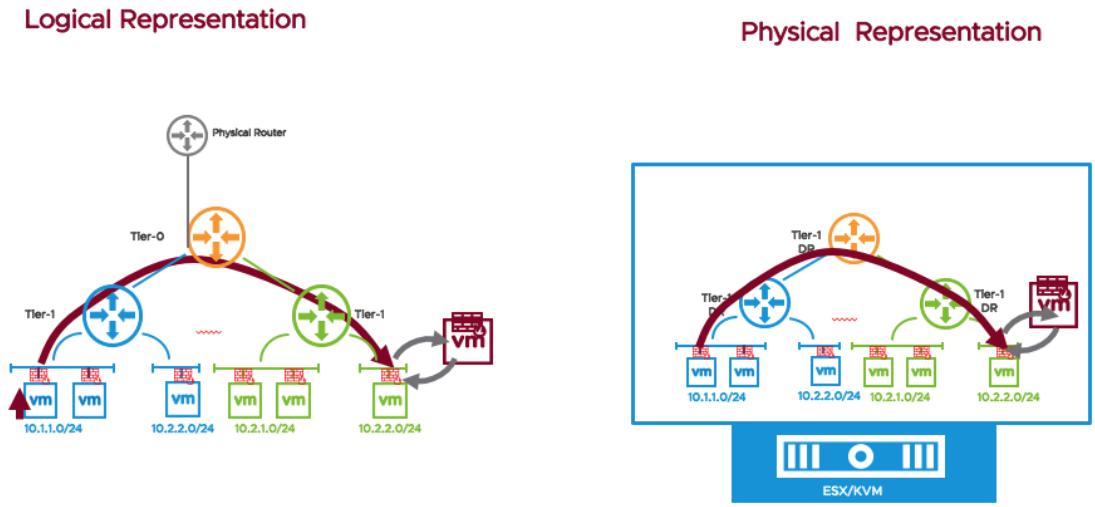

Figure 7 - 5 East West Service Insertion Per Host Model

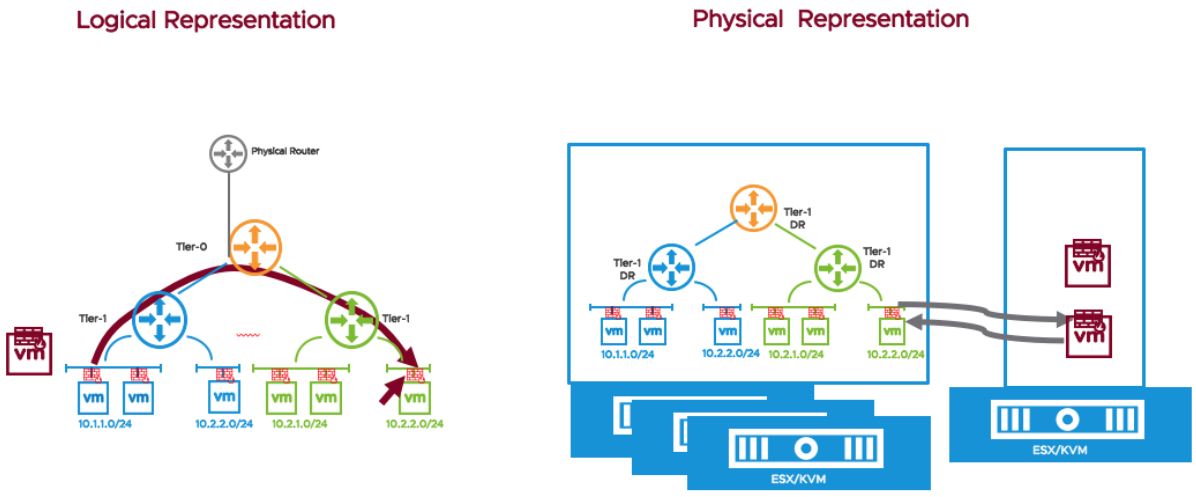

Figure 7 - 6 East West Service Insertion Service Cluster Model

In order to support East-West Service Insertion, at least one overlay transport zone with overlay logical switches must exist. All transport nodes must be of the type overlay because the service sends traffic on overlay-backed logical switches. (This is how the magic happens: NSX internally creates an infrastructure which allows sending the traffic around without the need to modify the existing infrastructure.) The overlay-backed logical switch is provisioned internally to NSX and is not visible to the user interface. Even if you plan on using only VLAN-backed logical switches for the Guest VMs, the service insertion plumbing passes traffic being processed through the overlay. Without this overlay infrastructure, a guest VM which is subject to east west service insertion cannot be vMotioned to another host and would go into a disconnected state.

Deploying East-West Service Insertion is slightly more involved than deploying North-South. The following steps are required to set up East-West service insertion:

1. Register the Service, Vendor Template, and Service Manager
2. Deploy a Service for East West Introspection
3. Add a Service Profile
4. Add a Service Chain
5. Add Redirection Rules

With East west service insertion, it is possible to string multiple services together to provide service chaining. Service Chaining provides standards-based delivery and flexible deployment options. Figure 7 - 7 below shows a service node with NGFW, IPS, and Network Monitoring services for service chaining. A flow may leverage one, two, or all three services as defined by the rules in the service insertion policy. Although service chaining is defined in the east West Security section, under the DFW, the dynamic service chain is attached to the T-0/T-1 Services Router (where the Tier-1 gateway firewall lives). Classification and redirection of traffic to the Services Plane happens at the T0/T1 uplink, which means service chaining is applied at the gateway. Note that Service Chaining provides support to north south traffic coming to and from VMs and Kubernetes containers.

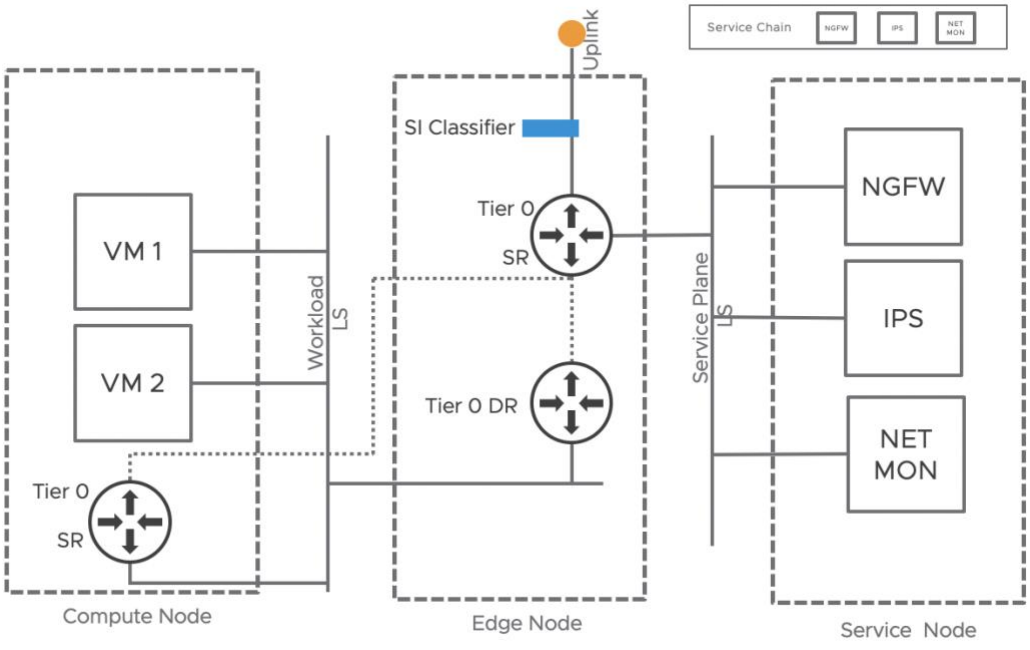

Figure 7 - 7 NSX-T Service Chaining

This case is similar to regular N/S SI, but instead of redirecting traffic to a bump-in-the-wire N/S service, a service chain is used instead. SI classification and redirection happens in the same location as regular N/S SI in the packet processing pipeline.
Given that the SI lookup happens on the uplink, processing will use IN/OUT directions as appropriate for the uplink itself. IN means the packet is being received from the internet, OUT mean the packet is being send to the internet through the uplink. This is the same as regular N/S SI.

Service Chaining Compare to Service Insertion:

- Support for additional use-cases/vendors
- Chaining of multiple services versus a single service
- Leverage the same service chain and service instances (SVMs) for multiple Logical Routers and E-W Service Insertion
- Support for Liveness detection
- No HA (Active/Standby), but load distribution with flow pinning

	N-S Network Introspection	E-W Network Introspection	N-S Network Introspection (Service Chaining)
Use Cases	SDDC/Tenant Perimeter, Kubernetes Namespaces	Advanced Security and Visibility Controls for Micro-segmentation,	SDDC/Tenant Perimeter, Kubernetes Namespaces
Partner Services	NGFW (IPS, Botnet filtering, URL Filtering)	NGFW, Network Visibility, Network Performance Management	NGFW, Network Visibility, Network Performance Management
Protected Workloads	Workload behind T0/T1 gateway on prem	K8S, VMs on ESXi	Workload behind T0/T1 gateway on prem
Traffic Interception	Uplink of T0 / T1 Gateway	Logical Port (VM vNIC/Container Interface)	Uplink of T0 / T1 Gateway
Transport	Layer 2 (Bump in the Wire)	Service Plane (NSH/Geneve)	Service Plane (NSH/Geneve)
SVM Placement	ESXI TN (Placement close to Edge)	Distributed on each ESXI Compute TN or Centralized on ESXI Service Cluster	Centralized ESXI Service Cluster
High Availability Support	Active/Standby	Load Distribution across multiple Service Instances	Load Distribution across multiple Service Instances
Service Chaining	No	Yes	Yes
Redirect / Copy Support	Redirect	Redirect and Copy	Redirect and Copy

Figure 7 - 8 NSX-T Network Introspection and Service Chaining Deployment Options

7.3 NSX Endpoint Protection – Guest Introspection

NSX-T provides the Endpoint Protection (EPP) platform to allow 3rd party partners to run agentless Anti-Virus/Anti-Malware (AV/AM) capabilities for virtualized workloads on ESXi. Traditional AV/AM services require agents be run inside the guest operating system of a virtual workload. These agents can consume small amounts of resources for each workload on an ESXi host. The Endpoint Protection platform allows the AV/AM partner to remove their agent from the virtual workload and provide the same services using a Service Virtual Machine (SVM) that is installed on each host. These SVMs consume much less virtual CPU and memory overall than the many running agents on every workload on the ESXi host. This chapter focuses on NSX-T Endpoint Protection capabilities:

- Platform for Partner integration for Agentless AV/AM deployments
- Use cases covered for EPP
- Architecture details
- Windows EPP vs Linux EPP
- Workflows – Registration, Deployment, Consumption

- Designing consistent EPP Policies across vCenter Server inventories
- Designing granular, cluster-based Policy and Partner SVM deployment

7.3.1 NSX Endpoint Protection – Architecture and Components

The high-level Endpoint Protection Architecture consists of the following components which are mandatory for NSX-T Endpoint Protection deployment. These components represent the items which an NSX-T administrator would configure or interact with the most for using the Endpoint Protection platform.

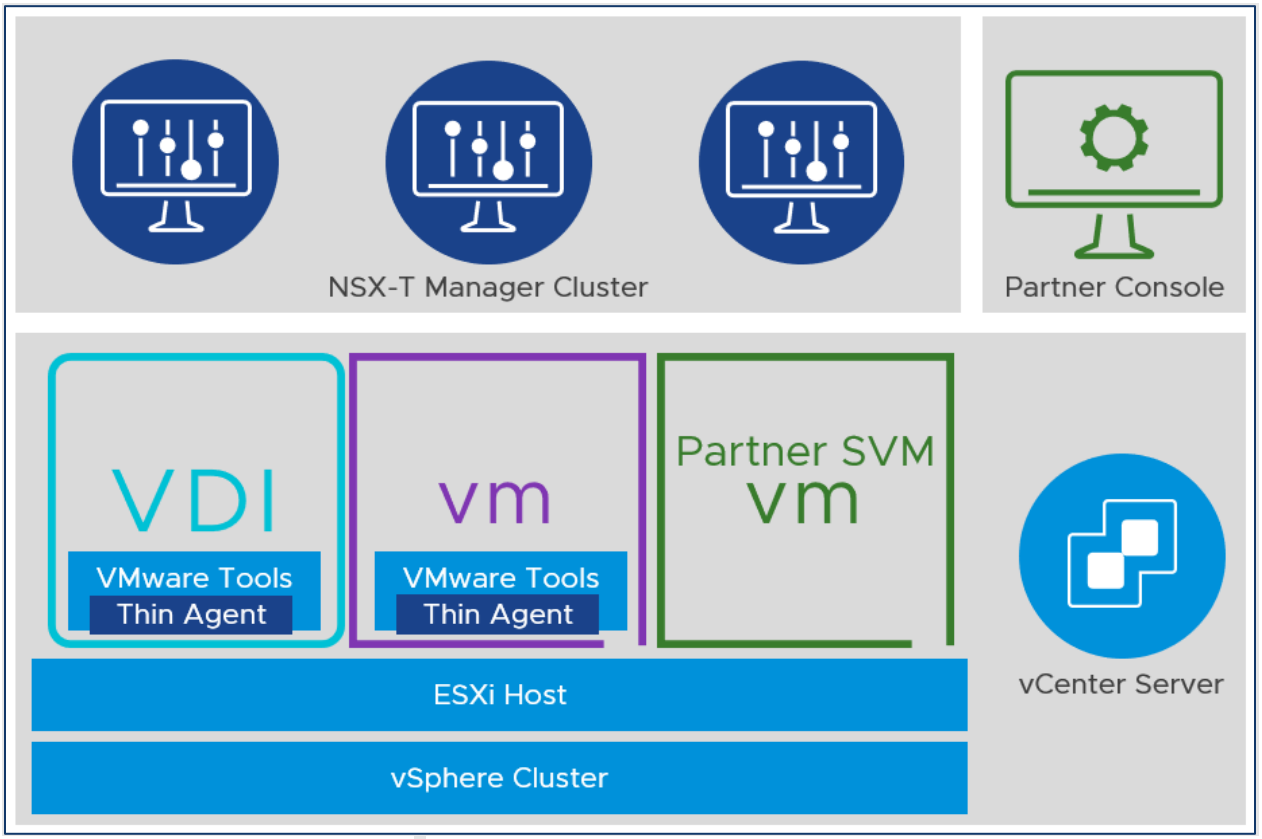

Figure 7 - 9 NSX-T Endpoint Protection Architecture – High-Level

- **NSX-T Manager Cluster**
  - The cluster of NSX-T Managers which the Partner console will interact with via REST API commands
  - Provides the User Interface for configuring Groups, Service Deployments, and Endpoint Protection Policies for virtual machine workloads
- **Partner Console**
  - Registers via REST API with the NSX-T Manager cluster
- **VMware Tools with Thin Agent**
  - Two drivers for file and network inspection deployed as part of the VMware Tools installation, needed to send events and information to the Partner SVM
- **Partner SVM**

    - The Partner provided virtual machine appliance that contains the partner's anti-malware engine.
- **ESXi Cluster**
    - Endpoint Protection currently only supports ESXi-based workloads and the hosts must be in a vSphere Cluster, even if only 1 host resides. A Partner SVM is deployed to ALL hosts within that vSphere cluster.
- **vCenter Server**
    - vCenter Server provides the management plane for ESXi hosts and clusters.
    - vCenter Server assists with the deployment and protection of the Partner SVMs using ESXi Agent Manager
- **VSS/VDS Portgroup or N-VDS Segment (Refer to** Figure 7 - 11**) – Management Plane Interface**
    - The VSS/VDS portgroup can used for connecting the Management network interface of the Partner SVM for communication to the Partner Console
    - The NSX prepped portgroup in the VDS, N-VDS Segment, Overlay or VLAN, can be used for connecting the Management network interface of the Partner SVM for communication to the Partner Console
- **vmservice-vswitch (Refer to** Figure 7 - 11**) – Control/Data Plane switch**
    - A standard vSphere Switch that provides a vmkernel port for the content multiplexer to communicate with the Partner SVM. Must run on a vSphere Standard Switch. Not configurable.
- **vmservice-vshield-pg (Refer to** Figure 7 - 11**) – Control/Data Plane portgroup**
    - A standard vSphere Switch port group located on the *vmservice-vswitch* that the Partner SVM connects the Control/Data Plane interface to. Must run on a vSphere Standard Switch. Not configurable.
- **NSX-T Transport Node Profile (Not Pictured) –** An NSX-T Transport Node profile provides a consistent configuration of the Transport Nodes (ESXi Hosts prepared with NSX-T) in the vSphere Cluster. This profile ensures any new host that joins a vSphere Cluster automatically has a Partner SVM deployed to the host for protecting workloads.
- **IP-Addressing Mechanism (Not Pictured) –** IP Addresses for the Partner SVM are necessary for the SVM to communicate to the Partner Console. These can be provided by NSX-T via IP Pool, or through a customer DHCP server.

Breaking each of these components down further and dividing them into their planes of operation, one can take a closer look at the internal components.

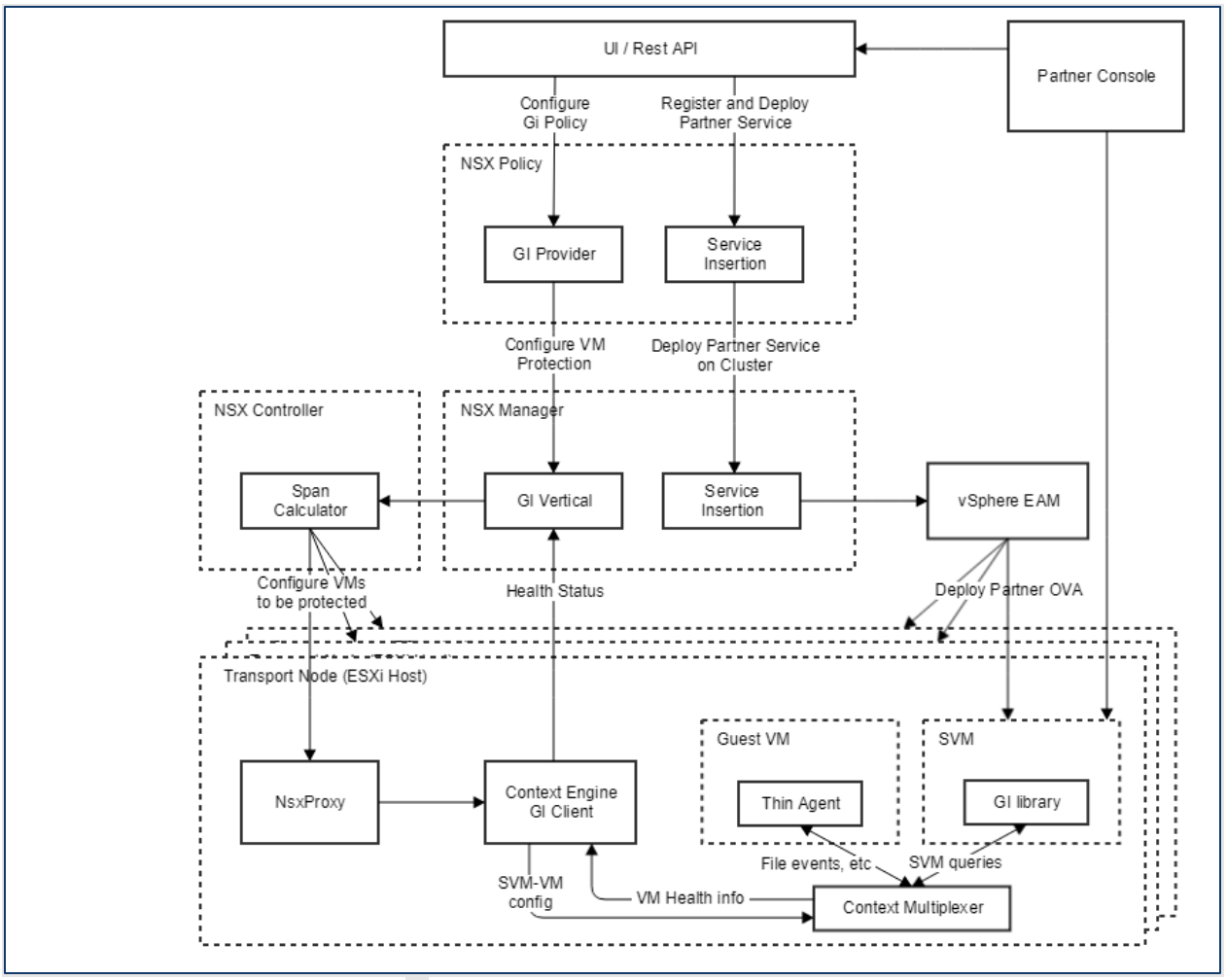

Figure 7 - 10 NSX-T Endpoint Protection Architecture - Low-level

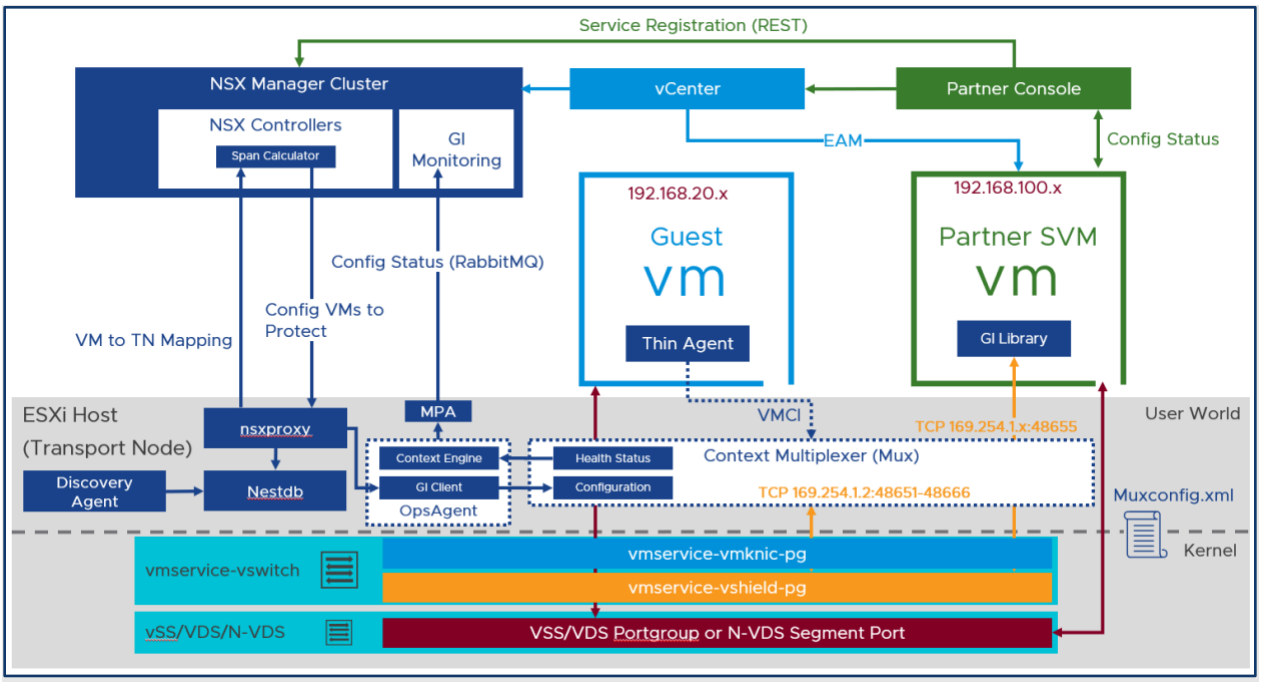

Figure 7 - 11 NSX-T Endpoint Protection Architecture - Including Networking

Figure 7 - 11 shows additional components of the NSX-T Endpoint Protection Architecture, specifically the ESXi host network configuration.

7.3.2 User Interface/REST API

The Endpoint Protection Platform User Interface is accessed through NSX-T Policy and REST API calls are made to the NSX-T Policy API. A dashboard is supplied under the Security tab for Endpoint Protection that supplies information around the deployments, components having issues, and configured VMs.

Figure 7 - 12 NSX-T Endpoint Protection Dashboard

7.3.3 Management Plane Components

The GI Provider is the component responsible for composing Endpoint Protection Policies and interacting with the Management Plane GI Vertical. It resides inside the NSX-T Manager(s) that constitute the NSX-T Management Cluster. Endpoint Protection leverages Service insertion for inserting partner services onto the NSX-T Transport Nodes. Each host has a vSphere ESX Agent Manager installed and configured to manage the Partner SVM lifecycle and protect the virtual machine. Finally, the GI vertical configures policies on NSX-T Groups of VMs and sends this configuration to the CCP Span Calculator.

7.3.4 Control Plane Components

The NSX-T Control Plane components consist of the Centralized Control Plane (CCP), that resides in the NSX-T Manager(s) and the Local Control Plane (LCP) that resides in each ESXi host. For NSX-T Endpoint Protection, the CCP pushes the VM Group configuration and subsequently the Endpoint Protection Policy, to the LCP of the hosts where the VMs reside. The CCP calculates the span for the NSX-T Group(s) of VMs and sends this information to the LCP on appropriate hosts.

7.3.5 Data Plane

The Data Plane of the NSX-T Endpoint Protection platform resides in several components. These components represent the plane in which the files, events, and information actually 'flow' for processing by the Endpoint Protection Platform and the Partner Service associated.

7.3.5.1 Thin Agent

The Thin Agent is a set of two drivers that are installed as part of the VMware Tools 'Complete' installation or by selectively installing them using the 'Custom' installation. For Windows machines, this is done via the following:

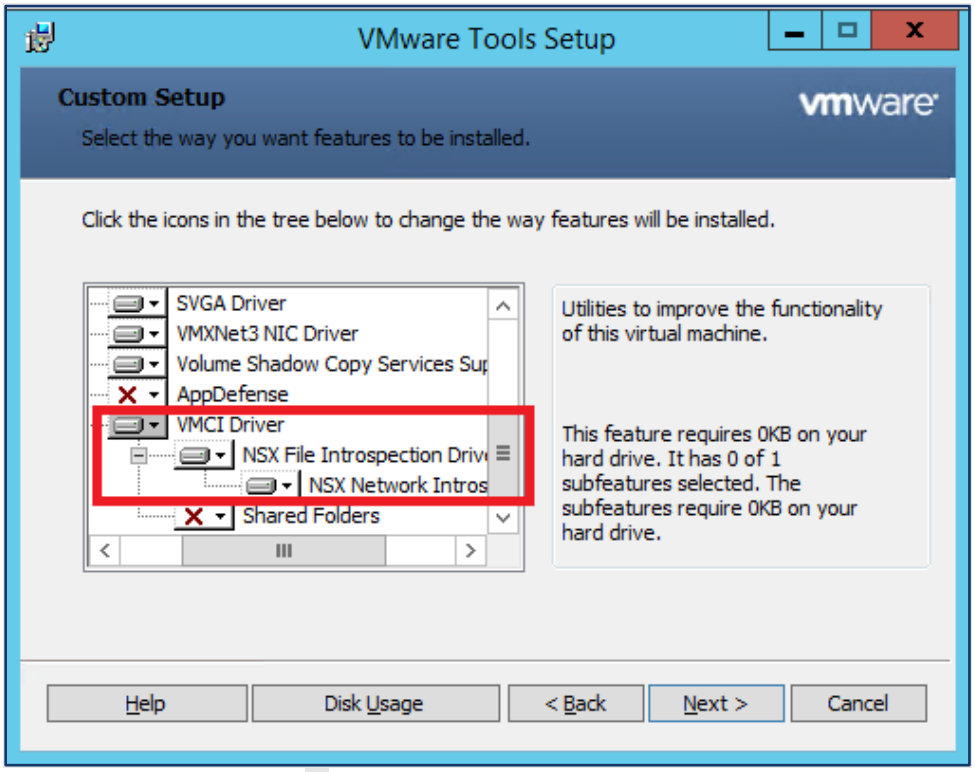

Figure 7 - 13 NSX-T Endpoint Protection Thin Agent Install

For Linux-based workloads, VMware Tools is not required, only the Endpoint Protection thin agent package is required. The package is accessed from https://packages.vmware.com/packages/nsx-gi/latest/index.html for the appropriate Linux operating system. The service created on the Linux system is located in ***/etc/init.d/vsepd.***
For a list of supported Operating Systems, please refer to the NSX-T Administrator Guide – Endpoint Protection.

7.3.5.2 GI Library

This is the library that is linked with the Partner SVM and acts as an interface between the Partner SVM and the Thin Agent for their communications.

7.3.5.3 Context Multiplexer (Mux)

This component is responsible for forwarding Thin Agent events to the configured Partner SVMs. It also forwards Partner SVM requests to the Thin Agent.

7.3.5.4 Context Engine GI Client

This component is responsible for sending Thin Agent and Mux health status to the GI Vertical.

7.3.5.5 Muxconfig.xml file

This file is used to track the Partner Service(s) that are deployed as well as the virtual machines configured for each service on the ESXi host. As machines are powered on and off, they are added and removed from the muxconfig.xml to enable and disable protection from the Partner Service.

7.3.6 Partner Components

NSX-T Endpoint Protection provides the platform for VMware certified partners to integrate their partner services. The following section goes into details around the necessary component from the VMware partner, that communicate with the NSX-T Endpoint Protection Platform.

7.3.6.1 Partner Management Plane

The Management Plane for the Partner Service is the Partner Console. The Partner Console is typically deployed as an OVA virtual machine and can be placed in a compute cluster, but generally placed into the management cluster for protection similar to other management plane appliance such as NSX-T Manager.

7.3.6.2 Partner Control/Data Plane

The Control/Data Plane for the Partner Service is comprised of the Partner Service VM (SVM). The Partner SVM is deployed on each ESXi host in a cluster. This SVM contains the partner's Anti-Malware engine.

7.3.7 Workflow Object Definitions

Before discussing NSX-T Endpoint Protection deployment, enforcement, and workflows, the objects that are configured and their definitions are required.

- **Deployment Template** – Partner Template that tells the Partner SVM deployment where to connect to the Partner Console and over which ports
- **Deployment Specification** – Partner SVM metadata and sizing characteristics
- **Service Deployment** – Represents the configuration details of all the necessary objects to perform a deployment of the Partner SVM. Contains the Computer Manager where the cluster, network, and data store reside for Partner SVM Deployment. Also contains the Deployment Specification and Deployment Template of the Partner SVM.
- **Service Instance** – Represents the Partner SVM deployments and the associated data about their, host location, deployment mode, deployment status, and health status.
- **Catalog** – Lists the Partner registrations that have been configured with NSX-T Manager
- **Service Profile** – Defines the vendor template that will be used in the Endpoint Protection Policy
- **Vendor Template** – Defines the template created from the Partner Console that contains the protection policy that the Partner will be enforcing on the workloads. This template is passed to the NSX-T Manager for use in the Endpoint Protection Service Profile.
- **Endpoint Protection Policy** – NSX-T Policy that uses the Group and the Service Profile to define the 'HOW' and 'WHAT' for endpoint protection.

Group – Defines the workloads that will be used in the Endpoint Protection Policy and protected.

7.3.8 NSX-T Endpoint Protection Deployment and Enforcement

NSX-T Endpoint Protection provides a robust set of capabilities that provide significant flexibility of deployment options and enforcement.

- **Multiple vCenter Server Support –** NSX-T Endpoint Protection supports a consistent Endpoint Protection Policy across multiple vCenter Server Computer Managers connected to NSX-T. Certified Partner must support multiple vCenter Server connectivity.
- **Cluster-based Endpoint Protection Policy Granularity –** NSX-T Endpoint Protection supports granular, per-cluster, policy deployment and enforcement. A different policy can be applied based on cluster workload needs. Example: VDI Desktop Cluster versus Server Workload Cluster policies.
- **Scalable Partner SVM deployment –** NSX-T Endpoint Protection supports deploying different Partner SVM sizes to different clusters based on cluster workload needs. Partner SVM sizing can reduce the number of resources necessary to perform agentless offload. Example: VDI Desktop Cluster versus Server Workload Cluster deployments where consolidation ratios are higher on VDI and may require a larger SVM with more resources to accommodate file-scanning needs.

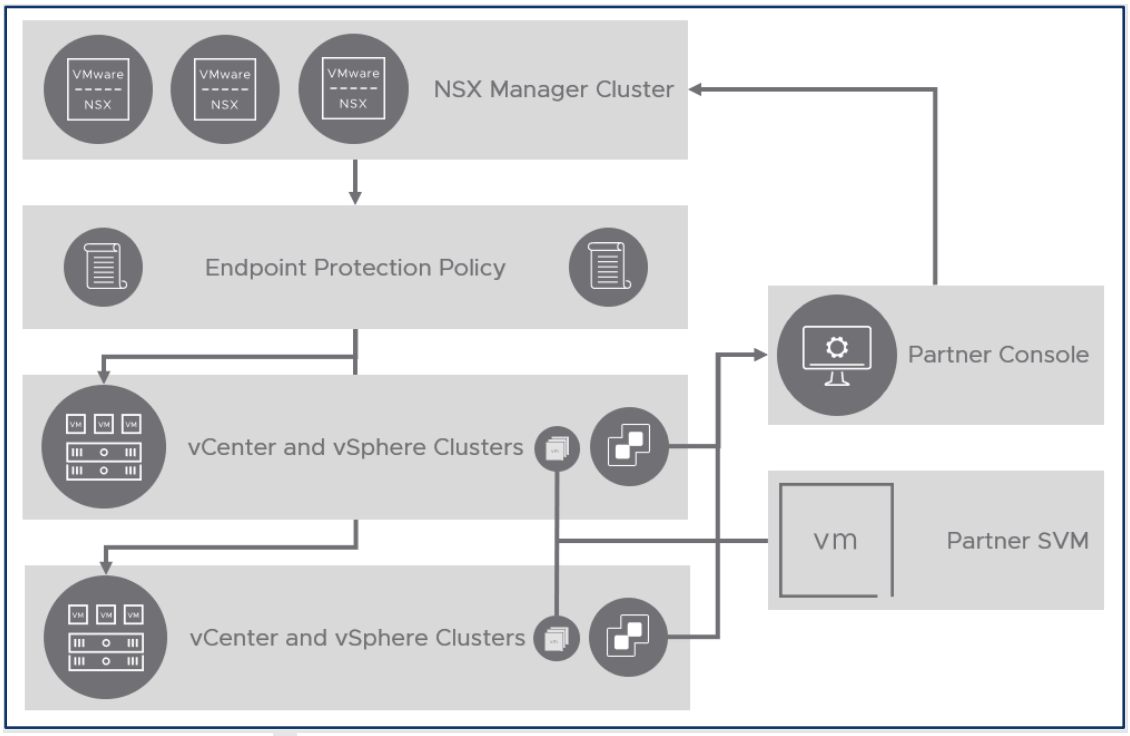

Figure 7 - 14 NSX-T Endpoint Protection Policy - Multi-vCenter Server Consistent

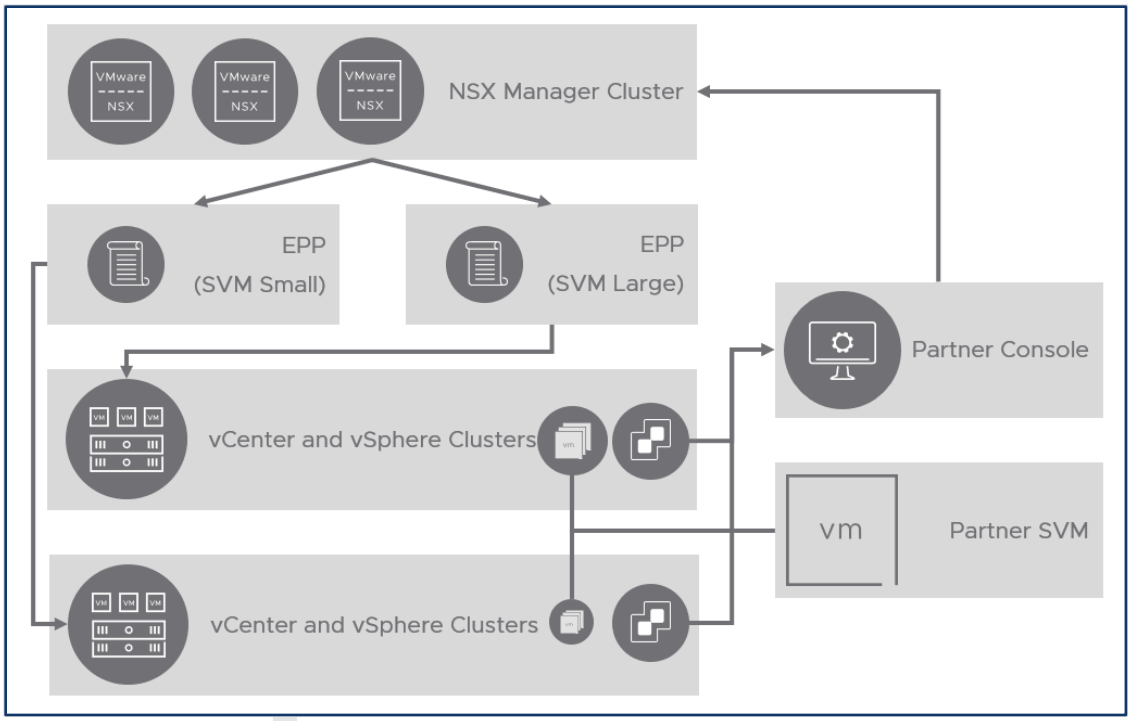

Figure 7 - 15 NSX-T Endpoint Protection - Cluster Granular and Partner Scalable

- **Workload/Partner SVM Networking Agnostic –** NSX-T Endpoint Protection supports workloads that reside on either VSS/VDS or N-VDS networking and supports deploying the Partner SVM on these same networking constructs.

7.3.9 NSX-T Endpoint Protection Design Considerations

The flexibility options in deployment and enforcement of NSX-T Endpoint Protection bring up specific design considerations prior to deployment. Before going into the design considerations in detail, it makes sense to call out a configuration detail, specific to Endpoint Protection.
There are very specific ESXi host configurations that can impact a design of the NSX-T Endpoint Protection deployment. ESXi hosts have settings local on each host where Agent VMs, specifically Endpoint Protection Partner SVMs, can be placed on a specific datastore and network that's locally significant to the ESXi host or part of shared networks and datastores present to other hosts. Generally, these settings are not needed and Service Deployment from NSX-T Manager will overwrite any locally controlled settings on the ESXi host. While these options are supported, they do not represent the majority of deployments and recommended options as they do not scale and are error-prone due the manual nature of configuration and the need to touch every ESXi host. The following sub-section will describe these options and how to use them, but the rest of the section will be based on the recommended deployment option of configuration through the NSX-T Manager.

7.3.9.1 Agent VM Settings in ESXi/vCenter Server

It is possible, although not recommended as a primary use case, to deploy the Partner SVMs from NSX-T, to locally specified networks and data stores on the ESXi host. These settings are configured on EACH ESXi individually in the Agent VM Settings Configure options. You can configure these options from vCenter Server and each host as well.

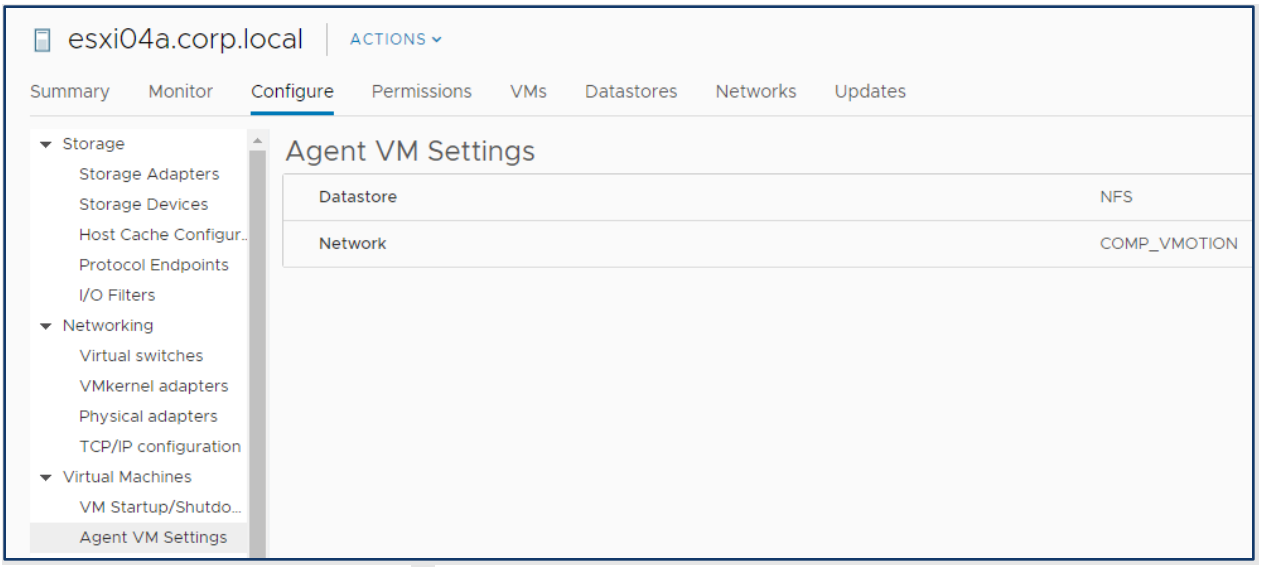

Figure 7 - 16 NSX-T Endpoint Protection - ESXi Agent VM Settings

7.3.9.2 Agent VM Settings in NSX-T

If local ESXi Agent VM Settings are used, the NSX-T Endpoint Protection Service Deployment needs to be configured appropriately and the 'Specified on Host' option used for the data store and management network.

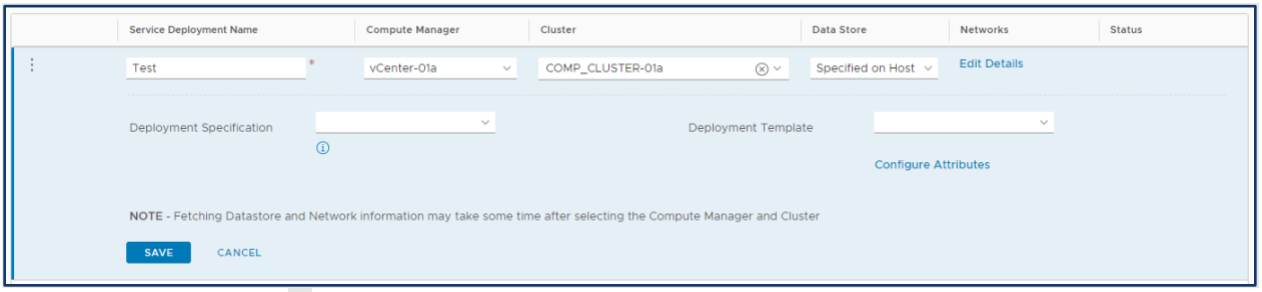

Figure 7 - 17 NSX-T Endpoint Protection - Service Deployment Specified on Host Data Store

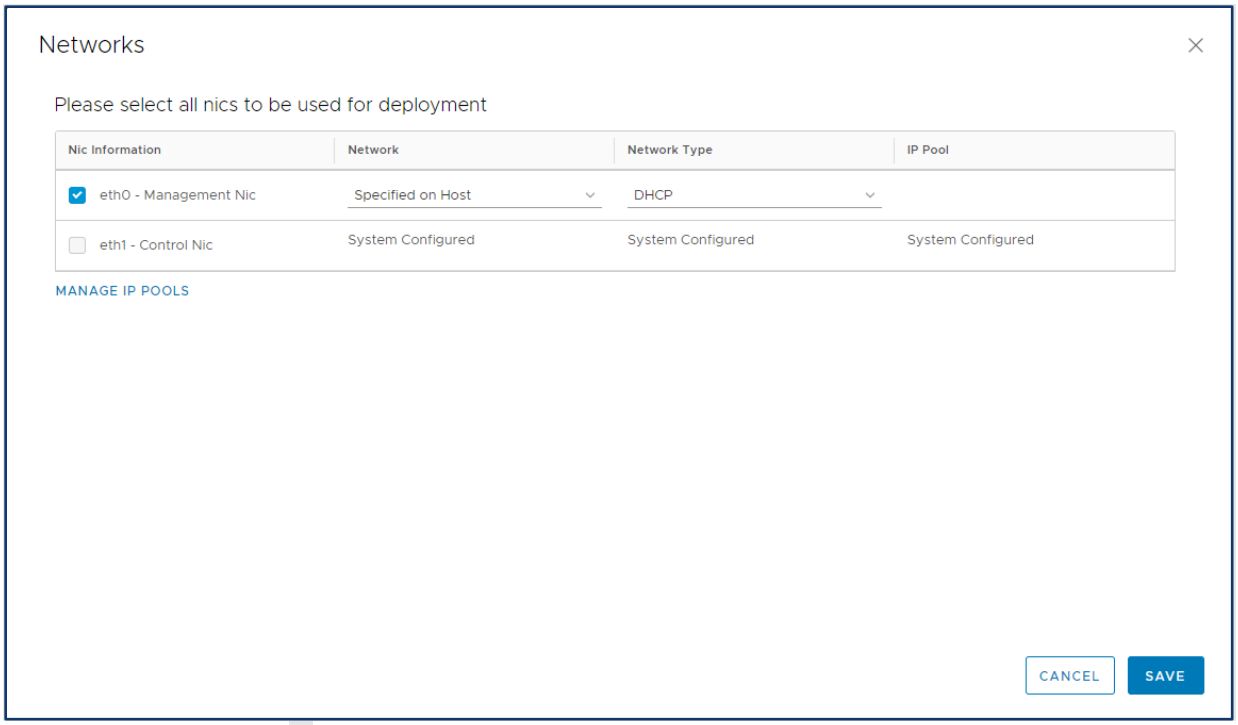

Figure 7 - 18 NSX-T Endpoint Protection - Service Deployment Specified on Host Network

7.3.9.3 Workload and Partner SVM Networking

NSX-T Endpoint Protection enforcement can be achieved for workloads on VLAN Backed and Overlay NSX-T Segment types and is also unique in that it does not require either of these segment types. NSX-T Endpoint Protection can provide protection to workloads residing on VSS/VDS Portgroups as well.

The Partner SVM that is deployed requires two network connections:

- **Management vNIC** – Connects to either a VSS/VDS Portgroup or an N-VDS VLAN or Overlay Segment.
- **Control vNIC** – Connects the Partner SVM to the Mux inside the ESXi host. Not configurable and automatically created on Service Deployment to the host

Regardless of networking construct used, the Management vNIC of the Partner SVM must be able to communicate with the Partner Console.

The Partner SVM requires an IP Address mechanism to provide the IP for the Management vNIC. This can be achieved by:

- **DHCP Server** – Customer hosted DHCP appliance/IPAM
- **NSX-T IPAM (IP Pool)** – NSX-T can provide an IP Pool and requisite configuration options for the Partner SVM to pull from. (This is done from the IP Management selection in the Networking option of the UI).

Regardless of IP Addressing mechanism used, the number of IP addresses in either the DHCP Scope or the NSX-T IPAM IP Pool should be sufficient to cover the number Partner SVMs deployed.

7.3.9.4 Partner SVM Data Store and Compute

The data store which the Partner SVM will be placed on is recommended to be shared across the entire cluster that is being deployed to, and provides enough disk space that will be able to host the size of the SVM multiplied by the number of hosts in the cluster. The size of the disk that each Partner SVM requires differs per partner. Consult the partner documentation to understand the disk requirements.

Partner SVMs are deployed to all hosts in a vSphere cluster. If a new host is added to the cluster, EAM triggers a deployment of a new Partner SVM to reside on the host and provide the same Endpoint Protection as assigned to all other hosts in the vSphere cluster.

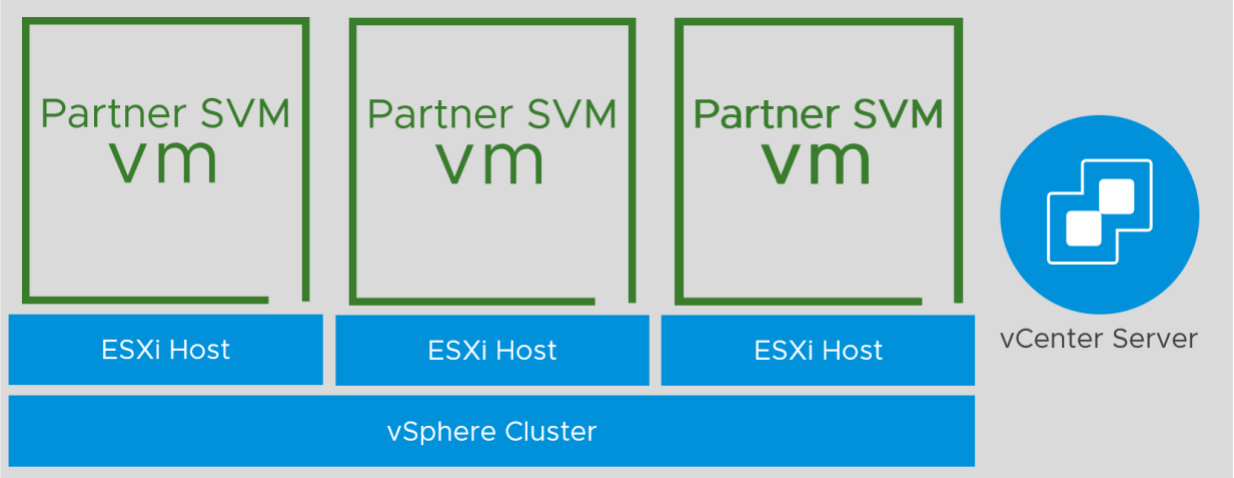

Figure 7 - 19 NSX-T Endpoint Protection - Partner SVM Cluster Deployment

7.3.9.5 Partner Console

The Partner Console is recommended to reside on a management cluster with vSphere HA configured to provide redundancy. Please consult the specific partner documentation on recommended high-availability configurations.

7.3.9.6 Service Deployment Restriction and Support

NSX-T Endpoint Protection Service Deployments are on a per-cluster, per-vCenter Server basis. One Service Deployment is required for each cluster. If a Partner provides more than one Deployment Specification, i.e. SVM size, selection of the appropriate size is recommended based on the cluster workloads that are hosted.

Once a Service Deployment is deployed, the NSX-T/vCenter Server specific options cannot be changed. Only the Partner Deployment Specification and Deployment Template can be changed. **If either of these options are changed, a redeployment of the Partner SVMs will occur and protection will be lost while redeployment is taking place.** Changing networks of the Partner SVMs is not supported. The recommendation is to remove Service Deployment and recreate on new data store. Storage vMotion of the Partner SVMs is supported, however any redeployment will result in the Partner SVMs attempting to be put back on the configured Service Deployment data store. The recommendation is to remove the Service Deployment and recreate on new data store.

7.3.9.7 Groups

NSX-T Groups define the workloads that will be protected by the Endpoint Protection Policy. Size of Groups follow the configuration maximums that are documented here. Considering that Groups can contains VMs that reside on hosts outside of Endpoint Protection and VMs can be part of multiple Groups, it is recommended to create new Groups that align to the VMs on protected clusters. Multiple Groups can be associated with the same Endpoint Protection Rule.

7.3.9.8 Service Profile

A partner can specify multiple templates that can be used based on the workload type that's being protected. It is required to create at least one Service Profile that will be used in an Endpoint Protection Policy. Multiple Service Profiles should be used when it's necessary to have more than one Endpoint Protection Policy. Example: VDI Service Profile and Endpoint Protection Policy and Server Service Profile and Endpoint Protection Policy. Only one Service Profile can be specified in an Endpoint Protection Rule.

7.3.9.9 Endpoint Protection Policy

NSX-T Endpoint Protection Policy provides the Endpoint Protection Rules that govern the protection over the Groups of workloads contained and apply a specific Service Profile. An Endpoint Protection Policy can have more than one Endpoint Protection Rule and, in each rule, the same or a different Service Profile. The recommended configuration of an Endpoint Protection Policy would be to group like policies with the same Service Profile into one Endpoint Protection Policy. This helps with troubleshooting and consistent deployment models.

NSX-T Endpoint Protection Rules are defined within an Endpoint Protection Policy and include one or more NSX-T Groups and exactly one Service Profile. Recommended configuration would be to add all of the groups necessary that are part of the same Service Profile, to the same Endpoint Protection Rule. Check the maximum amount of VMs that Endpoint Protection can support per NSX-T deployment and Group maximums documented here.

7.3.10 Endpoint Protection Workflow: Registration, Deployment, and Consumption

The Registration step of the Endpoint Protection Workflow is performed from the Partner Console. The Partner Console needs to register with the NSX-T Managers and the vCenter Servers where workload protection will be applied. Please consult the Partner documentation for the process of registering NSX-T and vCenter Server.

1. Connect Partner Console to vCenter Server(s)
2. Connect Partner Console to NSX-T Manager
3. Verify Service Definition Registration in Catalog

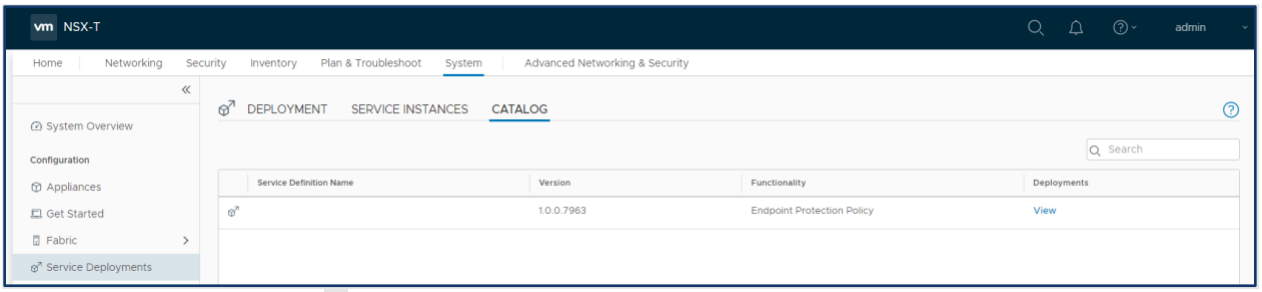

Figure 7 - 20 NSX-T Endpoint Protection Workflow - Partner Registration in Catalog

The Deployment step of the Endpoint Protection Workflow is performed in the Service Deployments > Deployment section of the NSX-T Manager.

1. A Compute Manager vCenter Server is selected
2. A Cluster is selected
3. A Data Store is selected
4. A Management Network is selected
5. An IP Addressing Mechanism is selected
6. A Deployment Specification is selected
7. A Deployment Template is selected
8. Deployment is executed per the configurations selected

Figure 7 - 21 NSX-T Endpoint Protection Workflow - Service Deployment

The Consumption step of the Endpoint Protection Workflow is performed in both the Partner Console and Security > Endpoint Protection Rules section of the NSX-T Manager.

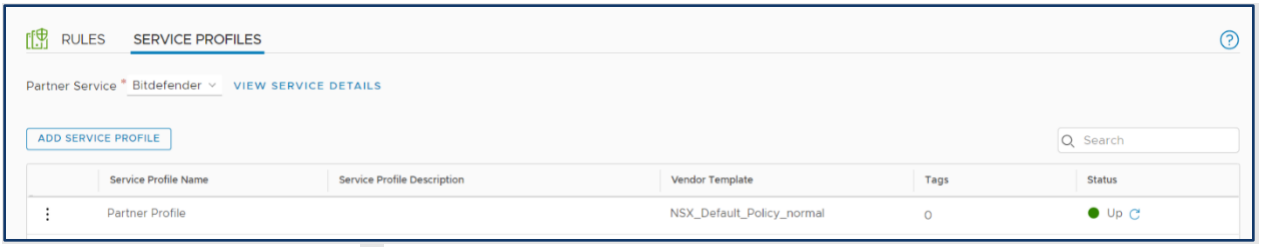

Figure 7 - 22 NSX-T Endpoint Protection Workflow - Service Profile Creation

1. The Partner Console will push default and new Vendor Template policies that are created and marked by NSX-T synchronization to the NSX-T Manager.
2. A Service Profile is created and Vendor Template selected
3. Endpoint Protection Policy created

4. Endpoint Protection Rule created
5. Endpoint Protection Rule Group(s) assigned/created
6. Endpoint Protection Rule Service Profile assigned
7. Publish

7.3.11 Partner Supportability

All partners that are currently certified and supported for the Endpoint Protection Platform are listed on the VMware Compatibility Guide. This is the definitive sources for joint VMware and Partner certified integrations.
https://www.vmware.com/resources/compatibility/search.php?deviceCategory=nsxt&details=1&solutioncategories=28&api=5&page=1&display_interval=10&sortColumn=Partner&sortOrder=Asc

8 Intrusion Detection and Prevention

Much like distributed firewalling changed the game on firewalling by providing a distributed, ubiquitous enforcement plane, NSX distributed IDS/IPS changes the game on IPS by providing a distributed, ubiquitous enforcement plane. However, there are additional benefits that the NSX distributed IPS model brings beyond ubiquity (which, in itself, is a game changer). NSX IPS is IPS distributed across all the hosts. Much like with DFW, the distributed nature allows the IPS capacity to grow linearly with compute capacity. Beyond that, however, there is an added benefit to distributing IPS. This is the added context. Legacy network Intrusion Detection and Prevention systems are deployed centrally in the network and rely either on traffic to be hair pinned through them or a copy of the traffic to be sent to them via techniques like SPAN or TAPs. These sensors typically match all traffic against all or a broad set of signatures and have very little context about the assets they are protecting. Applying all signatures to all traffic is very inefficient, as IDS/IPS unlike firewalling needs to look at the packet payload, not just the network headers. Each signature that needs to be matched against the traffic adds inspection overhead and potential latency introduced. Also, because legacy network IDS/IPS appliances just see packets without having context about the protected workloads, it's very difficult for security teams to determine the appropriate priority for each incident. Obviously, a successful intrusion against a vulnerable database server in production which holds mission-critical data needs more attention than someone in the IT staff triggering an IDS event by running a vulnerability scan. Because the NSX distributed IDS/IPS is applied to the vNIC of every workload, traffic does not need to be hair pinned to a centralized appliance, and one can be very selective as to what signatures are applied. Signatures related to a windows vulnerability don't need to be applied to linux workloads, or servers running Apache don't need signatures that detect an exploit of a database service. Through the Guest Introspection Framework, and in-guest drivers, NSX has access to context about each guest, including the operating system version, users logged in or any running process. This context can be leveraged to selectively apply only the relevant signatures, not only reducing the processing impact, but more importantly reducing the noise and quantity of false positives compared to what would be seen if all signatures are applied to all traffic with a traditional appliance.

NSX distributed IPS brings five main benefits:

- Elastic throughput
- Simplified Network architecture
- Operational Efficiency
- Higher Trigger Fidelity
- Utilize Stranded Compute

As with the NSX DFW, NSX IPS is network independent, and can be used to monitor intrusions for both workloads on traditional VLANs and workloads on Overlay segments. Thanks to the NCP, it can even monitor even Pods inside containers.

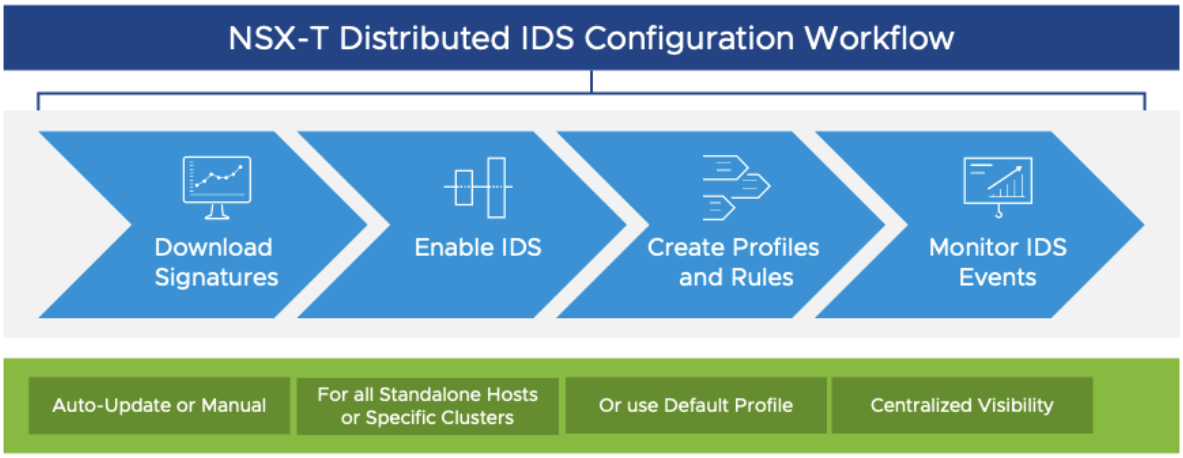

Figure 8 - 1 NSX-T IPS Configuration and Workflow

Configuring NSX IPS involves four steps, as showing in Figure 8 - 1 NSX-T IPS Configuration and Workflow above: Download Signatures, Enable IDS, Create Profiles and Rules, and Monitor events. After describing the IPS components, each step will be examined in detail.

8.1 *NSX IPS Components*

The NSX IPS components are the same as those described above for DFW as IPS functionality is collocated with DFW. In the Management plane, the Manager downloads IPS signature updates from the cloud service and users configure IPS profiles and rules. As with the DFW, the configuration is passed to the CCP after being stored in the Manager. Again, as with DFW, the CCP pushes the information to the LCP on the hosts. At the host, the signature information is stored in a database on the host and configured in the datapath. The ESXi host also collects traffic data and events to pass up to the NSX manager.

Figure 8 - 2 NSX-T IPS Components – LCP and host below shows the detail of the IPS components inside the host.

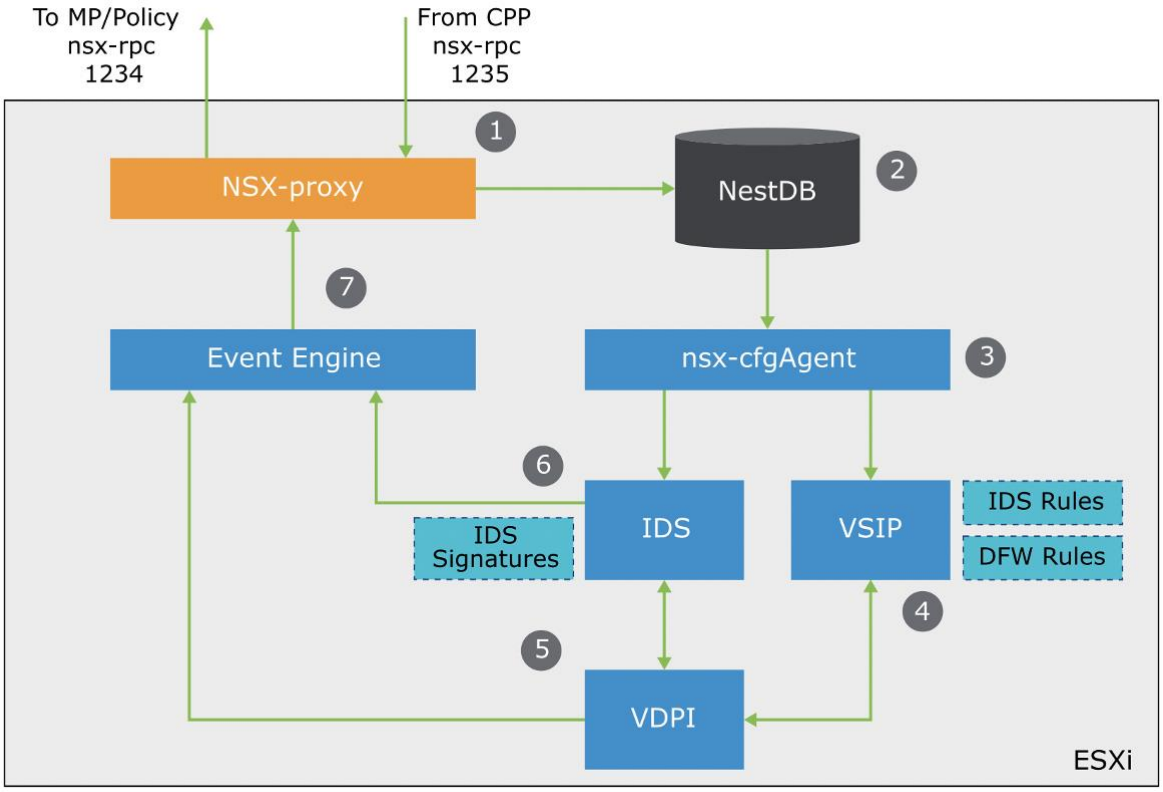

Figure 8 - 2 NSX-T IPS Components – LCP and host

When the configuration arrives at the host, the following takes place.

1. NSX-Proxy obtains configuration changes from CCP and writes data into NestDB.
2. NestDB stores signatures and IPS rules locally.
3. nsx-cfgAgent obtains the configuration from NestDB and writes signatures to IPS and IPS rules to VSIP.
4. vSIP evaluates traffic against IPS "interesting traffic" rules. If a match is found, the packet is punted to the vDPI (Mux)
5. vDPI copies the packet and send the copy through the IPS engine. The packet is released on the vSIP dataplane when IPS finishes inspection
6. IPS Event is generated by the IPS engine
7. Event Engine collects flow metadata and generates alerts.

The event engine is a multi-threaded engine (one thread per host core) deployed on every ESXi TN as part of host-prep which runs in User-space. This engine runs on all ESXi hosts regardless of the enabled state of IPS. (When NSX-T is installed on a host, everything that is required for distributed IDS/IPS to function is installed at that time. No additional software needs to be pushed to the host.) The event engine evaluates traffic against IPS signatures only when IPS is enabled on the TN and IPS Rules are configured. The IPS signatures are configured in profiles and programmed on each IPS Engine. Traffic is mapped to profiles to limit signature evaluation. Note that IPS performance is impacted more so by the inspected traffic, than by the number of signatures which are evaluated. The default set of signatures is programmed on each IPS engine, even when IPS is disabled. For highly secure air-gapped environments, there is support for offline signature update download which involves registration, authentication, and signature downloads in a zip file which can then be manually uploaded via the UI.

8.2 *IPS Signatures*

NSX-T IPS ships with over 11,000 curated signatures. These signatures are currently provided by one of the most well-known Thread Intelligence providers, Trustwave, and are curated based on the Emerging Threat and Trustwave Spiderlabs signatures sets. Because of our pluggable framework, additional signature providers can be added in the future.

Signature ID	Details	Product Affected	Attack Target	IDS Severity	CVSS	CVE	Category
4009306	ET EXPLOIT Possible WINS Server Remote Memory Corruption Vulnerability	Windows_DNS_server	DNS_Server	CRITICAL	0.0		

Figure 8 - 3 NSX-T IPS Signature

A signature is comprised of many components:

Description and **ID –** These are unique to each signature
Simple Strings or Regular Expressions – These are used to match traffic patterns
Modifiers - Are used to eliminate packets (packet payload size, ports, etc.)
Meta-data – Used to selectively enable signatures that are relevant to the workload being protected using the following fields for context:

- Affected Product - Broad category of workloads vulnerable to the exploit
- Attack Target – Specific service vulnerable to this exploit (Drupal Server or Joomla, for example)
- Deployment

Performance impact – Is an optional field.
Severity – Information included in most signatures

Signatures are classified into over 50 self-explanatory categories/types including Attempted DOS, Successful user privilege gain, and shell.code-detect. Each Classification-type has a **Type Rating** (1-9) based on the risk and fidelity associated with the type of event/attack. Type ratings are mapped to NSX IPS **Severity Rating** (4 - Critical, 3 - High, 2 - Medium, and 1 - Low). Signature Severity helps security teams prioritize incidents. A Higher score indicates a higher risk associated with the intrusion event. Severity is determined based on the following:

1. Severity specified in the signature itself|
2. CVSS Score specified in the signature
   (as per CVSS 3.0 specs)
3. Type-Rating associated with the classification-type

8.3 *Profiles*

Signatures are applied to IPS rules via Profiles.

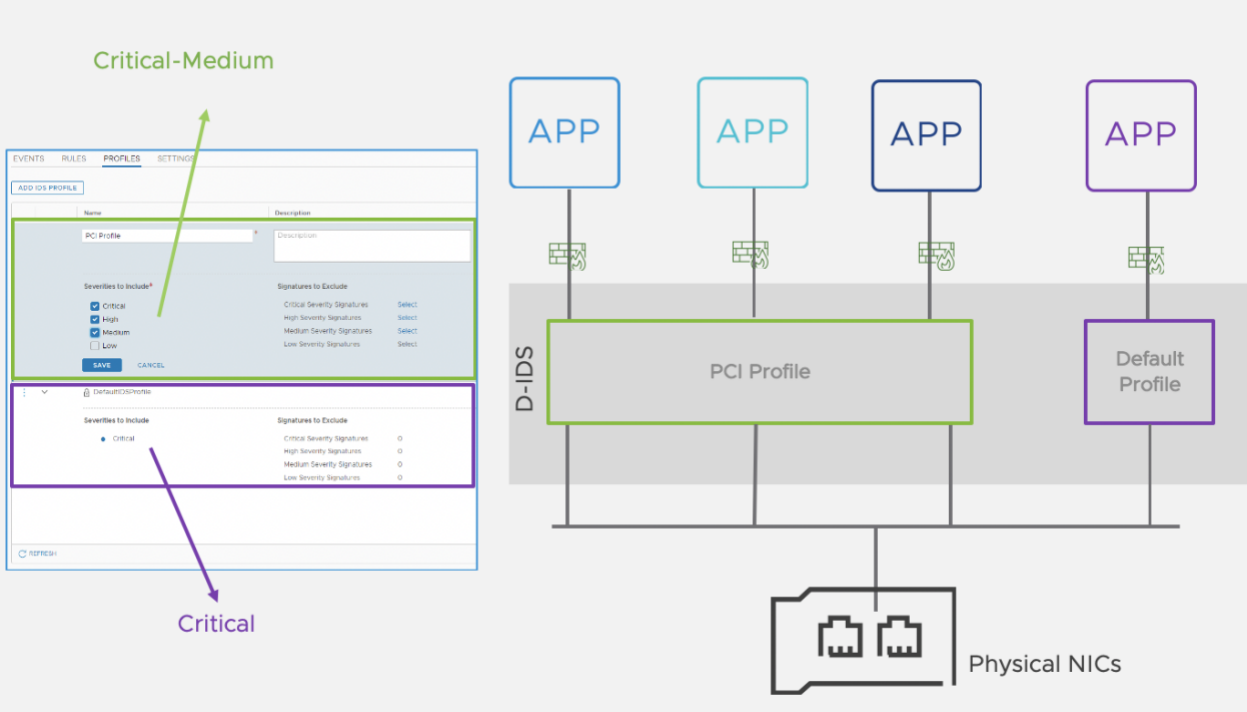

Figure 8 - 4 NSX-T IPS Signature Profile

A single profile is applied to matching traffic. The default signature-set enables all **critical** signatures. The IPS engine supports "tenants" to apply specific profiles to traffic per vNIC. This limits the number of false positives and reduces the performance impact. Profiles are used in different strategies such as a single or few broad profiles for all traffic or many granular/workload-specific profiles. The tradeoff is yours to make between administrative complexity and workload signature fidelity.

Profiles group signatures based on the following criteria:

- Classification Type
- Severity (Critical | High | Medium | Low)
- Deployment (Gateway | DC)
  Attack Target (Client | Server)
- Affected Product (Web_Browsers | Apache | …)
- Signatures can be excluded from a profile

For each profile, exclusions can be set to disable individual signatures that cause false positives, are noisy, or are just irrelevant for the protected workloads. Exclusions are set per severity level and can be filtered by Signature ID or Meta-data. The benefits of excluding signatures are reduced noise and improved performance. Excluding too many signatures comes with a risk of not detecting important threats.

8.4 *IPS Rules*

Rules are used to map an IPS profile to workloads and traffic. In other words: IPS rules define what is "interesting traffic" to be inspected by the IPS engine. By default, no rules are configured.

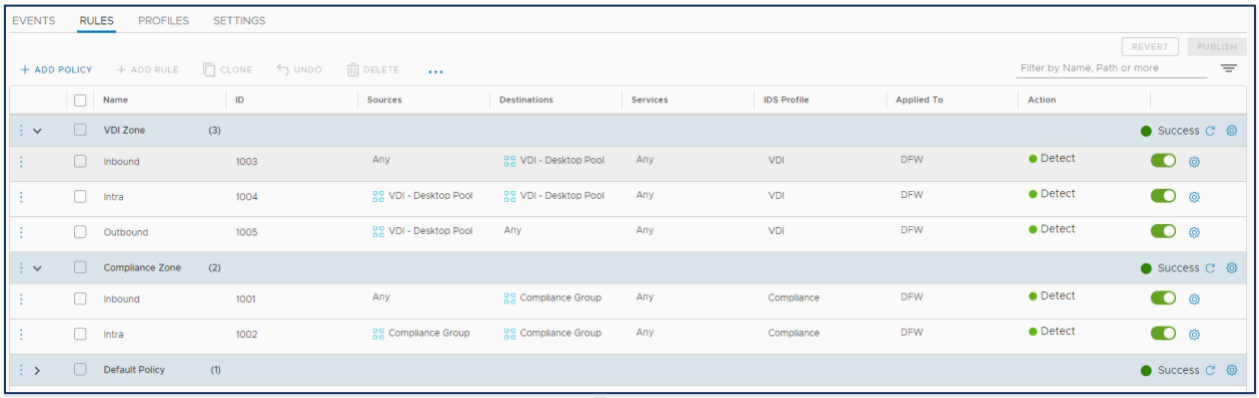

Figure 8 - 5 NSX-T IPS Rules

As one can see in Figure 8 - 5 NSX-T IPS Rules, IPS rules are similar to regular DFW rules or Service Insertion Rules. You can specify one IPS profile per rule. IPS rules are stateful and provide support for any type of group in the source and destination fields, just like DFW rules. However, the use of L7 APP-ID services inside IPS rules is not supported. As was addressed earlier with the DFW, the use of the Applied-To field to limit the scope of the rule is highly recommended.

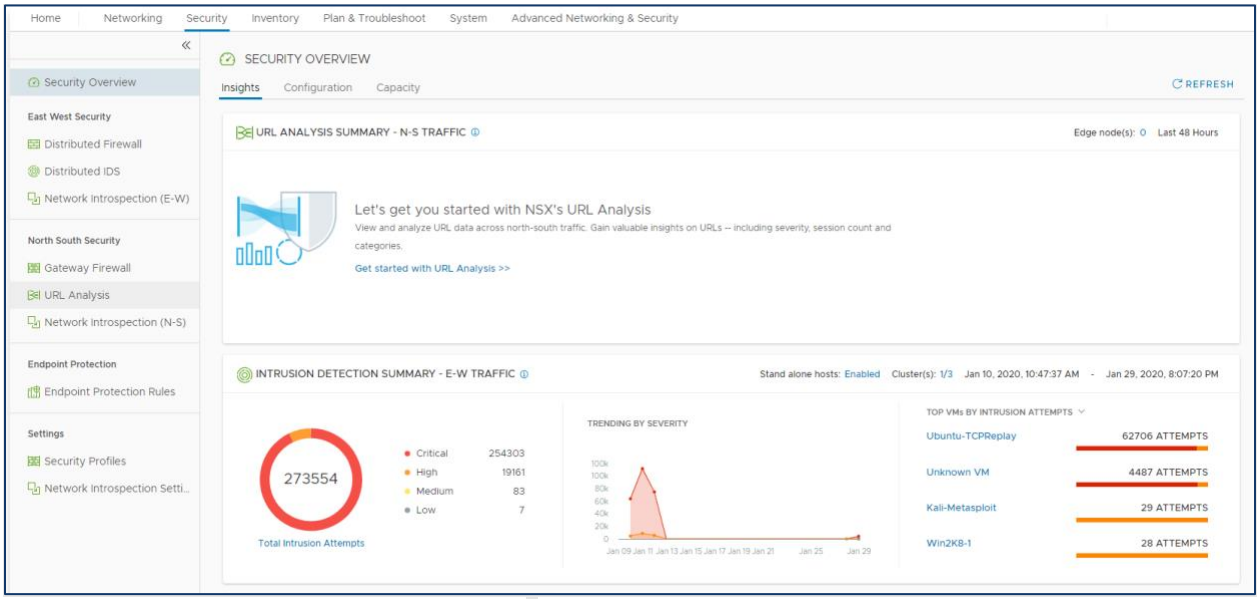

Figure 8 - 6 NSX-T IPS Insights Dashboard

The NSX Security Overview screen provides several key insights to help security teams. This screen provides three main dashboards: IPS Summary (for East West Traffic), URL Analysis (for North South Traffic), and DFW Rule Utilization.

The IPS dashboard (shown above in figure 7.6) provides the following information:

- Enabled state for standalone hosts and for clusters
  - Above shows the standalone hosts are enabled and 1 out 3 clusters are enabled.
- Date range for the data being displayed
  - Above, shows the date range is January 10, 2020 through January 29, 2020
- Total number of intrusion attempts, organized attempts by severity
  - Above shows 254303 Critical, 19161 High, and 83 Medium, and 7 Low. (If you ever see this in a live environment, brew a strong pot of coffee. It is going to be a long night!)
- Trending by Severity
  - In the figure above, it shows there was a peak on January 11th

- Top VMs by Intrusion Attempts or Top VMs by Vulnerability Severity
  - Above displays Top VMs by Intrusion Attempts

All of this information is intended to give a sense of the state of affairs in general and provide an indication of where to focus attention. If you click on the Total Intrusion Attempts, you are brought to the Events screen, shown below.

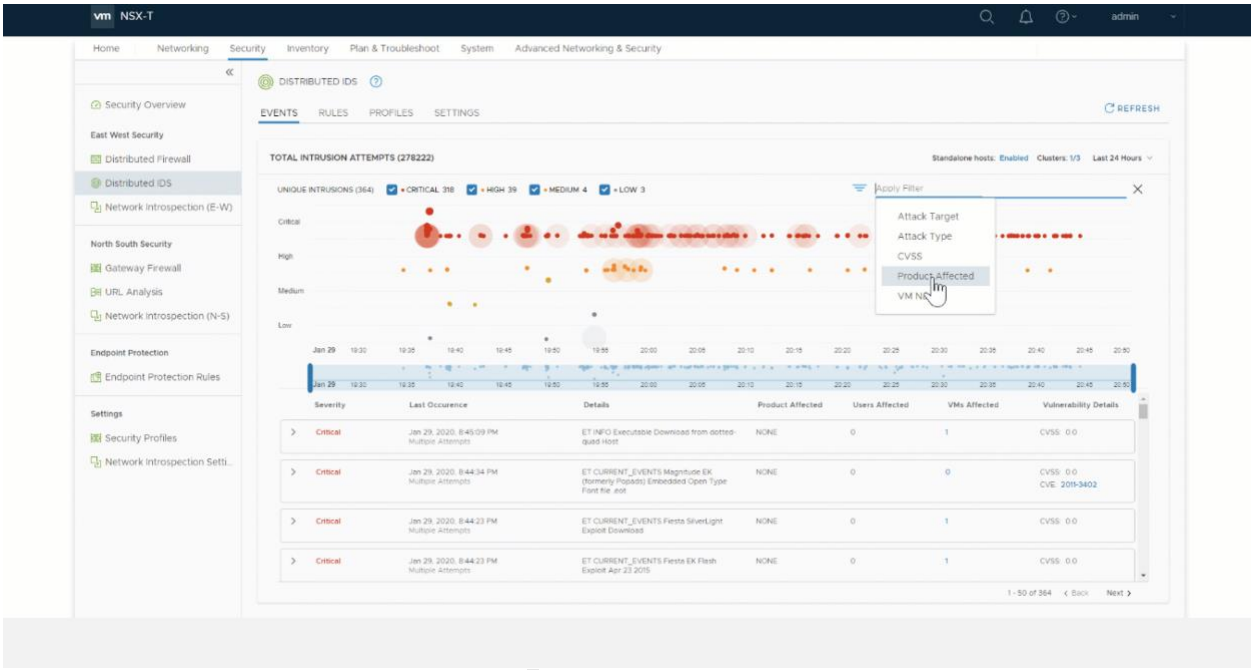

Figure 8 - 7 NSX-T IPS Centralized Events

The UI will contain last 14 days of data or 2 Million Records. There is a configurable timeframe on the far right for 24 hours, 48 hours, 7 days, or 14 days. The clickable colored dots above the timeline indicate unique types of intrusion attempts. The timeline below that can be used to zoom in or out. Finally, the event details are shown below in tabular form. On every severity level, there are check boxes to enable filtering. Event filtering can be based on:

- Attack-target (Server|Client|...)
- Attack-type (Trojan|Dos|web-attack|...)
- CVSS
- Product Affected
- VM Name

Figure 8 - 8 below shows the details of an event.

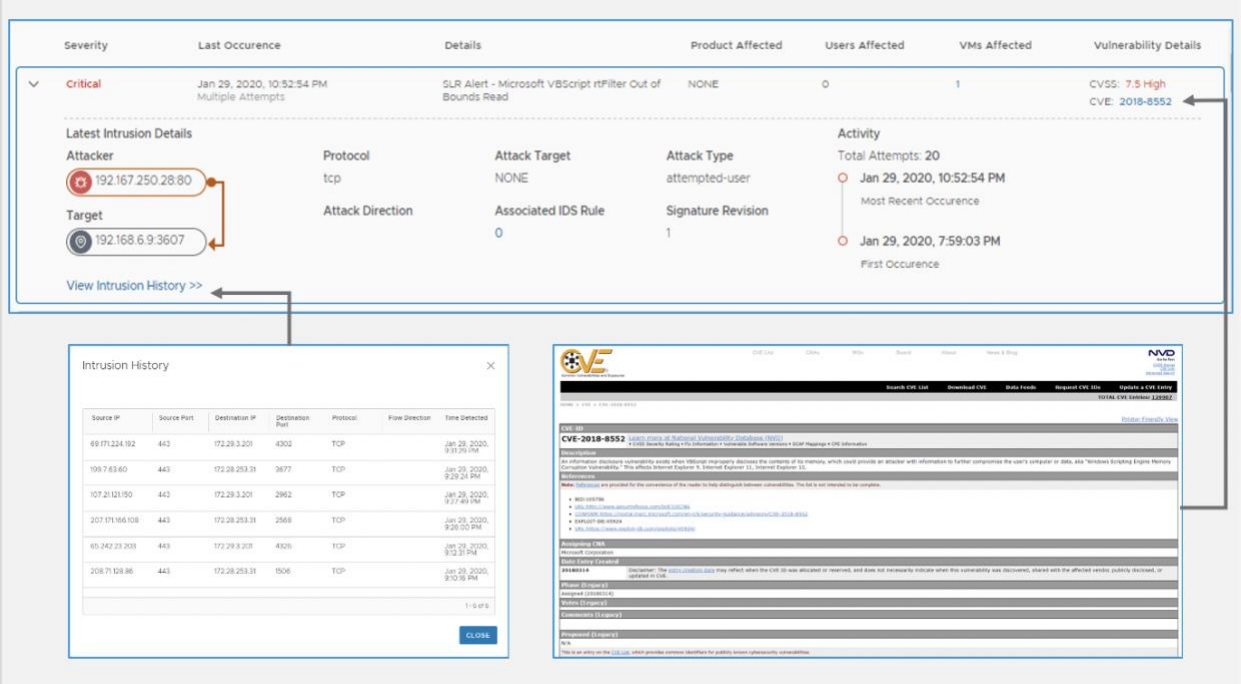

Figure 8 - 8 NSX-T IPS Event Details

Each event contains the following details:

- Severity
- Description/Details
- Attack Type
- Attack Target (if available)
- Signature Revision
- Product Affected (if available)
- Vulnerability Details
  - CVSS (Common Vulnerability Scoring System)
  - CVE ID (Common Vulnerabilities and Exposures)

Events can be stored on the host via a cli command for troubleshooting. By default, local event storage is disabled. When it is enabled the events are stored in the /var/log/nsx-idps/fast.log file.

As was defined earlier, the NSX IPS configuration high level workflow is essentially four steps: Signature download, Enabling IPS, Profile/Rule Definition, and Monitoring. Most of the time will be spent iterating between the last 2 steps after NSX distributed IPS is configured. New downloads may trigger a need to update profiles and rules, but most of the time will be spent monitoring. One important point to note with respect to IPS: Regular DFW, Layer-7 APP-ID Rules and IPS Rules can be applied to the same traffic, but traffic needs to be allowed by the DFW to be passed through IPS. In other words, IPS does not apply to dropped traffic.

8.5 *IPS Use Cases*

Although NSX IPS can be used in a wide variety of use cases, four common use cases are examined in the following section: Compliance, Zones, Appliance Replacement, and Lateral Threat Containment. Although they are highlighted as four individual use cases, it is entirely possible that they coexist.

8.5.1 IPS Use Case: Compliance

NSX IPS is typically used in compliance to enable software-based IPS/IPS for critical applications to easily achieve compliance requirements for PCI-DSS, HIPAA, SOX. Many customers need to meet regulatory compliance for their sensitive applications that deal with (for instance) healthcare or financial data such as HIPAA or PCI-DSS. These compliance requirements often specify the needs for IPS/IPS to prevent data theft. NSX enables customers to easily achieve regulatory compliance by enabling micro-segmentation to reduce the audit scope and by enabling IPS/IDS selectively on the workloads that need to meet compliance. Certain regulatory requirements specify the needs for Intrusion Detection to be enabled for all applications subject to those regulations. Without NSX IPS, that would require all traffic be funneled through a group of appliances, which could have an impact on data center architecture. With the combination of NSX DFW and NSX IPS, traffic can be microsegmented and tagged for IPS as showing in figure 7.10 below.

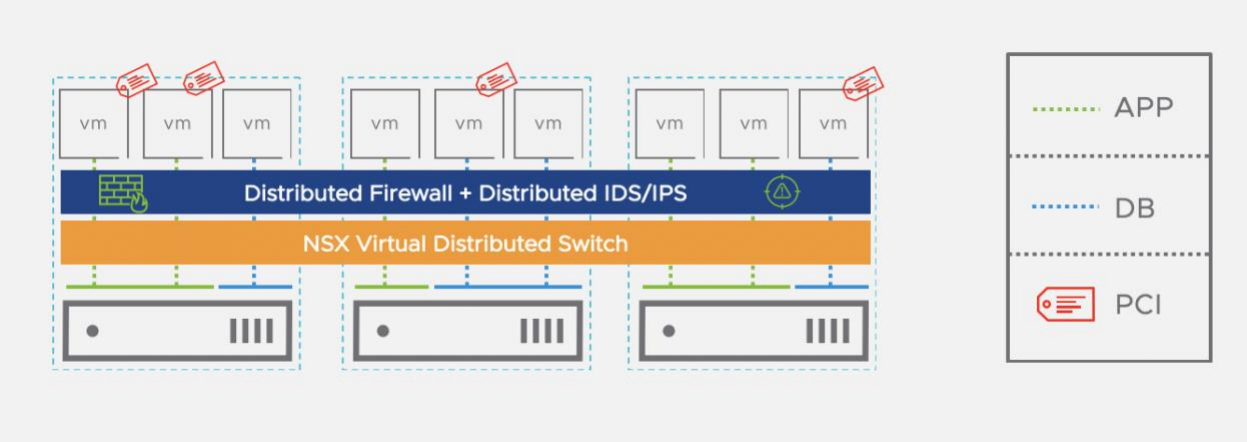

Figure 8 - 9 NSX-T IPS Compliance

In the example above, the PCI application is tagged so that it is firewalled off from the other applications which are coresident on the server hardware. IPS can be applied to only that application to meet compliance requirements, without requiring dedicated hardware. If desired IPS with a reduced signature set may be applied to only the database portion of the other applications, for example.

This use case highlights the following aspects of NSX IPS:

- Reduced compliance scope
- Selective enablement of IPS throughout the environment
- Apply signatures relevant to compliance zone
- Reducing performance impact and alert noise

NSX IPS allows customers to ensure and prove compliance, regardless of where the workloads reside which enables further consolidation of workloads with different compliance requirements on x86.

8.5.2 IPS Use Case: Creating Zones

NSX IPS allows customers to create Zones in software without cost and complexity of air-gapped networks or physical separation. Some customers provide centralized infrastructure services to different lines of business or need to provide external partner with access to some applications and

data. All customers need to provide proper segmentation between DMZ workloads that are exposed to the outside/guest wifi and the internal applications and data. Traditionally, this segmentation between tenants or between the DMZ and the rest of the environment was done by physically separating the infrastructure, meaning workloads and data for different tenants or different zones were hosted on different servers, each with their own dedicated firewalls. This leads to sub-optimal use of hardware resources. The NSX Distributed firewall and Distributed IPS/IPS allow customers to run workloads that belong to different tenants and different zones on the same hypervisor clusters and provide the same level of segmentation they would get with physical firewalls and IPS appliances while allowing much higher consolidation ratios.

8.5.3 IPS Use Case: Appliance Replacement

With the added functionality of NSX distributed IPS, many customers evolve from legacy appliance-based IPS architectures to NSX distributed IPS. As customers are virtualizing their data center infrastructure and networking, NSX enables them to replace physical security appliances with intrinsic security that is built into the hypervisor. Doing this for both firewalling with the distributed firewall and for IPS with the Distributed IPS/IPS provides a single security policy for both across the whole SDDC. Further, there is a real savings in terms of rack space and electricity and cooling with the intrinsic approach. Each data center grade security appliance draws on the order of 10 kW of power, which is almost 90,000 kW per year – per appliance! When those appliances are replaced by an intrinsic security architecture which uses the spare cycles of each CPU in the datacenter, the savings add up quickly. Savings, which come with an ENHANCED security posture.

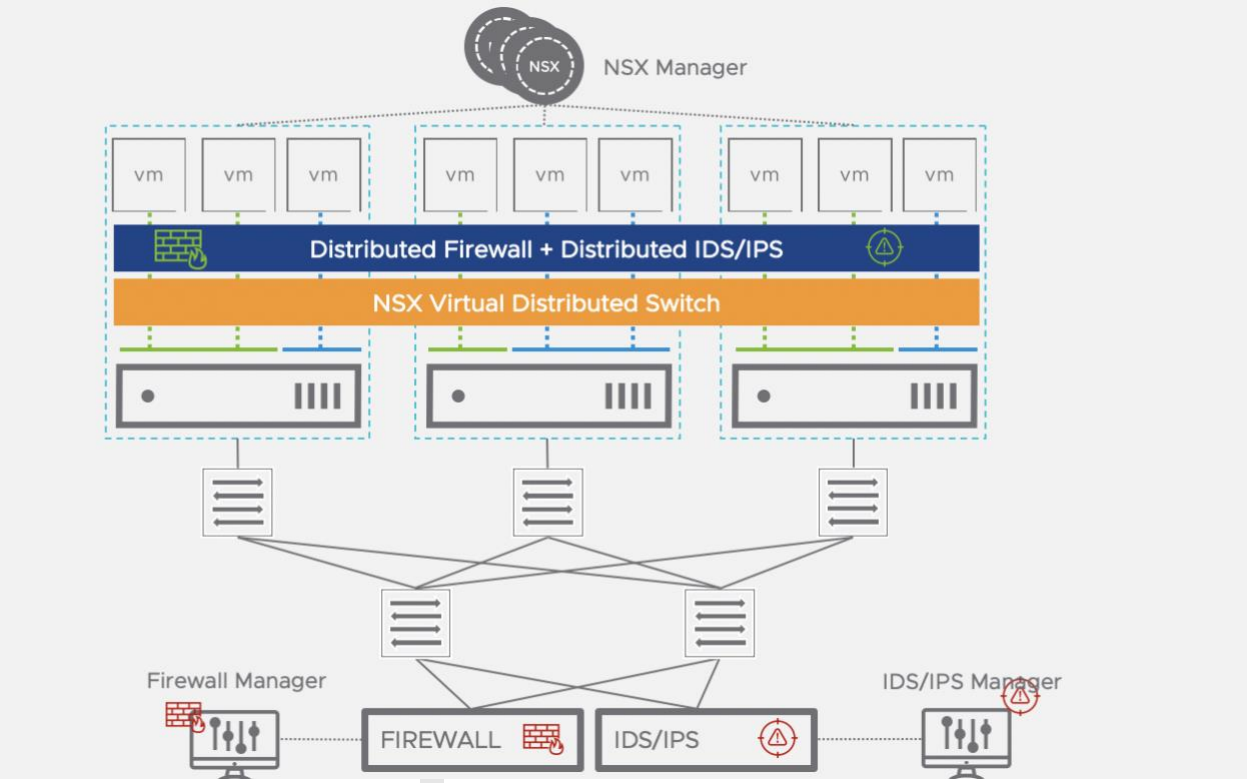

Figure 8 - 10 NSX-T NSX distributed IPS appliance replacement

Beyond the security appliance savings, there is a significant savings in networking infrastructure required to connect things up as shown in Figure 8 - 10. This use case alone can fund the change to an intrinsic security architecture.

8.5.4 IPS Use Case: Detecting Lateral Threats

NSX IPS allows customers to combine signature-based detection, anomaly detection, and protocol conformance checks. Almost invariably, the actual objective of an attack is not the same as where the attacker initially gained access. This means that an attacker will try to move through the environment in order to get to steal the valuable data they are after. Hence, being able to not just defend against the initial attack vector, but also against lateral movement is critical. Micro-segmentation using the distributed firewall is key in reducing the attack surface and makes lateral movement a lot more difficult. Now, for the first time, micro-segmentation becomes operationally feasible to front-end each of your workloads with an Intrusion Detection and Prevention service to detect and block attempts at exploiting vulnerabilities wherever they may exist. This protection exists regardless of whether the attacker is trying to gain initial access in the environment, or has already compromised a workload on the same VLAN and is now trying to move laterally to their target database on that same VLAN.

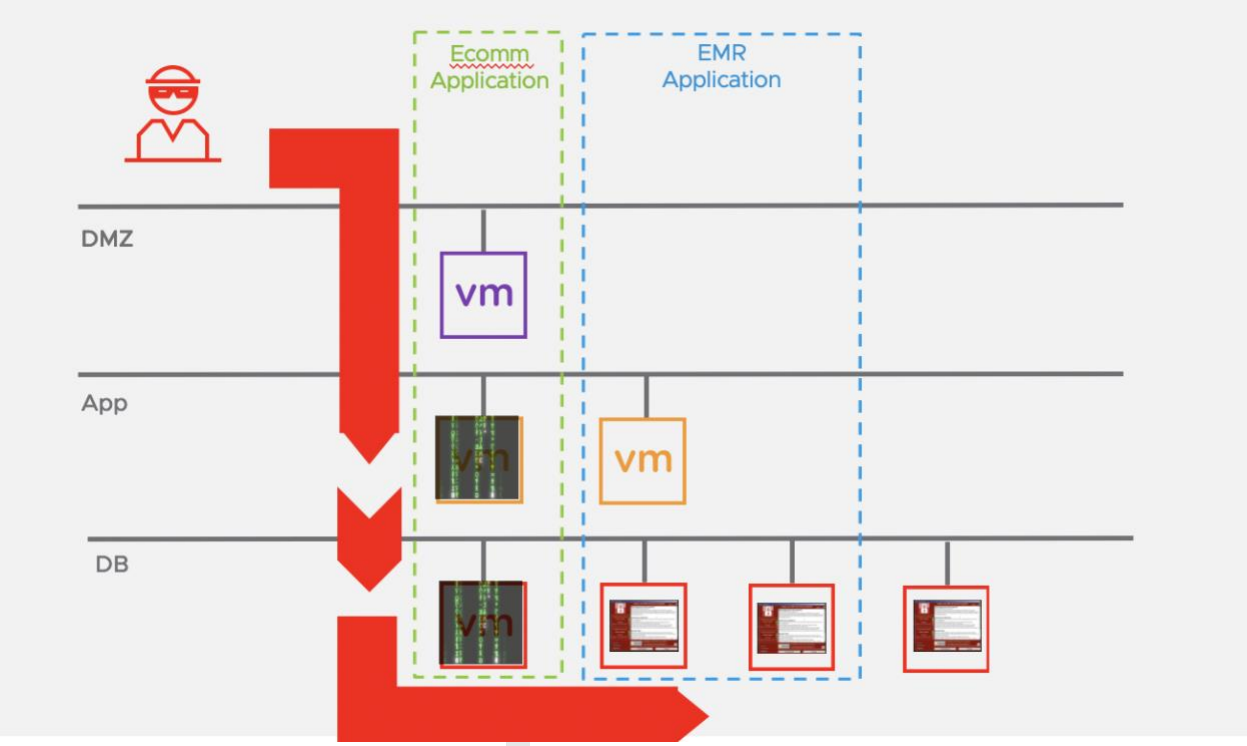

Figure 8 - 11 NSX-T Lateral Threat Movement

Distributed IPS front-ending every workload enables exploit-detection regardless of it being initial attack vector, lateral spread or exfiltration.

9 Federation

Most enterprise environments have multiple data centers for scale and/or disaster recovery, each with its own compute resources as well as its own network and security resources.

For simpler solutions, NSX-T offers multi-site. This solution NSX-T Multisite solution is based on 1 NSX-T Manager Cluster managing Transport Nodes (Hypervisors + Edge Nodes) physically in multiple sites. In case of high scale need, a second NSX-T Manager Cluster has to be installed. With multi-site, however, each NSX-T Manager Cluster is independent, and Network and Security objects are not shared between them. The sweet spot for multi-site is 2 locations in a metro region for DR. For truly diverse data centers, multi-site does not suffice; federation is designed to address this use case.

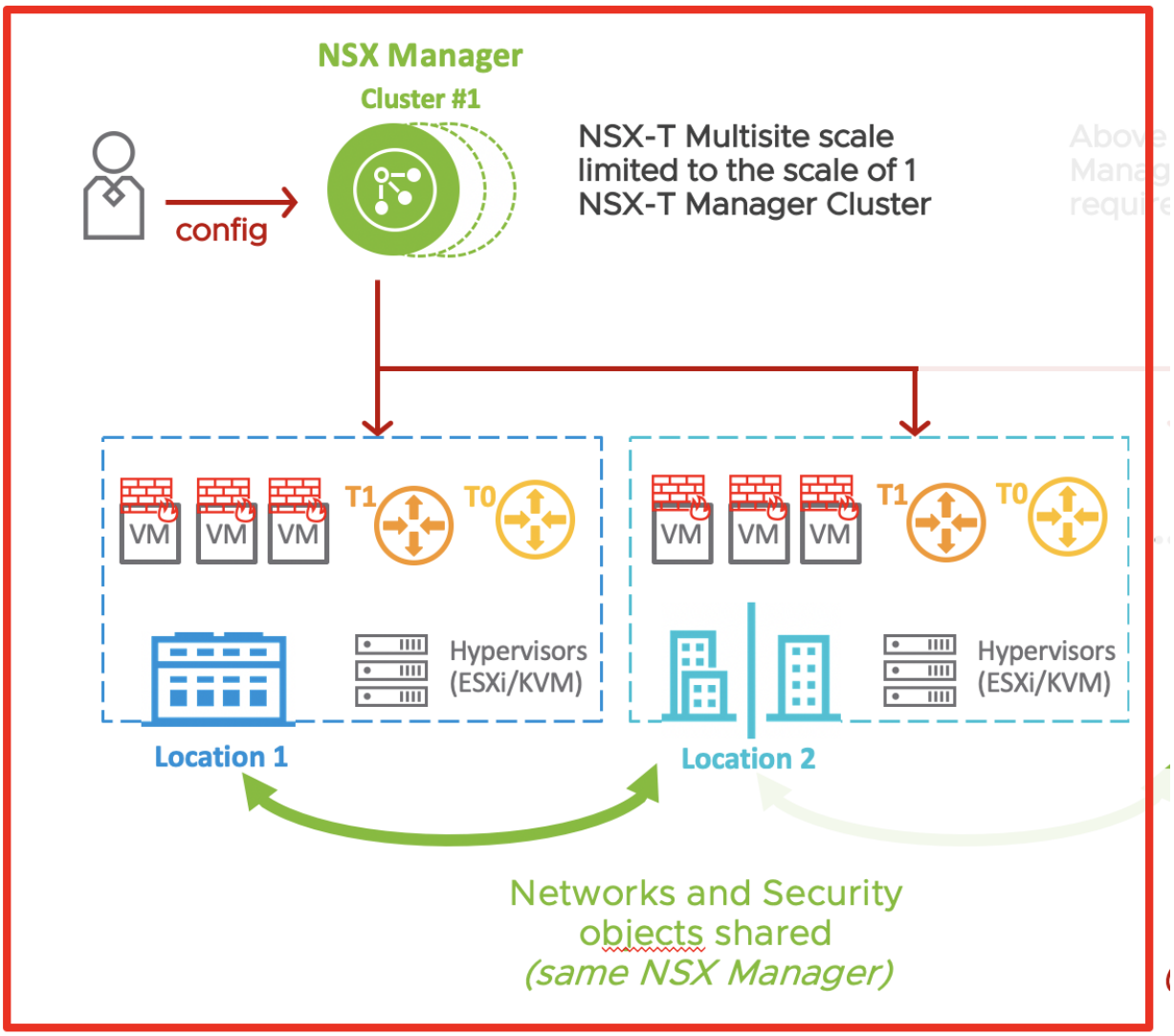

Figure 9 - 1 NSX-T MultiSite

With NSX-T Federation, you have Network and Security services offered by NSX Local Managers (LMs). Local Managers are the very same NSX-T Managers you know. Here they are called Local Managers to differentiate them from the new NSX element with NSX-T Federation: Global Manager

(GM). Global Manager offers Operational Simplicity with Network and Security configuration centrally done to the GM, and then transparently pushed to all LMs. And also offers Consistent Policy Configuration and Enforcement with network and security objects that are shared across LMs. So, you can create Global rules like “all my DMZ Web Servers can talk to my Active Directory Servers” and this single rule will be pushed and enforced to all the data centers. (Note: All manager connectivity – GM to LM and LM to LM- must not be NATed. Connectivity to the Edge Nodes must also not be NATed.)

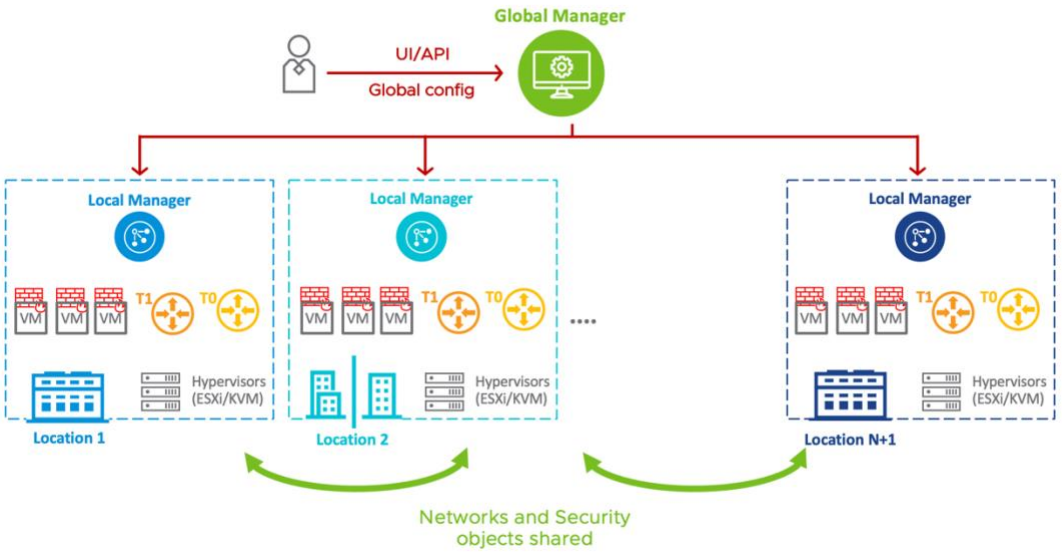

Figure 9 - 2 NSX-T Federation

9.1 *Managers in Federation*

Each manager logically depicted above represents a manager cluster of three appliances. Although the GM is represented as one object, it is functionally one active GM cluster with a standby GM cluster in another location as shown in figure 8.3 below.

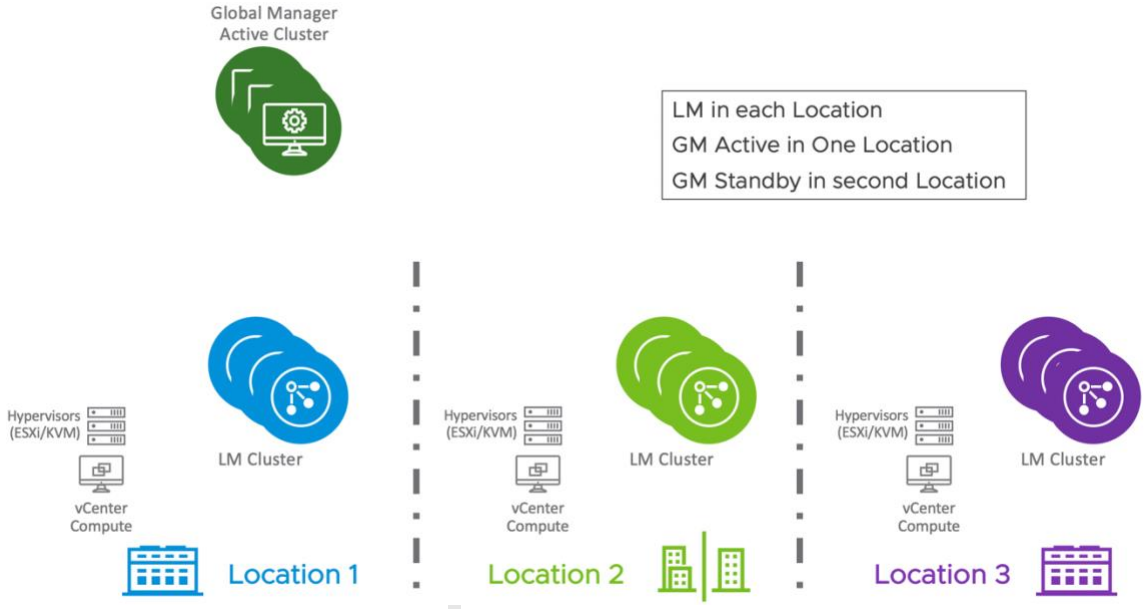

Figure 9 - 3 NSX-T Federation Clusters

The active GM cluster stores the configuration, syncs it to the standby GM, and pushes it to the relevant LM(s). If, for example a segment is stretched between Locations 1 and 2, but not 3, the config would only be pushed to the LMs at Location 1 and 2, but not the LM at 3. The control plane state is synced between the peer LMs. So, for example, group membership which spans sites is synced between the 2 LMs directly using the "async_replicator" process whose status is available via the cli.

The UI provides a means for selecting the Location for configuration as shown in Figure 9 - 4 below.

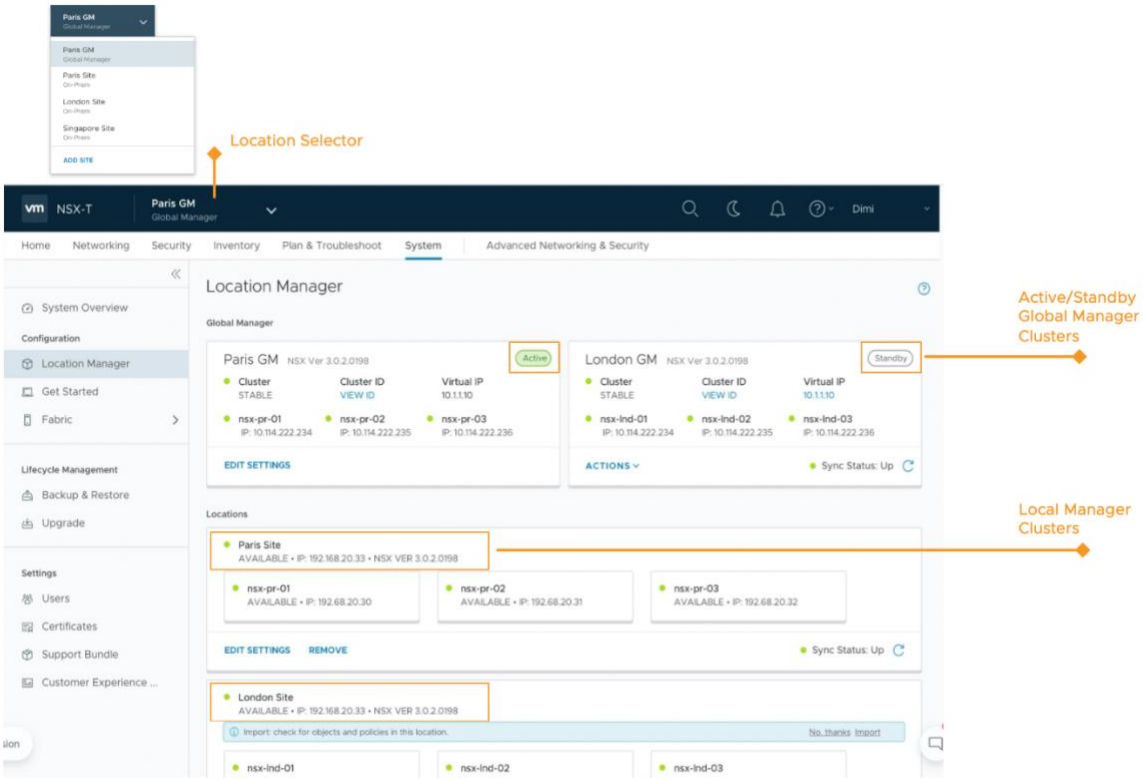

Figure 9 - 4 NSX-T Federation UI

As mentioned above, when interacting with the GM the configuration is pushed to the LM(s). However, the LM configuration remains local. It is not pushed up to the GM. This interaction is shown in Figure 9 - 5

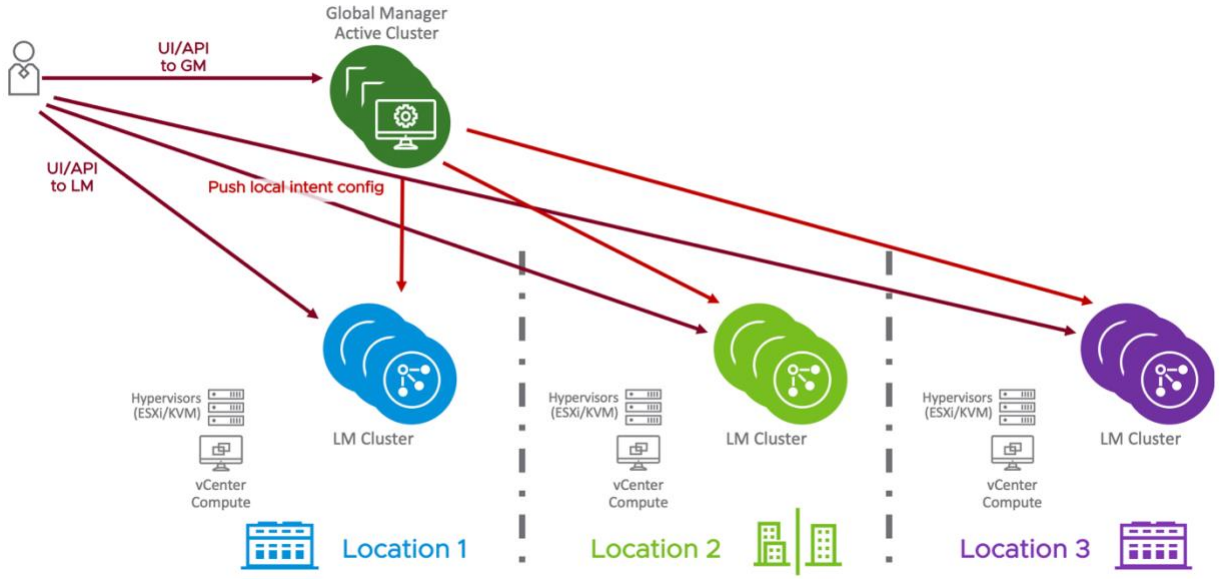

Figure 9 - 5 NSX-T Federation Config Push

9.2 *Groups*

As explained earlier, groups are a very efficient tool for policy configuration on NSX-T firewalling. Groups are also available with Federation, but now there are 3 different types of groups: Global, Regional, and Local. Global groups are relevant at all locations. Regional groups are relevant at more than one location, but not all locations. Finally, local groups are relevant at only one location. It is important to note that groups can mix spans as shown below in Figure 9 - 6.

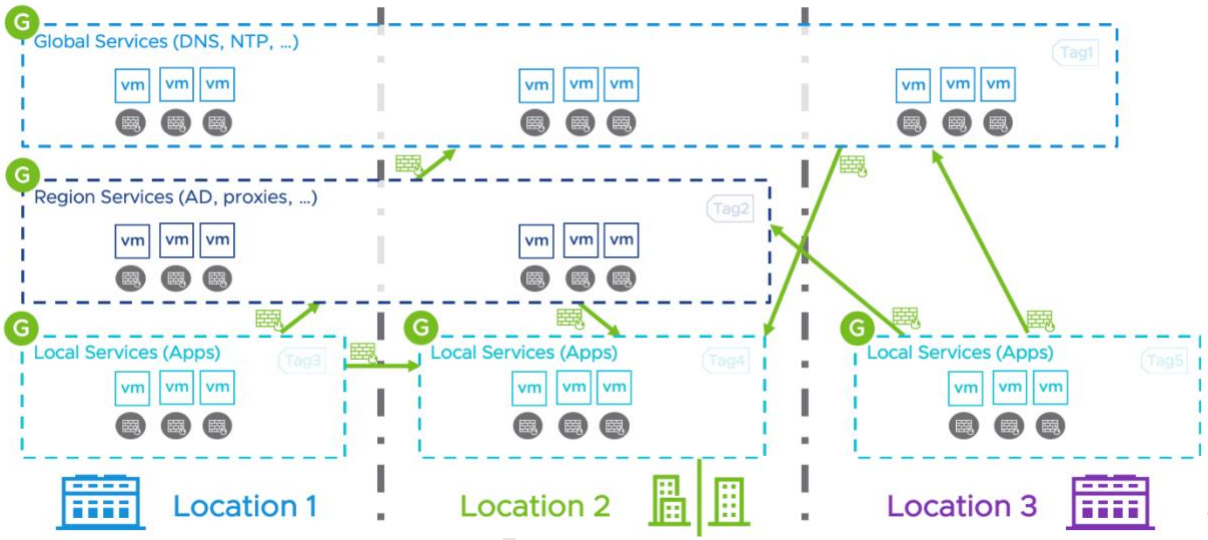

Figure 9 - 6 NSX-T Federation Groups

Figure 9 - 6 shows a global group that contains global services such as DNS and NTP. There is also a regional group which contains AD and proxy services. Finally, there are local App groups. Note that the Apps in Location 3 consume the Regional services as well as the global services and thus require firewall rules allowing this.

As mentioned above, group membership is updated directly from LM to LM. Groups can be defined by tags so membership may be quite dynamic.

The following figure shows a topology with stretched T1 services.

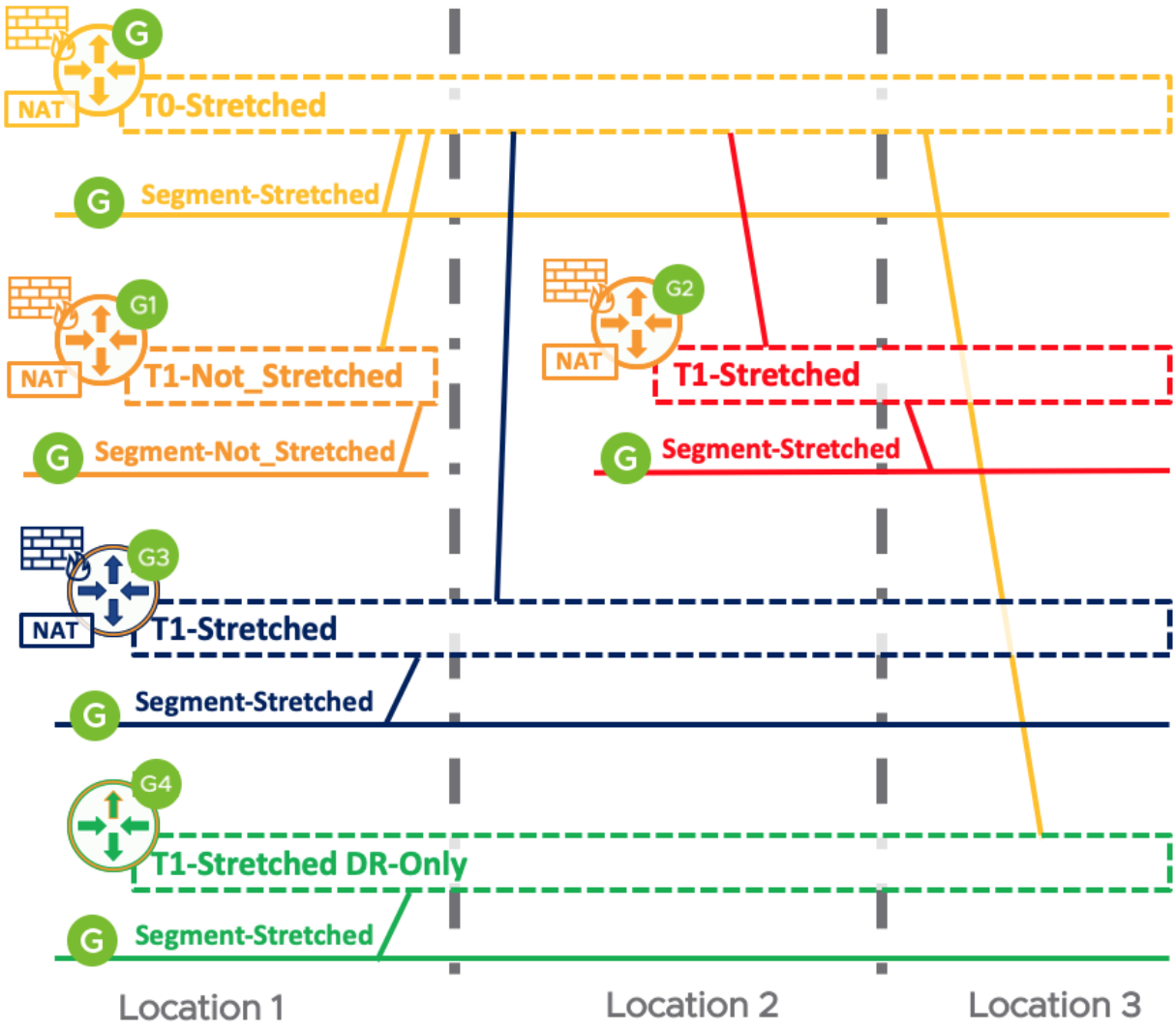

Figure 9 - 7 Stretched T1 services

Here, there are 4 groups, of which three (G1, G2, and G3) are used in NAT rules. G1 is a local group to Location 1. G2 is a regional group spanned across Locations 2 and 3. G3 and G4 are global groups, meaning they span all three locations. The span of a T1 is (by definition) equal to or a subset of the T0 to which it is connected.

For a complete discussion of Federation use cases and configuration, please see the Federation Design Document.

10 Management and Operations

One of the challenges of using legacy firewalling tools for securing modern infrastructure is that (due to their architectural nature) they lack the tools needed to effectively secure and manage a datacenter infrastructure. As mentioned in the introduction, these legacy firewalls are designed to be at a perimeter with an inside and an outside – a safe side and a suspicious side. East West Firewalling has no such bearings. East West Firewalling is about securing everything. One of the greatest challenges that customers face in implementing East West Firewalling is in defining policy for an infrastructure which has been around for years or even decades. How do you secure an environment which you don't know or understand? This is where modern policy management tools come in. VMware offers 2 such tools: vRNI and NSX Intelligence. Each tool has its own use cases and sweet spots. This chapter examines those.

This chapter closes with a look at operations. A detailed list of the tasks required for a successful NSX implementation is provided.

10.1 *vRealize Network Insight (vRNI)*

vRNI is the perfect tool to understand an environment where NSX does not exist. vRNI uses netflow/IPFIX to understand traffic patterns. It has visibility to the virtual and physical world by tapping the switches and routers in both worlds. vRNI provides an understanding not only of what is talking to what on which ports, but also a sense of the volume of that traffic flow.

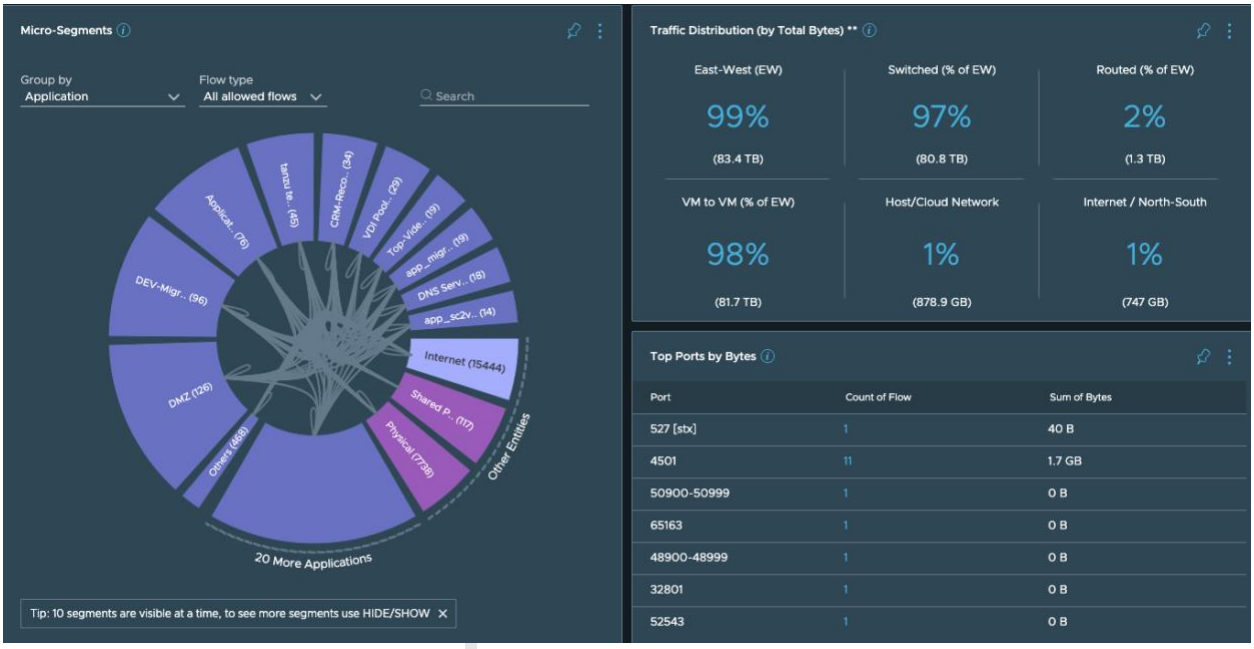

Figure 10 - 1 vRNI Micro-segmentation Planning Window

vRNI is a great tool to assess the rough order of magnitude of the undertaking in question. vRNI has been used by customers to:

- Determine application interdependencies

- Determine application volume, and use
- NSX compliance and policy suggestions
- Troubleshoot day two issues

For NSX compliance and policy suggestions, vRNI can determine flows that are unprotected by NSX even if both endpoints are unprotected. One of the views in the Security Planning section is unprotected flows. When those flows are displayed, the entirety of the suggested security policy can be exported in YAML or XML. Or, one can click on one wedge/application to see a suggested security policy (exportable to XML or CSV) for only that wedge. This policy can easily be ingested into NSX using a simple python script such as the one found on the vRNI github repository (https://github.com/vrealize-network-insight/vrni-rule-import-vmc-nsxt). **Error! Reference source not found.** shows both those screens.

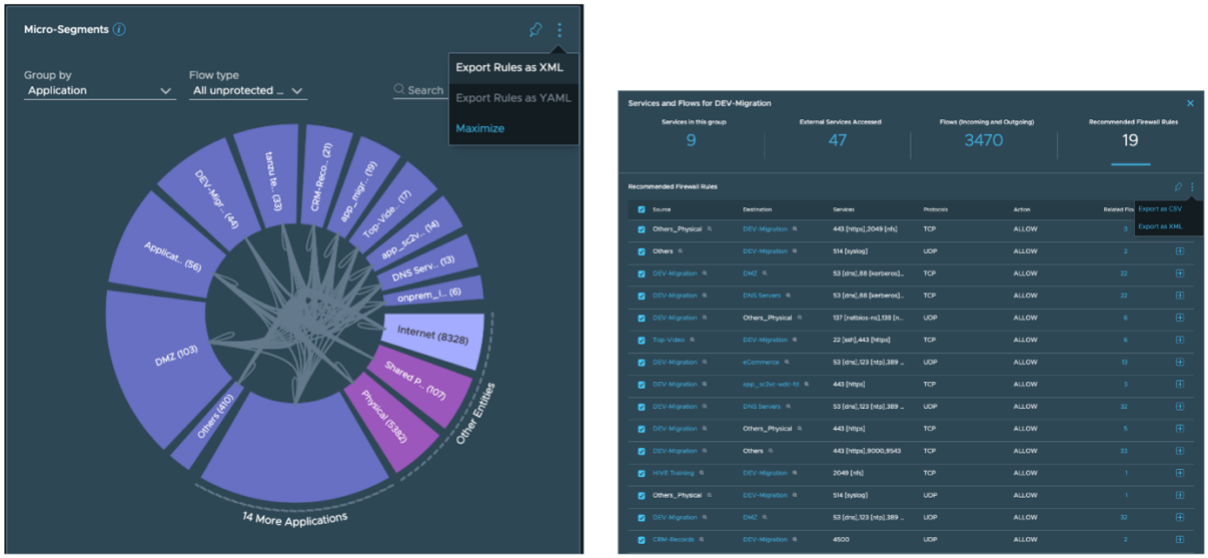

Figure 10 - 2vRNI Unprotected Flows and Recommended Firewall Rules

Troubleshooting day two issues is where vRNI excels the most. vRNI makes the entire infrastructure searchable. If one dos not know where 2 endpoints are, a vRNI query for the path between them will plot them out, with intermediate switches, routers, firewalls, and load balancers all depicted, even if either one or both endpoints are containers or in public clouds (vRNI integrates into AWS VPCs and Azure VNETS when provided credentials) – even if one or both endpoints are containers in public clouds. Figure 10.3 shows a flow which traverses 2 data centers and an AWS VPC. What is important to note is that a user can query the path without knowing where the 2 endpoints with equal simplicity as if the endpoints were two VMs sitting next to each other on a host.

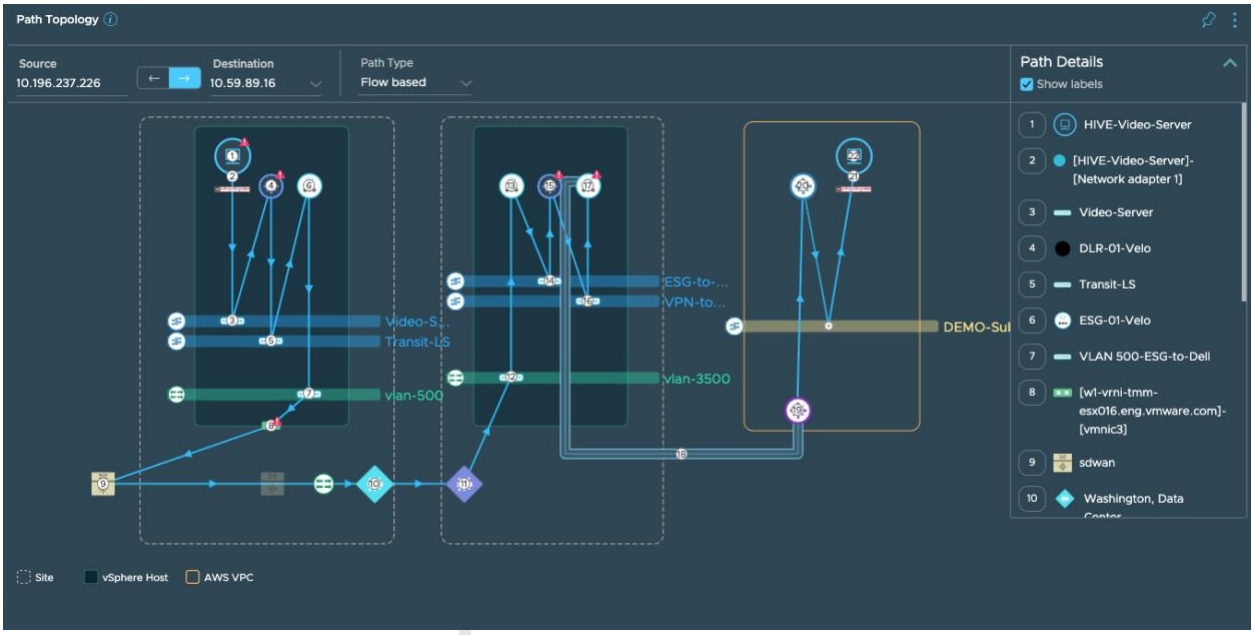

Figure 10 - 3 vRNI Path Tool 2 data centers and AWS VPC

vRNI also provides native integration into every major firewall vendor management platform. This allows vRNI to provide a visual end to end path which includes firewalls along the way and relevant security policy. This is show in Figure 10 - 4 below.

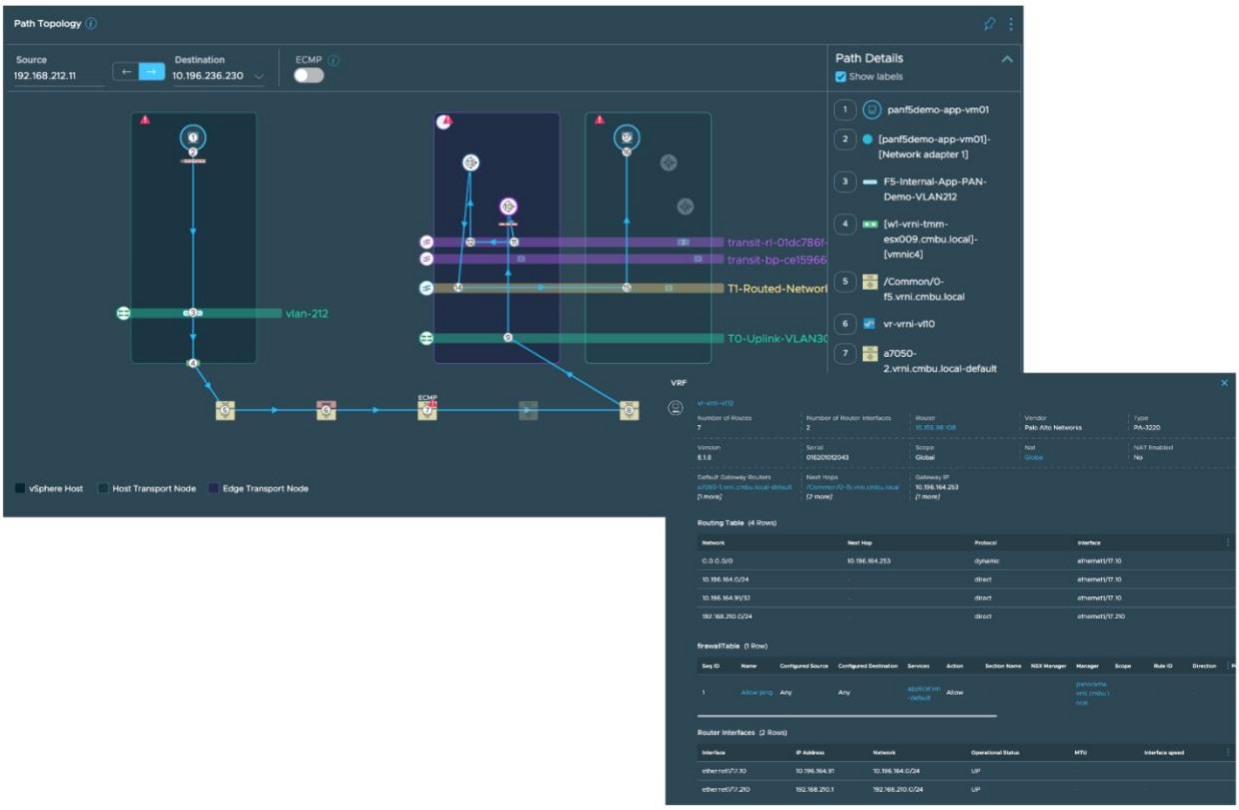

Figure 10 - 4 vRNI Path Tool with Palo Alto Networks Physical Firewall

As can be seen from the figure, not only is the firewall noted in the path, but one can click on the object and view the portion of the firewall policy relevant to these endpoints through vRNI integration with Panorama. When NSX is deployed, vRNI can help with compliance by pointing out unprotected

flows. It can also help alert on factors that may affect the health of the NSX infrastructure components such as VM storage issues.

10.2 *NSX Intelligence*

When NSX is installed, NSX Intelligence is the optimal tool for visualization and policy planning, closing the speed and action gap with network and host informed analytics. NSX Intelligence is a lightweight central appliance with distributed processing engines inline within the hypervisors which take a single pass approach to provide intelligent policy formulation, as well as security and network analytics. Because NSX Intelligence processing engines lie within the hypervisors, they can increase in processing capacity linearly with the increase in compute.

NSX Intelligence has the luxury of complete L7 inspection and endpoint context for every workload. This is combined with bi-directional intelligence feeds from external sources. When NSX is installed, NSX Intelligence is the optimal tool to:

- Automate Micro-segmentation/Firewalling at Scale
- Demonstrate and Maintain Policy Compliance
- Simplify Security Incident Troubleshooting

When used to automate firewalling at scale, NSX Intelligence provides a repository for policy management and enforcement. NSX Intelligence will generate new recommendations upon detecting changes to policy. This allows you to create a baseline recommendation, then let NSX Intelligence learn the desired DFW policy. As part of optimal policy design, NSX Intelligence can discover groups of up to 250 members based on VM membership changes. In providing automated firewall policy recommendation which can be pushed directly to the NSX firewall, NSX Intelligence speeds up the securing of complex, unknown east west environments. NSX Intelligence also provides an iterative workflow with continuous updates to topology visualization.

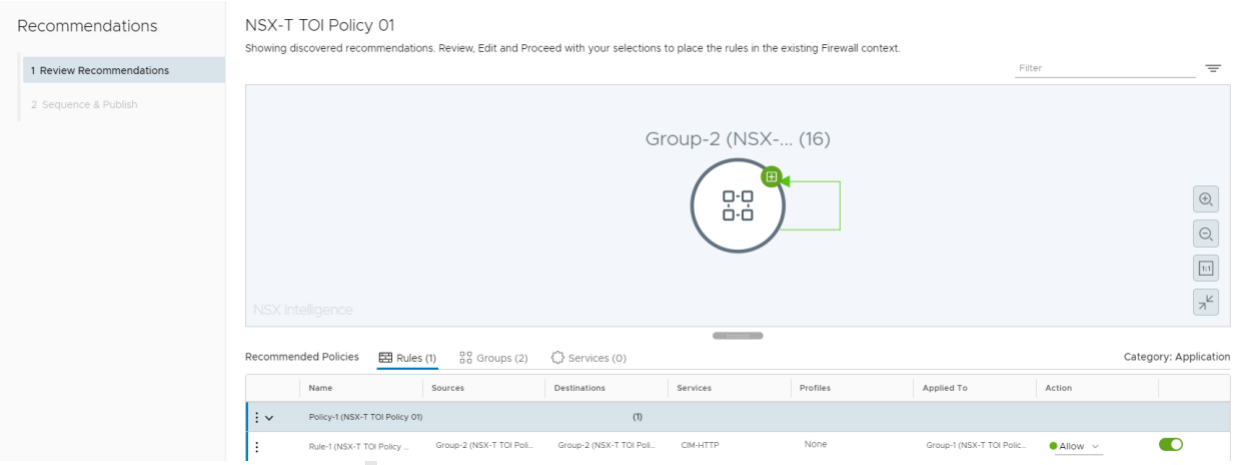

Figure 10 - 5 NSX Intelligence Policy Recommendation, Viewed in DFW table

For compliance, NSX Intelligence provides a complete record of every flow from, from every workload. NSX Intelligence also provides correlated flows and policies to highlight misconfigurations, policy exemptions, and on-compliant flows between workloads of security scopes. Most importantly, NSX intelligence provides continuous analysis so that the above information is always accurate and current.

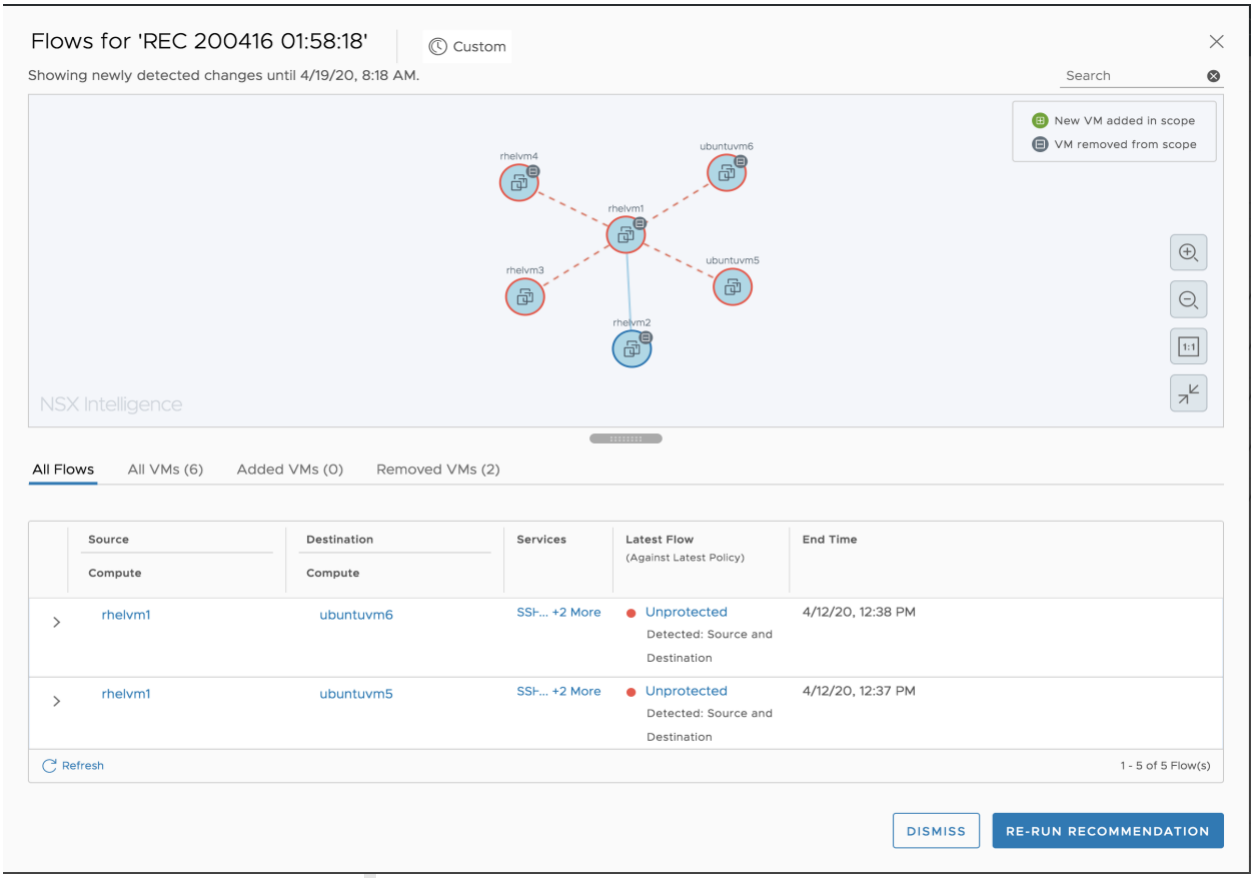

Figure 10 - 6 NSX Intelligence New Recommendation Upon Detected Changes

For troubleshooting, NSX Intelligence provides comprehensive visibility of the NSX environment for security teams. Notably, NSX Intelligence provides Layer 7 analysis of every flow, without sampling, for optimal fidelity. The Drill-down topology visualization combines application maps and complete workload inventory. By default, community grouping is activated when the UI detects more than 1,000 nodes. Multi level visualization allows NSX Intelligence to scale to enterprise environments. This is shown in figure Figure 10 - 7 below.

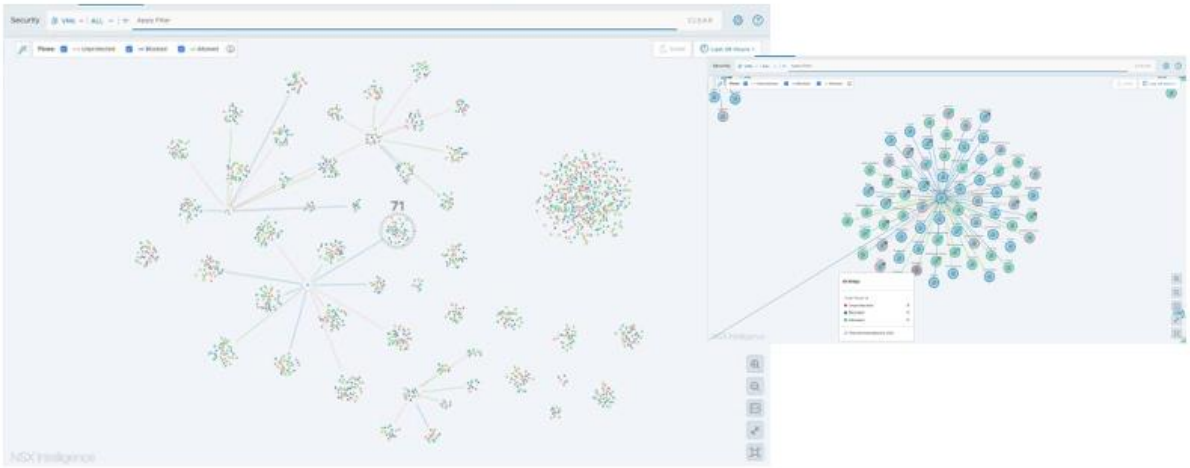

Figure 10 - 7 NSX Intelligence

To summarize, vRNI and NSX Intelligence are two complementary tools which coordinate for a complete security management solution. vRNI is the perfect tool for understanding the scope of an environment without NSX. Once NSX is installed, the simplified rule recommendations and deployment mean one-click firewalling. For day two operations, vRNI assists in the micro-seg planning by app modeling and grouping, leveraging information from sources such as Service Now. For end-to-end infrastructure visibility across both the physical and virtual environments, nothing beats vRNI.

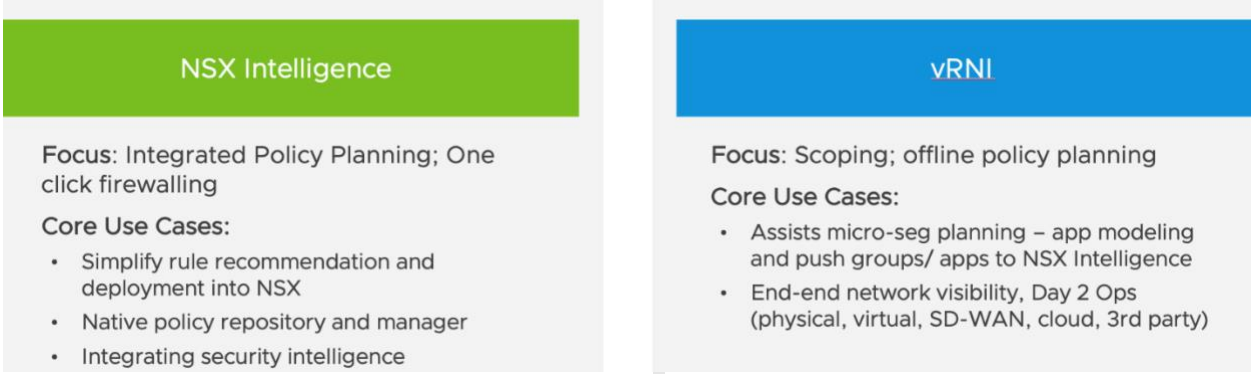

Figure 10 - 8 NSX Intelligence vs vRNI

10.3 *SIEM*

System Information and Event Management tools (aka SIEM or syslog tools) is an important part of any security approach for early detection of attacks and breaches. SIEM tools collect and aggregate data from a variety of sources (devices, endpoints, applications, and even services). In addition to writing RFC 5424 compliant syslog messages to a local file (in the /var/log/ directory), NSX can configure a remote syslogging server via the cli (with the *set logging-server* command). Syslog is supported on the NSX Manager, the NSX Edges, and the hypervisors. On hypervisors, the *tac*, *tail*, *grep*, and *more* commands can be used to navigate the logs. The audit log is part of syslog.

NSX includes a license for vRealize Log Insight. Log Insight (as its lovingly known) commonly is used to front end larger SIEM installations such as Splunk to reduce the cost burden of the latter. In doing so, many customers also find the NSX-T content pack for LI to provide significant value. Instructions for pointing NSX-T audit and syslogs to LI can be found in the VVD here.

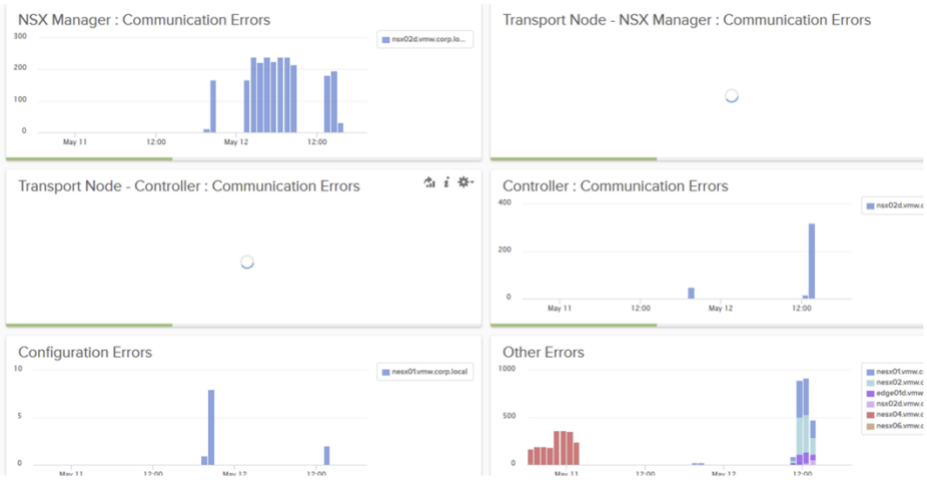

Figure 10 - 9 NSX Content Pack for vRealize Log Insight

For those who prefer to ingest the NSX syslog data directly into Splunk, there is an NSX-T App for Splunk.

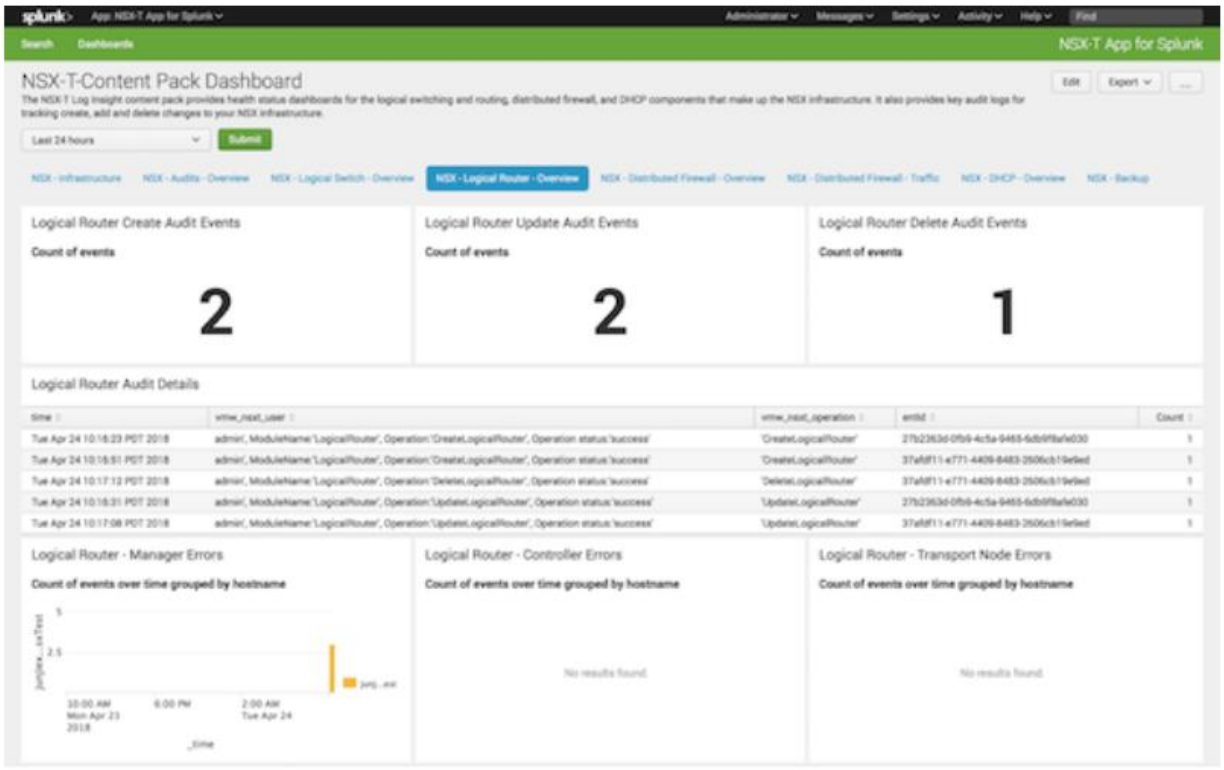

Figure 10 - 10 NSX Content Pack for Splunk

10.4 *NSX Operations*

One of the most frequently asked questions from customers is “How do I operationalize this?” This question is similar to the question of “How do I run a household?” In both cases, while there is no strict list of exactly how to do things, there is a list of tasks which need to be addressed for a success. Much like who takes the garbage out will vary house to house, a household cannot succeed without that task being done. Similarly, which person assumes the given NSX task will vary from company to

company, based on the culture at each company. But for a successful implementation, the listed tasks need to be addressed. The following table describes some of the high-level tasks that need to be done to operationalize NSX:

Task	Responsible Role	Comments	Role in NSX
Architecture			
Design and publish detailed NSX designs and drawings	Cloud Security Architect	In collaboration with Network Architect.	Planning activity
Application assessment and migration strategy	Cloud Security Architect	Migration of applications from existing physical firewall to logical firewall	Depends on Approach
Define common services for applications	Cloud Security Architect	Services for modern and traditional applications. IT services for applications eg: DNS,NTP,DHCP,AD. Admin need to understand where the Services reside and how to track changes.	Security Admin
Define Security Group model for DFW	Cloud Security Architect	Working with Engineers, and collaboration with Application teams.	Security Admin
Obtain overall approval from Security on architecture	Cloud Security Architect		Planning activity
Define support staff authorization policy and list (who/what) for NSX Manager (Admin / User Roles)	Cloud Security Architect	Working with Engineering and Cloud Operations leadership	Enterprise Admin
Define, signoff, and publish NSX Security object naming and tagging conventions	Cloud Security Architect		Security Admin
Design blueprint security tagging policy	Cloud Security Architect		Security Admin
Define policy and ports for application access (e.g., Web, App, DB)	Cloud Security Architect		Security Admin
Security approval process for new services (e.g., FW policies)	Cloud Security Architect	Working with Engineers. For example, health monitoring.	Security Admin

Task	Responsible Role	Comments	Role in NSX
Architecture			
Specify posture for Security Zone	Cloud Security Architect, Cloud Network Architect	Build the security framework for Test and Development zone, Production zone, DMZ etc.	Security Admin
Guide engineering and operations teams with implementation and onboarding	Cloud Architect, Cloud Admin		Not Applicable to a specific role in NSX
Identify, evaluate, and recommend automation and operations tools	Cloud Architect		Not Applicable to a specific role in NSX
Define alerting and notification model	Cloud Architect		NSX Admin
Auditing and reporting processes for compliance	Cloud Architect	Define the processes for audit and reporting for compliance	Not Applicable to a specific role in NSX
Engage in Tier 3 support as needed	Cloud Security Architect	Advanced troubleshooting and architectural changes	Not Applicable to a specific role in NSX
Planning for security in the physical network	Cloud Security Architect	In collaboration with Infrastructure Security Team. For example, inter-rack connectivity and communication with NSX appliances.	Not Applicable to a specific role in NSX
Engineering			
Building the automation and orchestration model	Cloud Tooling Engineer	Development of blueprints, templates for automation. Eg: vRealize suite, Openstack,Puppet,Chef etc.	Not Applicable to a specific role in NSX
Deploy & test defined blueprints	Cloud Security Engineer		Not Applicable to a specific role in NSX

Task	Responsible Role	Comments	Role in NSX
Engineering			
Deploy Operations Tools for Monitoring and Troubleshooting	Cloud Tooling Engineer	vRNI Dashboards, NSX Dashbaords, Runbooks. VI admin requires to configure syslog configuration for Hosts	Security Admin
Build, manage, and maintain NSX Infrastructure	Cloud Infrastructure Engineer	Deploy test, validate and certify the infrastructure. Capabilities, configurations, integrations and interoperability. Ensure fulfilment of requirements (capacity, availability, security and compliance), ensure backup and restore of NSX Manager data. Upgrade and patch infrastructure and tools.	NSX Admin
Build, manage, maintain and customize provisioning, monitoring and troubleshooting tools	Cloud Tooling Engineer		NSX Admin
Build all Common Services for applications	Cloud Security Engineer	Working with Architects, build the services for applications like firewall services	Security Admin
Modify policy/ports on ongoing basis	Cloud Security Engineer	Check to see if this can be handed over to Operations team. Implement routine, approved and exception changes.	Security Admin
Implement microsegementation security model	Cloud Security Engineer	Security groups, tags, policies, service insertion. Using NSX Intelligence.	Security Admin
Implementing logging for security events based on Architecture	Cloud Security Engineer	Cloud Security Architect will also be involved. Via vRNI, Log Insight, Splunk	NSX Admin

Task	Responsible Role	Comments	Role in NSX
Engineering			
Implement alerts and notifications	Cloud Tooling Engineer	Implement alerts and notifications for events in monitoring systems.	NSX Admin
Implement Access Control to NSX infrastructure components	Cloud Security Engineer	Working with Architects	Enterprise Admin
Build security for the physical network (e.g., physical firewall rules)	Cloud Security Architect	In collaboration with Infrastructure Security Team. For example, inter-rack connectivity and communication with NSX appliances.	Not Applicable to a specific role in NSX
Tier 2 support	Cloud Security Engineer	Diagnose and analyze root cause of issues. Apply patches and fixes as needed.	n/a
Operations			
NOC and SOC staff manage NSX operations	Operations Director	Working with Engineers	Auditor
Deploy Application topologies based on blueprints/templates	Operations Engineer	vRA catalog to deploy network topologies and instances	Automated already by engineering
Tier 1 support for infrastructure and security	Operations Engineer	Document tickets, respond to alerts and alarms, basic break-fix tasks, document alerts/alarm messages, track tickets to closure, and escalate to Tier 2 as needed.	Not Applicable to a specific role in NSX
Respond to exception/failureissues on build/run automation	Operations Engineer	Working with Engineering.	Not Applicable to a specific role in NSX
Monitoring, alerting, and troubleshooting NSX and physical security infrastructure	Operations Engineer	Infrastructure, applications and security. Responding to alerts and notifications. vRNI.	Follow the runbooks from eng to perform the tasks. No specific NSX role required.

Appendix

Below is listing of VMware Security Products and features across the heterogeneous infrastructure which is common today. Infrastructure today extends along a continuum from physical servers on prem to VMs in hypervisors (sometimes a variety of hypervisors like ESXi and KVM) to containers, on prem and in the cloud, to Software as a Service (SaaS) offerings like Office365 (O365) and SalesForce (SFDC). VMware offers the tools to secure this heterogeneous environment in a consistent manner, while allowing the qualities of each solution to shine.

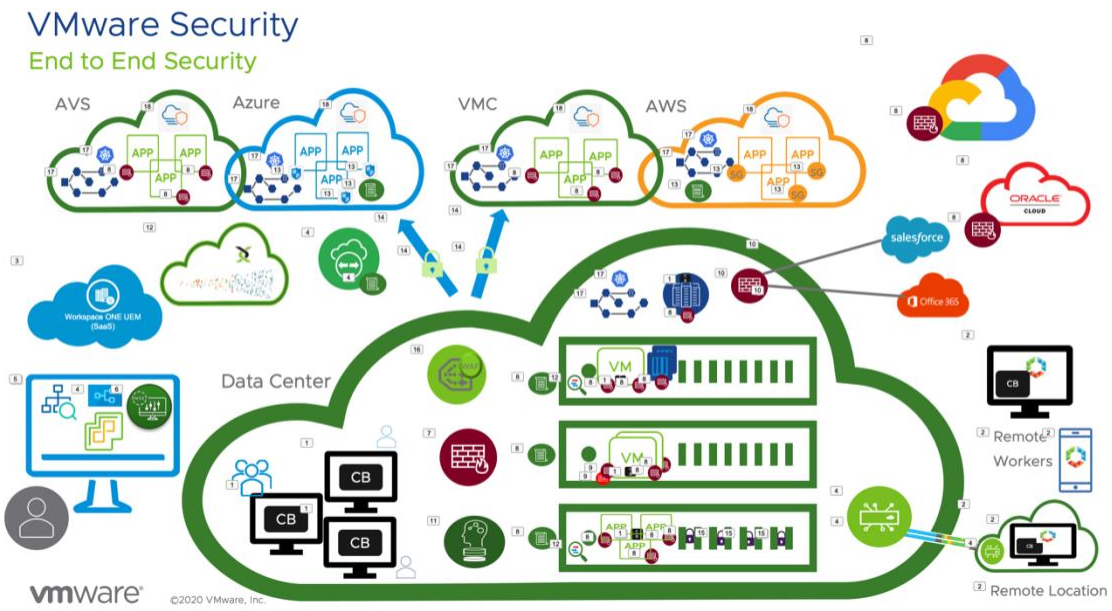

Figure 1- 1 VMware Security Offering

1 – VMware Carbon Black – CB allows customers to detect and stop threats with endpoint and workload security.
2- VMware Horizon View - Horizon allows for the secure delivery of virtual desktop infrastructure.
3 – WorkspaceONE – WorkspaceONE is a Unified Endpoint Manager (UEM) which provides a single point of definition and control of the intersection of application/user/device/location.
4 – VMware SD-WAN by VeloCloud – Traffic to remote locations can be secured (and optimized through DMPO – Dynamic MultiPath Optimization) using SD-WAN by VeloCloud. Now, with Secure Access Service Edge (SASE) functionality, the admin can also define secure connectivity policy.
5– vRealize Network Insight - vRNI provides visibility into the physical underlying infrastructure of switches and routers as well as the virtual infrastructure through netflow, or into the legacy firewall infrastructure through integration with a variety of firewall managers. This visibility is complemented by a cross sectional view of the virtual infrastructure from native Amazon Web Services (AWS) and Microsoft Azure environments to branches to ESXi VMs and Kubernetes (K8) containers. This ubiquitous view combines into a complete picture of the environment which is searchable. In addition, an admin can get firewall policy suggestions or just determine the path with applicable security policies along every step from point A to point B.

6 – NSX-T Data Center – This document will focus on the security features of NSX. To provide context in the greater picture, items 7 through 14 provide a listing of the security components of NSX.
7 –NSX Gateway Firewall – NSX Gateway Firewall secures the data center boundary. It also provides security at the physical to virtual boundary as well as tenant boundaries, in multi-tenant environments.
8 – NSX Distributed Firewall - For East-West security, the admin can centrally define policy from the NSX Manager. NSX leverages a distributed local control plane to implement policy definition using local constructs (be they firewall rules on every virtual NIC (vnic) of a VM or agents running on physical servers). NSX Distributed Firewall runs on ESXi or KVM hypervisors, on prem or in several clouds. It also runs as part of the NSX Container Plug-in (NCP) which supports K8, RedHat OpenShift, and Tanzu container platforms.
9 – NSX Identity Firewall – NSX IDFW uses Active Directory User SIDs to provide user-context for single-user Horizon/Citrix VDI and server OS cases, and server OS use cases, as well as multi-user, RDSH use cases such as Horizon Apps and Citrix Published Applications/Virtual Apps.
10 – NSX URL Filtering – NSX also provides URL filtering capabilities, whether it is to ensure that malicious websites are not being accessed (such as by ransomware for Command and Control) or by users misguided sense of where to download software.
11 – NSX Intelligence – NSX Intelligence is a native distributed analytics platform, that leverages workload and network context from NSX, to deliver converged security policy management, analytics, and compliance.
12 – NSX Advanced Threat Prevention (ATP) – From the Lastline acquisition, ATP delivers network traffic analysis and advanced malware analysis with comprehensive network detection and response capabilities.
13 – NSX IPS - For intrusion detection, NSX brings industry first distributed IPS (Intrusion Detection and Prevention System). This not only provides distributed, scalable IPS but also prevents misfires through unparalleled context.
14 - NSX Cloud – For AWS and Azure native workloads, NSX Cloud offers a single point of policy control across VPCs and VNETs to ensure policy consistency. For AWS and Azure native environments, security can be implemented either via agents on workloads or natively via cloud controls
15 – IPSec VPN – To access cloud environments (such as for direct connect) or anywhere else, NSX ensures the in flight traffic is encrypted using IPSec VPN.
16 – vSAN Disk Encryption –For data at rest, vSAN disk encryption ensures data is safe.
17– Web Application Firewall – NSX provides integrated load balancing. With our Advanced LB, comes iWAF: intelligent WAF that uses analytics and machine learning to tune policy and insights into attack traffic.
18 - Tanzu Service Mesh – For the security of microservice applications across K8 clusters and clouds, VMware provides Tanzu's service mesh.
19 – Secure State - Finally, VMware Secure State correlates risk across this dynamic cloud infrastructure, reporting on risk such as "any any allow" configuration changes.

This vast offering of products and features allows for pervasive and granular security policy definition from endpoints to servers to containers to microservices. It also allows for encrypting data both in flight and at rest. Finally, this also allows for the detection of suspicious behaviors on endpoints or in the network across a heterogeneous environment.

Intrinsic Security is Security by Design