WHITEPAPER

VMWARE® NSX® MICRO-SEGMENTATION CYBERSECURITY BENCHMARK

A MICRO-AUDIT OF NSX THREAT MITAGATION EFFECTIVENESS

COALFIRE RESEARCH AND OPINION
FINAL VERSION 1.0

vmware®

COALFIRE℠

North America | Latin America | EMEA
877.224.8077 | info@coalfire.com | coalfire.com

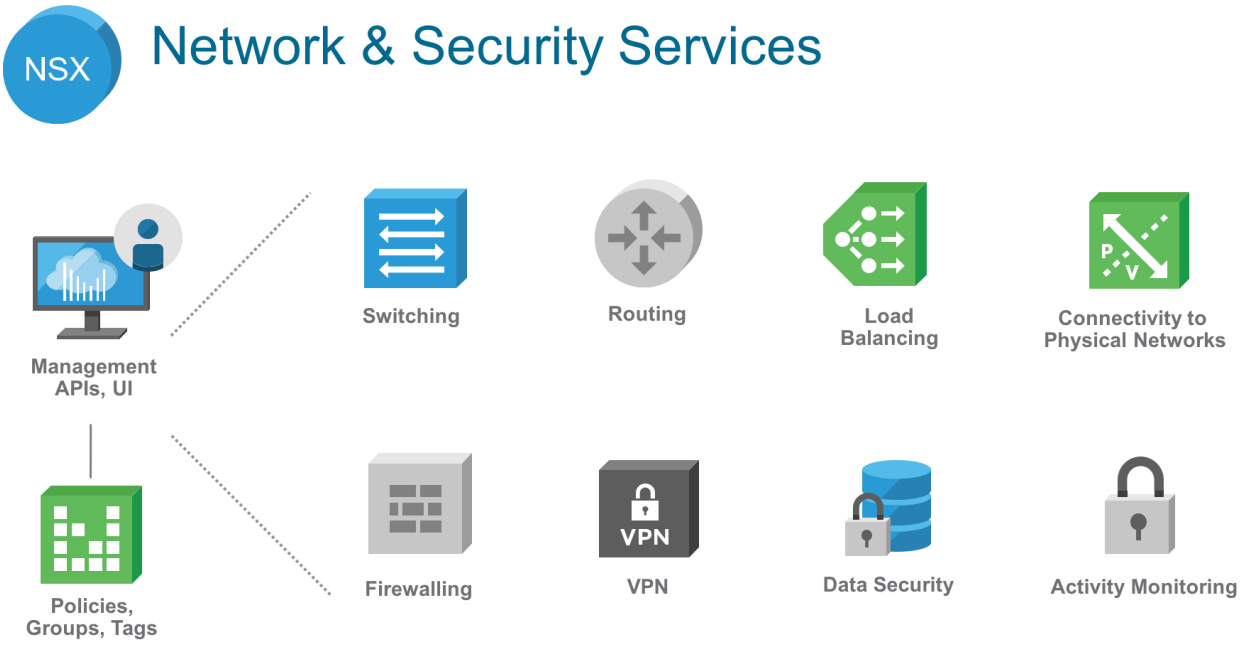
TABLE OF CONTENTS

INTRODUCING MICRO-SEGMENTATION AND 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

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, VLAN and physical network bridging components – all constructed to satisfy the requirement to protect every flow inside the software-defined data center and to facilitate initial segmentation requirements all the way to a true Zero Trust model. Intrinsic tools such as central 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.



The key that allows NSX to make a least privilege/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 five following 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 active directory groups, logical switches, VMs, port groups IPsets, etc.). Each application can now have its own security perimeter without relying on VLANs. See the *DFW Policy Rules Whitepaper* for more information.

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 and guest introspection capabilities.

NIST SPECIAL PUBLICATION 800-125B RECOMMENDATIONS

Emerging cybersecurity standards, such as those being developed by the National Institute of Standards and Technology (NIST – the US federal technology agency responsible for applied standards for technology and measurement), are contributing to an emerging global consensus on Information Security, particularly with regard to 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

VMware NSX-based micro-segmentation purports to meet all four of these recommendations. Coalfire Systems' testing of the NSX product during this "micro-audit" intends to examine the form and function of NSX to determine the following:

- Does VMware NSX functionally satisfy NIST SP 800-125B recommendations VM-FW-R1, VM-FW-R2, VM-FW-R3 and VM-FW-R4?
- Are the precepts of micro-segmentation, as defined in the complete definition, satisfied conceptually and in testing by NSX?
- Can real-world threats be stopped by NSX in E-W (peer transits on the L2 network) and N-S (network to network transits via L3), using industry-standard Penetration Testing tools?

Based on the determination of these three objectives, Coalfire will also render an opinion on the potential suitability of the VMware NSX product to deliver effective security controls to real-world legacy and emerging virtualized software-defined data centers.

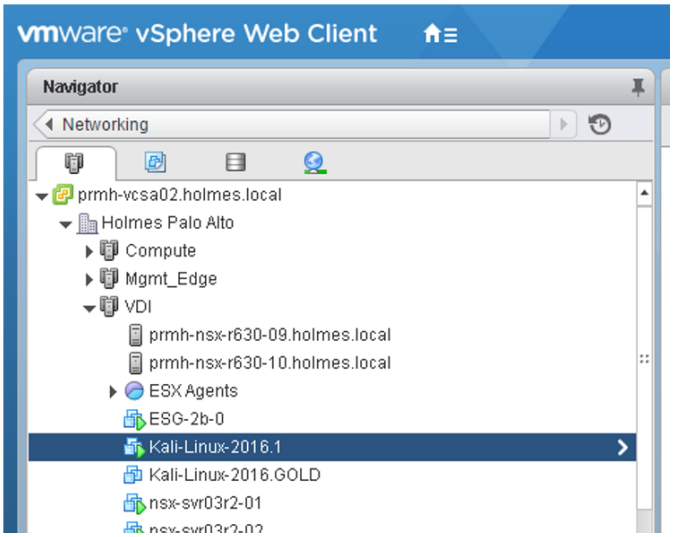
OVERVIEW OF OUR NSX "MICRO-AUDIT" ON EFFECTIVENESS IN THREAT MITIGATION

Coalfire developed the following "micro-audit" methodology to perform testing on NSX to determine answers to the three objectives: 1) NIST SP800-125B recommendations fulfillment; 2) Satisfaction of the architectural definition of micro-segmentation; and, most importantly 3) Real-world threat mitigation to east-west (E-W) and north-south (N-S) threat propagation. In our final Coalfire Opinion section, we used the outputs of our "micro-audit" to answer the effectiveness question.

First a short disclaimer: The "micro-audit" is not intended to be a specific regulatory compliance audit against the specifics of a particular regulation (PCI-DSS, HIPAA, SOC, FedRAMP, CJIS, etc.), but is intended to use Penetration Testing (aka "exploit") methodologies to perform actual network transactions and to validate the response of NSX acting on those transactions to mitigate the security threat. By no means are we performing exhaustive "Hail Mary" testing with these exploits, nor are we conducting a comprehensive survey of all threat types.

Our methodology uses a series of five (5) specific network configurations or "patterns", which aim to be representative of likely real-world network scenarios, which are found in typical customer VMware implementations.

The test environment consists of a complete deployment of VMware NSX version 6.2.4, implemented following best practices for software-defined data center deployment using a three cluster configuration,



running VMware ESXi 6.0 and vCenter 6.0.

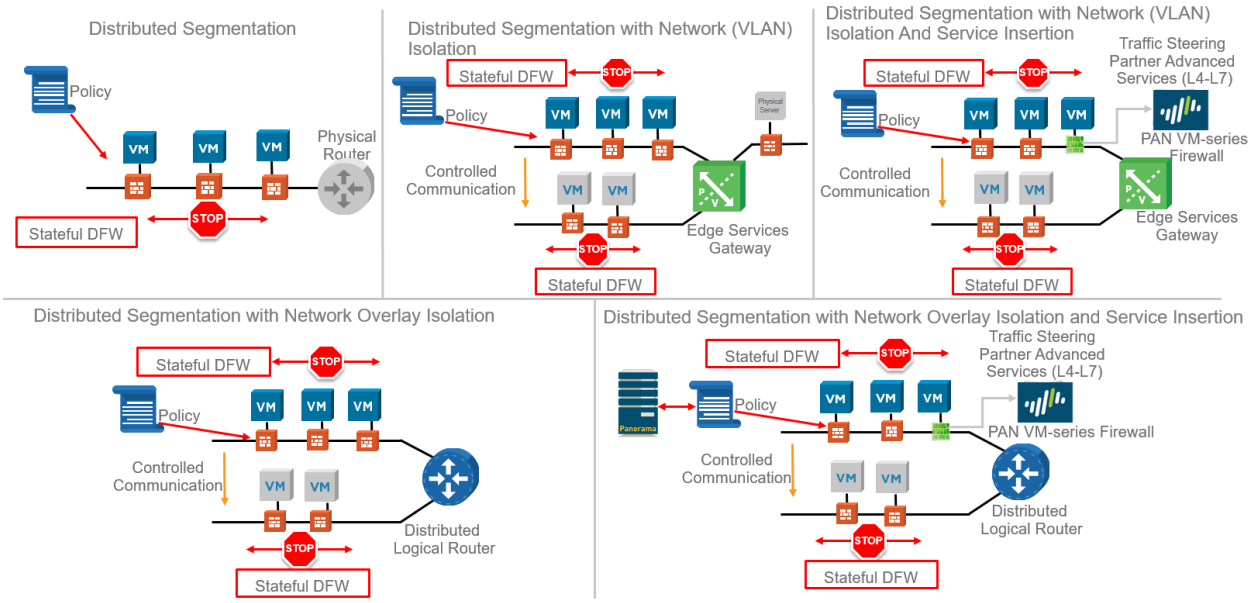
Within this three vSphere cluster arrangement, suitable VMs were created under lab conditions that provided examples of: an exploited server/workstation based on Kali Linux (used to launch threats from); a Windows 8.1 workstation; and a Windows 2003R2 SP2 server. These VMs were templated to facilitate rapid creation and deployment of test machines to populate the network design patterns used for testing.

Complete NSX integration within the vCenter environment, including service virtual machines enabling third-party network introspection, Edge Service Gateways (ESG), logging, L2 switch integration and other features, were enabled. Our NSX test-bed delivered the full feature-set for the product.

NETWORK DESIGN PATTERNS AND TEST METHODOLOGY

Network design patterns represent actual network configurations that are likely to be used in the real-world, as well as those specifically chosen to show test cases that include:

- Flat Network, which are found frequently in smaller software-defined data centers
- L2 and L3 networks with centralized virtual or physical routers, representative of typical data center rack implementations built on hybrid physical and network virtualization platform / distributed virtual switch (dVS)
- Networks with connection to other physical servers
- Overlay-based networks using the Distributed Firewalls (DFW) and Distributed Logical Routers (DLR)
- Physical VLAN and overlay-based networks using service insertion technologies running on dedicated VMs (in our case, Palo Alto Networks NextGen FW with Panorama)



Each of these network design patterns is to be used in a threat simulation, which is based on the following methodology.

THREAT SIMULATION METHODOLOGY

Our examination and testing of the VMware NSX technology is based on simulated exploits that depict likely malware and virus behavior in actual production network scenarios. Our testing uses the Rapid 7 free-edition of MetaSploit, running on a Kali Linux VM. This Kali Linux VM performs the function of an exploited machine, being used as a vector to attack other machines on the network(s).

Real-world attacks typically begin with a successful compromise of this (exploited) machine within the network and then follow with attack propagation to other machines that share the network with the exploited VM. In our testing, we wanted to illustrate the three representative attack types using these Microsoft Windows variants:

- Zero-day attacks, where maximum compromise of the target machine occurs. Maximum compromise gives the attacker complete machine access with full administrative rights. We

chose a legacy version of Windows Server 2003R2, devoid of patches that would remove zero-day vulnerability to our SMB Worm attack (see below) as a target for this exploit.

- Browser-based attacks that exploit weaknesses in browser add-on security, using a fully patched Windows 8.1 workstation.
- Installed application vulnerability to represent the more complex scenario where application software has been installed on an otherwise secure and patched VM. Using a Microsoft Windows 8.1 workstation, we installed the Magento e-commerce storefront package and the required MySQL and free-ware WinAMP bundled components.

Microsoft Windows machines were chosen to maximize the potential effect of the Kali Linux attack and to provide successful exploitation for each of our control scenarios (design patterns 1a, 2a, 3a, etc.). We did not choose our target machines to match likely operating system types typically used on the named networks (VDI, App, DB), where one would expect workstations on VDI and servers on App and DB. Machines were also "softened" by disabling their Windows firewalls and any anti-malware Windows Defender functionality to promote maximum impact by the MetaSploit tools.

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

Our threat simulation focuses on an abbreviated attack scenario based on the **Reconnaissance** and **Exploitation** stages of the kill chain. Specifically, we:

- **Recon** via use of the "db_nmap –v –A {IP Range}" command on the Kali Linux MetaSploit console
- Presume **Weaponization** and **Delivery**, with the particular MetaSploit exploit scenario (see below) chosen with knowledge of its lethality on the target machine(s)
- Invoke **Exploitation** by running the MetaSploit attack and observing the results via the msconsole. Successful exploitation is evident by MetaSploit dropping into the Meterpreter console (SMB and Magento) or other indication of delivery of lethal payload, in the case of the Java ARA
- Abort our threat simulation with an expectation that subsequent **Installation**, **Command and Control** and **Actions on Target** events would follow an actual **Exploitation**

ATTACK VIA METASPLOIT EXPLOITS

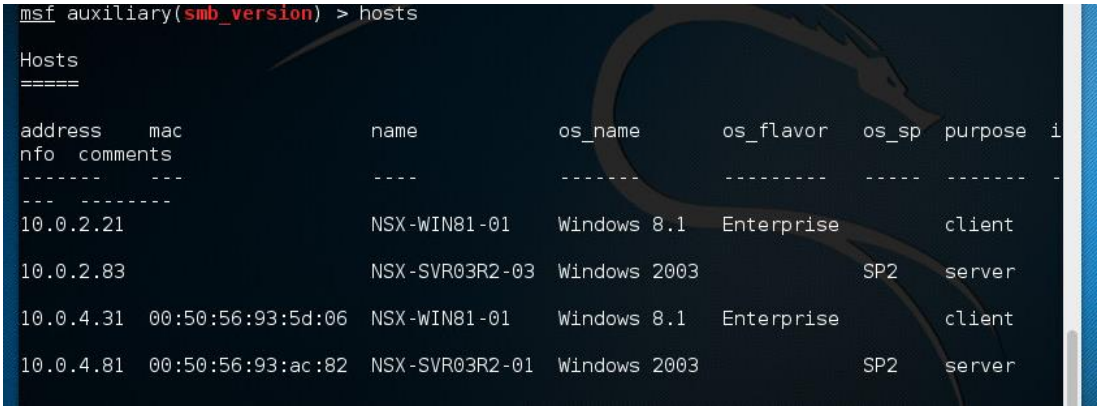
In our particular exploitation scenarios, we are instigating the events through manual use of the MetaSploit console, following these basic steps:

PREPARATION / Recon (2 steps)

DB_NMAP Scanner – All targets – Our MetaSploit console running on the Kali Linux require a set of target hosts in its database. Pen testers typically use NMAP either externally or on the DB_NMAP to generate the list of target IPs to prepare for MetaSploit exploits and use of other utilities. We ran DB_NMAP –v –A before each exploit.

Results from the DB_NMAP command generally create an "early revision" of hosts detail, which has a number of inaccuracies in the table, with approximation of OS version, and other details, which a tester typically corrects by loading and running the auxiliary tool SMB_VERSION.

Auxiliary Tool SMB_VERSION – All targets -- was used to further refine the version, language and option details on all targets. This is a customary step in many Pen Tests to correct inaccuracies in machine OS typing, versions, etc. Our process runs auxiliary/scanner/smb/smb_version before each exploit.



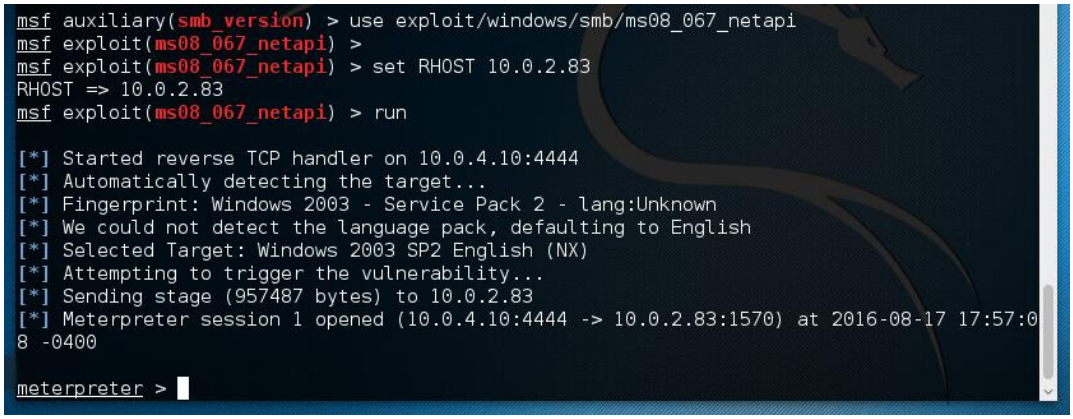
EXPLOITS / Weaponization (one or several used)

We used the following exploits, which are listed here with reference information that is used to locate them in various threat databases or is commonly the "handle" for the particular weapon.

Classic Worm Exploit – SMB MS08-067 Remote Code Execution

Targeting Windows Servers and XP workstations and possibly the most (in)famous exploit of all time for Windows machines, this vulnerability in an SMB Server Service allowing remote code execution on the target machine with full administrator rights. Using MetaSploit exploit/windows/smb/ms_08_067_netapi tools, the target machine becomes completely available for hijacking and total domination. We chose this exploit to show the "worst-case" possibility that may be generated by a "zero day" exploit, which has maximal impact, and uses a necessary service as the vector. In our SMB exploit, our Kali Linux machine acts as the infected PC, and it launches the exploit at other machines in the Network pattern.

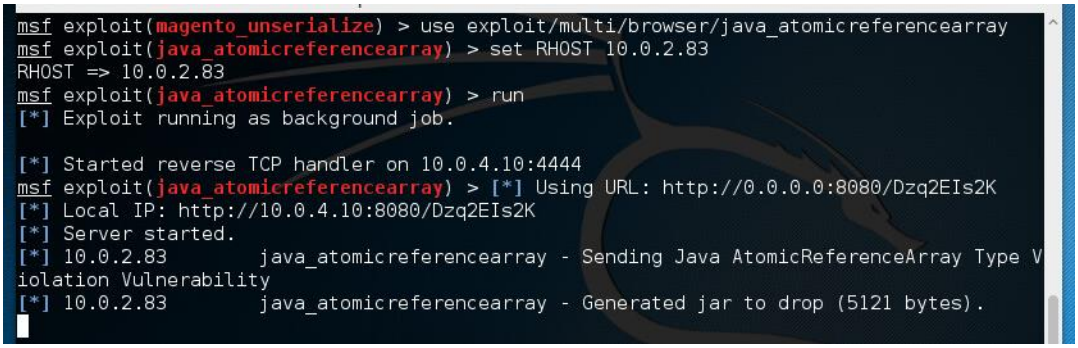
Reference to this exploit is found at: https://www.rapid7.com/db/vulnerabilities/WINDOWS-HOTFIX-MS08-067



Browser-based Java Atomic Reference Array – CVE-2012-0507 JRE Sandbox Escape

Lethal to Windows, Linux, Appliances, etc.; one of the most recognized and pervasive exposures found in 2012, this pervasive exploit leverages an unsafe class of Java variable storage to inject malicious code, which may cause the Java SE/RE to run privileged code on whatever platform might have browsed the deadly source. Our Java attack creates the deadly payload on the Kali Linux machine and makes it available to all vulnerable machines via either browser or machine automated access to the "URL of death." This 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.

When the target machine (10.0.2.83 in this case) browses to the provided web server, the MetaSploit console confirms delivery of the lethal JAR by the confirming message “Sending Java AtomicReferenceArray Type Violation Vulnerability” and a generated JAR message. This lethal JAR is

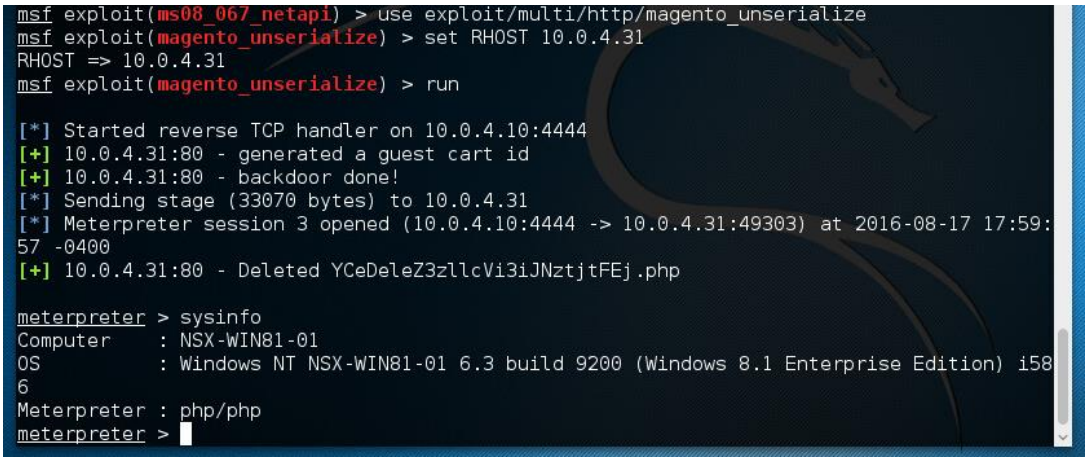


not executed in our testing.

The MetaSploit references for this exploit are:
https://www.rapid7.com/db/modules/exploit/multi/browser/java_atomicreferencearray

Service-based Magento PHP Unserialize – CVE 2016-4010 Remote Code Execution

Windows servers and workstations that are hosting the Magento storefront application are targeted in this prime example of an exploit being created on an otherwise secure Windows server/workstation by installation of an application with a vulnerability in it. The Magento_unserialize Remote Code Execution exploit takes advantage of insecure PHP object injection and subsequent execution of that object as a trusted process. The attacker profiles the machine for the vulnerability with the MSF console check and then launches the exploit, which then serves as a point further exploitation. Frequent use of Windows administrator group credentials for this service gives the intruder a platform for rapid compromise of the target network.



Our use of this exploit is to provide a reference of a current, fully patched Windows machine, to confirm NSX benchmarks and to prove effectiveness the DFW and other features of NSX on modern Windows Server 2012R2 and Windows 8.1/10 targets.

The threat profile and links for this exploit are here:
https://www.rapid7.com/db/modules/exploit/multi/http/magento_unserialize

NSX MICRO-SEGMENTATION DESIGN PATTERNS

Our testing was based on five representative network design patterns that lent themselves to the micro-segmentation use cases. Beginning with the simplest, "flat" networks, these patterns were used to map likely real-world network scenarios to NSX implementation for our testing. Each design pattern is described and technical details of the virtual switching configuration are provided.

For the purposes of clarity, "VLAN" describes a traditional dVS port-group to VLAN configuration, while "network overlay" connotes use of NSX logical switches utilizing VXLAN for logical layer 2 overlays on existing IP networks.

Each design pattern has an "a" and "b" variant. We refer to the "a" variant as the "control", where no NSX micro-segmentation has been enabled. Consider this "control" version the reference point to confirm that the threat is successful.

In the "b" variant, the NSX micro-segmentation function is in effect. We used the NSX Service Composer typically to control application of security group rules and supporting security policies, which contained the firewall rules for the DFW and ESG features.

Design Patterns 1a and 1b – Flat Network with Physical Router

The first design pattern 1a depicts a flat network with no filter policies and no routing policies. As mentioned previously this design is the "control" for a single segment NSX micro-segmentation, which follows in pattern 1b.

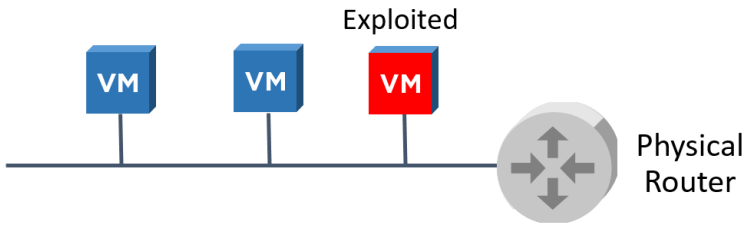


Figure 1 - Pattern 1a Flat Network w/ Physical Router

Setting

Representation of VDI environment
Network IP: 10.0.4.0/24
Compute Cluster
L2 Segment: DPortGroup-VLAN198-VDI

Tests to run (on all network pattern environments)

Two Windows Exploits and one Linux Compromise –
- Microsoft SMB RCE -- Windows
- Magento unserialize RCE – Windows or Linux
- Java ARA – Browser Based Java / Linux Launched

Assets on the Network

- 1 Windows Server 2003R2 and 1 Windows 8.1 VM on user network representing VDI assets (nsx-svr03r2-01 and nsx-win81-01)
- Kali Linux VM representing the compromised VDI assets on the same user network (Kali-Linux-2016.1)
- Physical router or firewall at the edge

Security Narrative

All VDI assets can communicate with each other without any barriers to communicate.

No Rule Set, essentially any source, any destination, any port, is allowed for the network. All east-west traffic is allowed. North-south traffic may be filtered with rules from edge firewall with access from the secure network to the Internet allowed. No direct access from the Internet to the inside network allowed.

SMB Attack is launched directly at the target machine.

Java ARA Exploit creates a .JAR exploit on the Kali (infected) machine, which is browsed by the target(s) and subsequently delivers the infected payload to their JRE.

Expected Outcome

From the exploited virtual desktop, access is gained to all other virtual desktops on the network. All machines are compromised. There is no limit to the proliferation of the exploit to adjacent machines on the network.

SMB attacks result in a compromised host, with MetaSploit dropping into the Meterpreter for subsequent exploitation of the machine from the administrative command line.

Java ARA attacks are engaged by the target machine's browser receiving a payload from http://<Kali>:8080/<scrambled jar>, which is detected on the hosting (port 8080) Kali machine when a successful delivery of the JAR payload has been confirmed. The simulated detonation of the payload is not part of our exploit, nor observed.

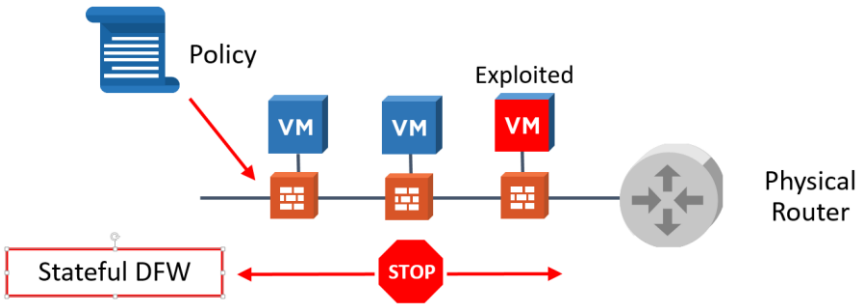


Figure 2 - Pattern 1b Flat Network Micro-Segmented with DFW

Pattern 1b, depicted above, represents the application of micro-segmentation on the flat network design. This was configured via Policy application, based on VMware machine name rule matches in the Security Group configuration as constructed with the NSX Service Composer. Details of this test are described here:

Setting

Similar to 1a, Representative of VDI environment
Network IP: 10.0.4.0/24
L2 Segment: DPortGroup-VLAN198-VDI

Tests to run

Two Windows Exploits and one Linux Compromise –
Microsoft SMB RCE -- Windows

Magento unserialize RCE – Windows or Linux
Java ARA – Browser Based Java / Linux Launched

Assets on the Network

- 1 Windows Server 2003R2 and 1 Windows 8.1 VM on user network representing VDI assets (nsx-svr03r2-01 and nsx-win81-01)
- Kali Linux VM representing the compromised VDI assets on the same user network (Kali-Linux-2016.1)
- Physical router or firewall at the edge

Security Narrative

East-west communication between virtual machines is disallowed by policy applied to the NSX DFW. DFW is protecting each virtual machine. Essentially, each virtual desktop on the network is isolated from each other. Only north-south communication is able to occur as allowed either by the vDFW and/or by the physical router/firewall at the organizations edge. In a typical environment, this would be representative of workstations being allowed to access the Internet for web browsing, access to public web applications and so forth.

Expected Outcome

No proliferation of exploit to adjacent virtual machines. Exploit will not be able to proliferate laterally or perform additional internal reconnaissance.

Design Patterns 2a and 2b – Distributed Segmentation with Network (VLAN) Isolation

In our next set of network designs, we represent a typical multi-segment topology with a VDI network and an App network being routed by the VMware NSX Edge Service Gateway (ESG). On a third network segment (DB in our test environment), a physical server is connected via a dedicated switch integrated to the ESG.

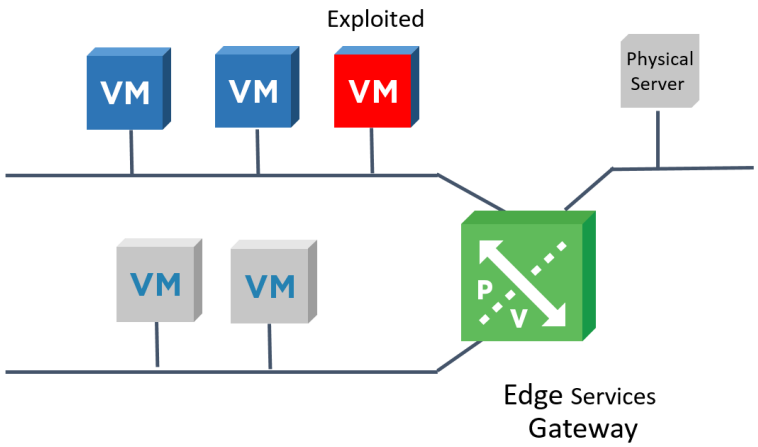


Figure 3 - Pattern 2a Distributed Segmentation with Network (VLAN) Isolation

Setting

Representative of multi layered client-server or client-web/intranet/web app architecture. VLAN segmentation between VDI, DB and App networks. No filter policies or rules restricting access between VLANs.

Network IP VDI: 10.0.4.0/24
L2 Segment VDI: DPortGroup-VLAN198-VDI
Network IP Server: 10.0.2.0/24
L2 Segment Server: DPortGroup-VLAN196-App
Network IP Physical 10.0.3.100 (for the physical server)
L2 Segment Physical: DPortGroup-VLAN197-DB

Tests to run

Nmap recon
Exploits 1 & 2, SMB and JavaARA
Plus Magento unserialize

Assets on the Network

- 1 Windows Server 2003R2 machines and 1 Windows 8.1 PC on VDI network
- Kali Linux VM representing compromised user desktop on VDI network
- 1 Windows Server 2003R2 machines and 1 Windows 8.1 PC on App network
- Physical Machine (Server) – on DB network
- Edge Services Gateway

For the purpose of this test, we used VLAN tagging on vDS Port Groups and utilizing the port groups and a physical network device (Cisco switch) to service the VLANs. The Edge Services Gateway handles the

routing of traffic between the VLANs and handles actual egress to the physical network/VLAN. Access between the hypervisors also uses the physical network switch, as normally required in vDS multi-hypervisor interconnects.

Network Security

Similar to 1a, on the VDI network east-west communication between virtual machines on the user network is unrestricted. The VDI network is segmented by a single VLAN.

The servers on the App (Server) and DB networks are segmented onto separate VLANs.

A route has been set up on the Edge Service Gateway to allow communication between the VDI VLAN assets, the DB VLAN and the App VLAN assets. This north-south traffic can occur without filtering of traffic between VLANs.

Expected Outcome

As in scenario 1a, exploit is allowed to replicate east-west on the VDI VLAN to adjacent desktops. Similarly, the exploit is allowed to replicate north-south to server virtual machines in the App and DB VLANs. Expect NMAP to have full visibility and all exploits to be successful.

The NSX protected pattern 2b follows, which depicts use of the NSX Stateful Distributed Firewall (DFW) delivering per-VM firewall protection, orchestrated by the Service Composer settings, similar to network design pattern 1b shown in the previous section.

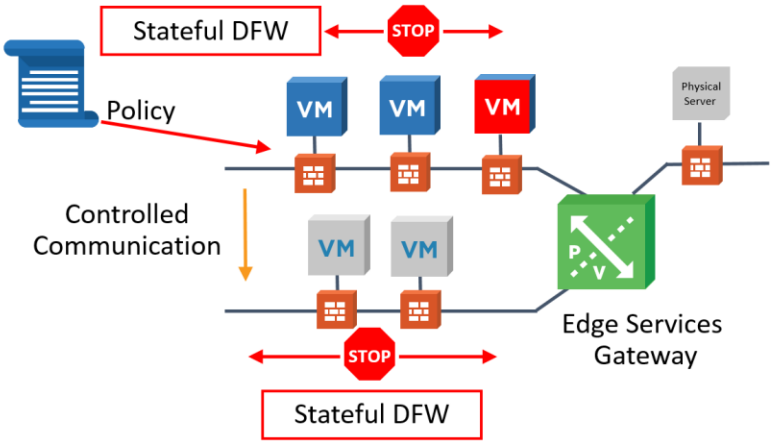


Figure 4 - Pattern 2b Distributed Segmentation with NSX DFW Isolation

Setting

Representative of multi layered client-server or client-web/intranet/web app architecture. VLAN segmentation between VDI, DB and App networks. Routing rules are unrestricted. Distributed micro-segmentation using NSX Distributed Firewall.

Network IP VDI: 10.0.4.0/24
L2 Segment VDI: DPortGroup-VLAN198-VDI
Network IP Server: 10.0.2.0/24
L2 Segment Server: DPortGroup-VLAN196-App
Network IP Physical 10.0.3.100 (for the physical server)
L2 Segment Physical: DPortGroup-VLAN197-DB

Tests to run

Nmap recon
Exploits 1 & 2, SMB and JavaARA
Plus Magento unserialize

Assets on the Network

- 1 Windows Server 2003R2 machines and 1 Windows 8.1 PC on VDI network
- Kali Linux VM representing compromised user desktop on VDI network
- 1 Windows Server 2003R2 machines and 1 Windows 8.1 PC on App network
- Physical Machine (Server) – on DB network
- Edge Services Gateway

For the purpose of this test, we used VLAN tagging on vDS Port Groups and utilizing the port groups and a physical network device (Cisco switch) to service the VLANs. The Edge Services Gateway handles the routing of traffic between the VLANs and handles actual egress to the physical network/VLAN. Access between the hypervisors also uses the physical network switch, as normally required in vDS multi-hypervisor interconnects.

Similar to 2a, on the VDI network, east-west communication between virtual machines on the user network is L2 adjacent, yet prohibited by the NSX firewall rule applied via the Service Composer. The DB (physical server) and App (Servers) networks are also VLAN connected and similarly prohibited by either Block or Reject (see below) on Egress by the NSX DFW.

A route has been set up on the Edge Service Gateway to allow communication between the VDI VLAN assets and the Server VLAN assets. This north-south traffic can occur without filtering of traffic between VLANs. Yet, unlike scenarios 1a and 2a, traffic between L2 networked objects across the routed L3 boundaries are subject to machine-to-machine NSX micro-segmentation actions:

In our test cases, we aimed to either Block or Reject these network operations.

- **Action**: Define enforcement method for this policy rule. Available options are:

Action	Description
Block	Block silently the traffic.
Allow	Allow the traffic.
Reject (introduced since NSX 6.1)	Reject action will send back to initiator: • RST packets for TCP connections. • ICMP unreachable with network administratively prohibited code for UDP, ICMP and other IP connections.

Expected Outcome

Like scenario 1b, except with the extended networks App and DB, access to all devices east-west and north-south will be blocked by the DFW rules. NMAP scans will fail, with “device not available” response.

Reject and Block functions will have differing return-code actions, as the Reject RST packets will affect NMAP and the exploits, with different return codes, but still inaction on the exploits and the recon.

Design Patterns 3a/3b – Distributed Segmentation with Network (VLAN) Isolation and Service Insertion

As an extrapolation of design 2a/2b, this scenario sets the stage to use the Palo Alto Networks (PAN) NextGen firewall virtual appliance to steer traffic into the L4-L7 inspection and action processes of the PAN VA. This diagram depicts the 3b mode of this test. The 3a variant was created by "disable" rules on the firewall actions, configured under Service Composer.

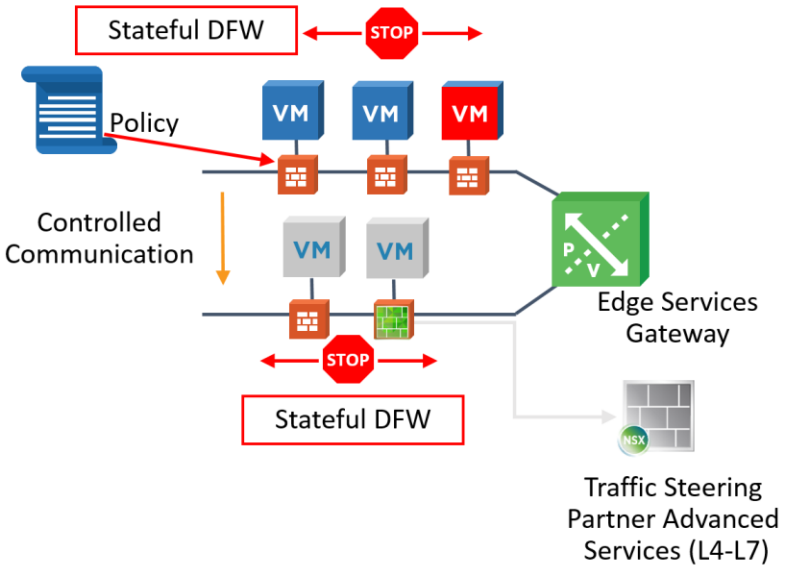


Figure 5 – Patterns 3a/3b Distributed Segmentation with Network (VLAN) Isolation and Service Insertion

Setting

Representative of multi layered client-server or client-web/intranet/web app architecture. VLAN segmentation between VDI and App (Server) networks. Routing rules are unrestricted. Distributed segmentation using NSX Distributed Firewall. Next Gen virtual application firewall service insertion for enhanced packet inspection and policy.

Network IP VDI: 10.0.4.0/24
L2 Segment VDI: DPortGroup-VLAN198-VDI
Network IP Server: 10.0.2.0/24
L2 Segment Server: DPortGroup-VLAN196-App
Traffic Steering via use of Palo Alto Networks (PAN) NextGen Firewall supplying L4-L7 inspection and firewall Features

Tests to run

Linux, Windows and HTTP based exploits –
Browser Based – Java Atomic Reference Array
Confirming test of Magento unserialize
SMB exploit also for confirmation

Assets on the Network

- 1 Windows Server 2003R2 machines and 1 Windows 8.1 PC on VDI network

- Kali Linux VM representing compromised user desktop on VDI network providing poisonous web service via Apache server running port 8080 delivery
- 1 Windows Server 2003R2 machines and 1 Windows 8.1 PC on App network
- Edge Services Gateway
- Service Insertion of Palo Alto Networks or Check Point vSec Appliance applied at the Virtual Machine at the DFW layer – Applied to Web Server in the LAMP Stack

For the purpose of this test, we are using VLAN tagging on vDS Port Groups and utilizing the port groups without going out to a physical network device to service the VLANs. The Edge Services Gateway handles the routing of traffic between the VLANs.

Network Security

L4 Distributed Firewall Rules:

On VDI Network: No communication east-west on the VDI VLAN. Each Virtual Desktop on the network is isolated from each other. Only north-south communication is able to occur. In a typical environment, this would be the workstation allowed to access applications, websites, and data repositories on the network. This scenario is typical for a corporate network with NGFW functions augmenting the SDN.

The servers have been isolated onto a separate App (server) VLAN. For multi-tier applications, the servers could also be segmented onto separate VLANs with a VLAN each representing web, application and database.

Between VDI Network and App (server) Network, north-south Traffic can occur with implicit denies and limited to only necessary traffic. Port 80, 8080 (actual Java ARA exploit) and/or 443 to only the web server on the server VLAN as an example.

Implicit rules to deny traffic from VDI network to the other servers in the LAMP stack. Implicit rule to allow only specified as necessary ports from the web server to the other servers in the LAMP stack.

Layer 4-7 firewall service inserted at critical point, between client and client access at the web server.

Expected Outcome

No proliferation of exploit to adjacent virtual machines in the VDI network. No proliferation of exploit from VDI to App (server) networks. Exploit is stopped at the source of the attack. Automation response could further isolate this compromised desktop for forensic analysis or cleaning.

The nature of the Java ARA exploit is to dangle the poisonous URL to be browser accessed, and successful exploitation consists of delivering the JAR payload to the target browser, be it on Windows, Linux or another device. The nature of the actual attack would follow the execution of the payload; but, for our purposes, the Service Insertion event will prohibit the delivery of the poisonous payload.

Layer 4-7 protections are gained by DFW and Service Insertion.

Design Patterns 4a/4b – Distributed Segmentation with Network Overlay Isolation

In this network topology, we use the NSX Distributed Logical Router (DLR) as the routing entity between the dVS networks, based on service overlay networking. The drawing depicts the 4b variation, where security controls are supplied by NSX Policies. These controls delivery granular policy ACLs to the DFW and DLR components.

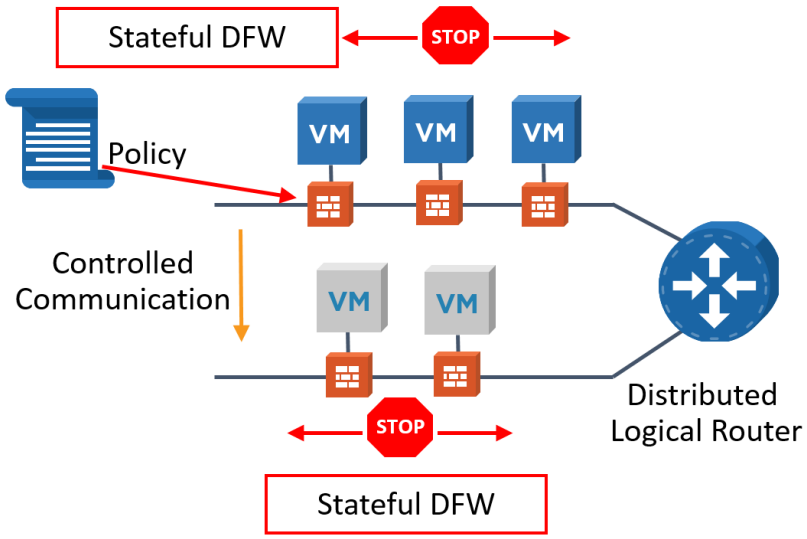


Figure 6 – Patterns 4a/4b Distributed Segmentation with Network Overlay Isolation

Representative of multi layered client-server or client-web/intranet/web app architecture. VLAN segmentation between VDI and App networks. No filter policies or rules restricting access between VLANs. Use of the NSX Distributed Logical Router (DLR) as the routing entity, in place of the Edge Services gateway.

Setting

Network IP VDI: 10.0.4.0/24
L2 Segment VDI: vxw-dvs-56-virtualwire-5-sid-5004-VDI-Tier-01
Network IP Server: 10.0.2.0/24
L2 Segment Server: vxw-dvs-56-virtualwire-3-sid-5002-App-Tier-01

Tests to run

Nmap recon
Exploits 1 & 2, SMB and JavaARA
Plus Magento unserialize

Assets on the Network

- 1 Windows Server 2003R2 machines and 1 Windows 8.1 PC on VDI network
- Kali Linux VM representing compromised user desktop on VDI network
- 1 Windows Server 2003R2 machines and 1 Windows 8.1 PC on App network
- Physical Machine (Server) – on DB network
- Edge Services Gateway

For the purpose of this test, we are using NSX VXLAN overlay L2 segment. The NSX Distributed Logical Router handles the routing of traffic between the L2 segments. Although access between the hypervisors uses the physical network switch ports, the VXLAN overlay maps all physical L2 VLANs in use by those switches.

Network Security

Similar to 1a, on the VDI network, east-west communication between virtual machines on the user network is unrestricted. The VDI network is segmented by a single VXLAN overlay. The App (Server) network is segmented on another VXLAN overlay.

A route has been set up on the DLR to allow communication between the VDI and App VXLAN overlay assets. This north-south traffic can occur without filtering of traffic between overlays.

Expected Outcome Scenario 4a (Rules Allow)

As in scenario 1a, exploit is allowed to replicate east-west on the VDI overlay to adjacent desktops. Similarly, the exploit is allowed to replicate north-south to server virtual machines in the App and DB VXLANs. Expect NMAP to have full visibility and all exploits to be successful.

Expected Outcome Scenario 4b (Rules Block and Reject)

Like scenario 1b, except with the extended App network, access to all devices east-west and north-south will be blocked by the DFW rules. NMAP scans will fail, with “device not available” response.

Reject and Block functions will have differing return-code actions, as the Reject RST packets will affect NMAP and the exploits, with different return codes, but still inaction on the exploits and the recon.

Design Patterns 5a/5b – Distributed Segmentation with Network Overlay Isolation and Service Insertion

Our final network topology adds service insertion in the 5b, "non-control" variant of our testing. With no service insertion, 5a maps directly to the 3a event. If no traffic is steered to the PAN NextGen firewall, no Layer 4-7 inspection and action will take place.

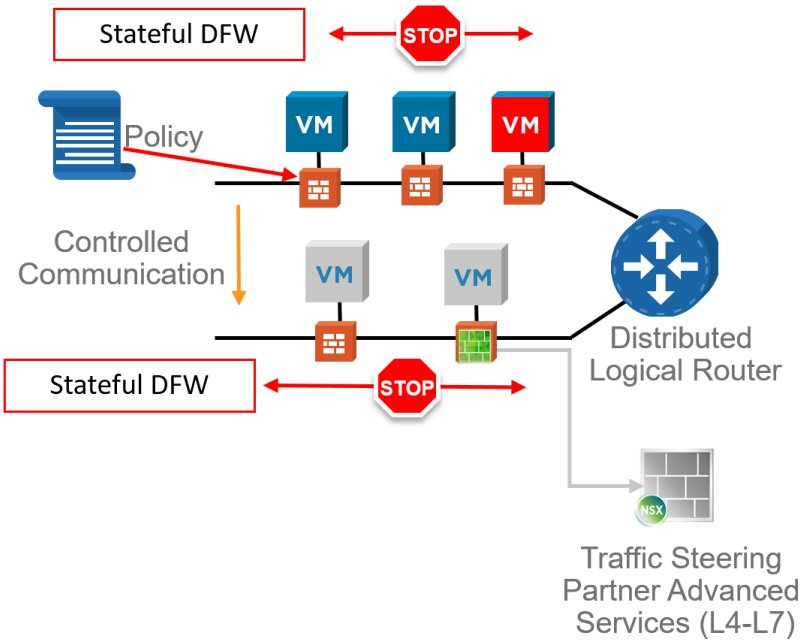


Figure 6 – Patterns 5a/5b Distributed Segmentation with Network Overlay Isolation and Service Insertion

Setting

Representative of multi layered client-server or client-web/intranet/web app architecture. VLAN segmentation between VDI and App (Server) networks. Routing via the Distributed Logical router with rules that are unrestricted. Distributed segmentation using NSX Distributed Firewall. Next Gen virtual application firewall service insertion for enhanced packet inspection and policy.

Network IP VDI: 10.0.4.0/24
L2 Segment VDI: vxw-dvs-56-virtualwire-5-sid-5004-VDI-Tier-01
Network IP Server: 10.0.2.0/24
L2 Segment Server: vxw-dvs-56-virtualwire-3-sid-5002-App-Tier-01
Traffic Steering via use of Palo Alto Networks (PAN) NextGen Firewall supplying L4-L7 inspection and firewall Features

Tests to run

Linux, Windows and HTTP based exploits
- Browser Based – Java Atomic Reference Array
- Confirming test of Magento unserialize
- SMB exploit also for confirmation

Assets on the Network

- 1 Windows Server 2003R2 machines and 1 Windows 8.1 PC on VDI network
- Kali Linux VM representing compromised user desktop on VDI network providing poisonous web service via Apache server running port 8080 delivery
- 1 Windows Server 2003R2 machines and 1 Windows 8.1 PC on App network
- NSX Distributed Logical Router (DLR)
- Service Insertion of Palo Alto Networks VM-Series Panorama firewall applied at the Virtual Machine at the DFW layer – Applied to Web Server in the LAMP Stack

For the purpose of this test, we are using NSX VXLAN overlay L2 segment. The NSX Distributed Logical Router handles the routing of traffic between the L2 segments. Although access between the hypervisors uses the physical network switch ports, the VXLAN overlay maps all physical L2 VLANs in use by those switches.

Traffic steering is enabled on the DFW to re-route traffic from VDI machines accessing Web services on the App Network.

Network Security

L4 Distributed Firewall Rules:

On VDI Network: No communication east-west on the VDI VXLAN overlay. Each Virtual Desktop on the network is isolated from each other. Only north-south communication is able to occur. In a typical environment, this would be the workstation allowed to access applications, websites, data repositories on the network. This scenario is typical for a corporate network with NGFW functions augmenting the SDN.

The servers have been isolated onto a separate App (server) overlay. For multi-tier applications, the servers could also be segmented onto separate L2 Segment VXLANs, for web, application and database isolation.

Between VDI Network and App (server) Network, north-south traffic can occur with implicit denies and limited to only necessary traffic. Port 80, 8080 (actual Java ARA exploit) and/or 443 would be permitted to the web server on the server VXLAN, for example.

Implicit rules to deny traffic from VDI network to the other servers in the LAMP stack. Implicit rule to allow only specified as necessary ports from the web server to the other servers in the LAMP stack.

Layer 4-7 firewall service inserted at critical point, between client and client access at the web server.

Expected Outcome

No proliferation of exploit to adjacent virtual machines in the VDI network. No proliferation of exploit from VDI to App (server) networks. Exploit is stopped at the source of the attack. Automation response could further isolate this compromised desktop for forensic analysis or cleaning.

The nature of the Java ARA exploit is to dangle the poisonous URL to be browser accessed, and successful exploitation consists of delivering the JAR payload to the target browser, be it on Windows, Linux or another device. The nature of the actual attack would follow the execution of the payload, but, for our purposes, the Service Insertion event will prohibit the delivery of the poisonous payload.

Layer 4-7 protections are gained by DFW and Service Insertion.

VALIDATION EXERCISES AND FINDINGS

The process of validation consisted of running the aforementioned Kali Linux VM-based reconnaissance (Recon) and MetaSploit exploits (Exploit) against the listed servers, in the configurations depicted in detail in the **NSX Micro-segmentation Design Patterns** section above. Each test typically consisted of a control "a", where no NSX protections were employed, and an NSX protected "b" variation.

Here we present our summarized findings, on a per-design pattern basis. Our validation exercises and the results are illustrated with supporting screenshots of NSX consoles and of the actual VMs under test. Some test results had nuanced outcomes, which, when listed in the table, appear with a yellow color-code and a reference to more information, which appears in the text below the table.

Our tables use these color-codes with the meanings described:

KEY

Unimpeded – No action of NSX on network flow

NSX Controlled – NSX Firewalled network flows

" w/ caveat – Modified network flows by NSX

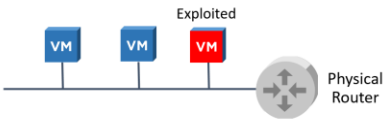
Kill Chain designation matches chart on right

Meaning	Color
Unimpeded	
NSX Controlled	
NSX Controlled w/ caveat	
Kill Chain Recon	
Kill Chain Exploit	

- Reconnaissance
- Weaponization
- Delivery
- Exploitation
- Installation
- Command & Control
- Actions on Target

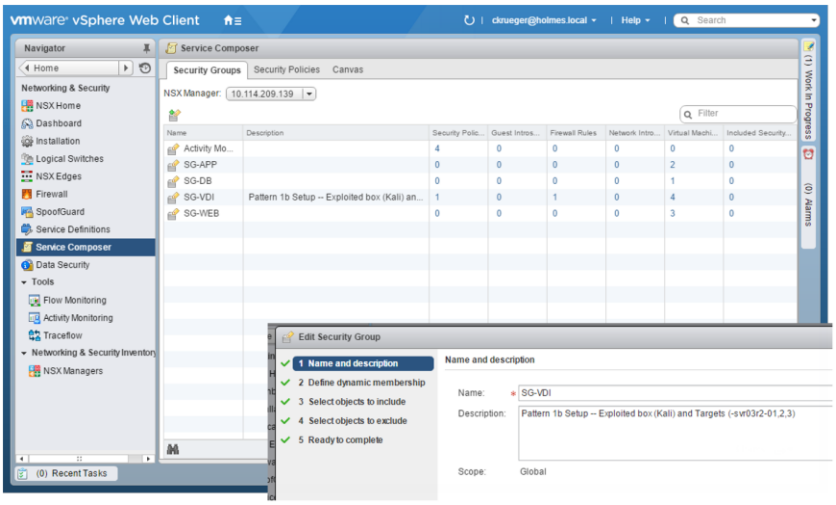
Design Patterns 1a and 1b – Flat Network with Physical Router

In the simplest configuration, our single-segment flat network topology, we observed this behavior:

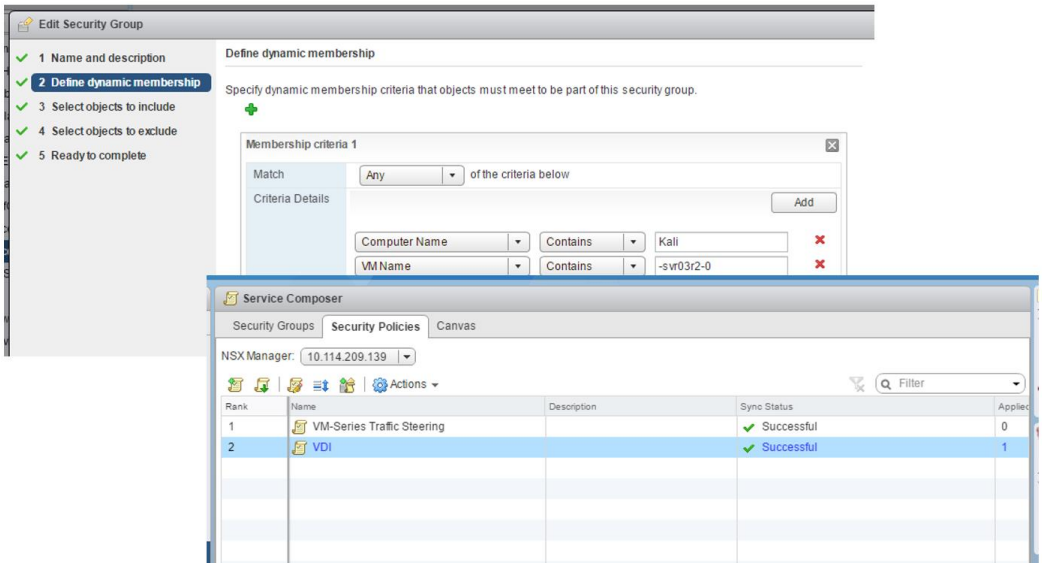


1: Flat Network with Physical Router	Recon (nmap)	Recon (aux SMB)	SMB Exploit	JavaARA Exploit	Magento Exploit
Control 1a					
NSX 1b					

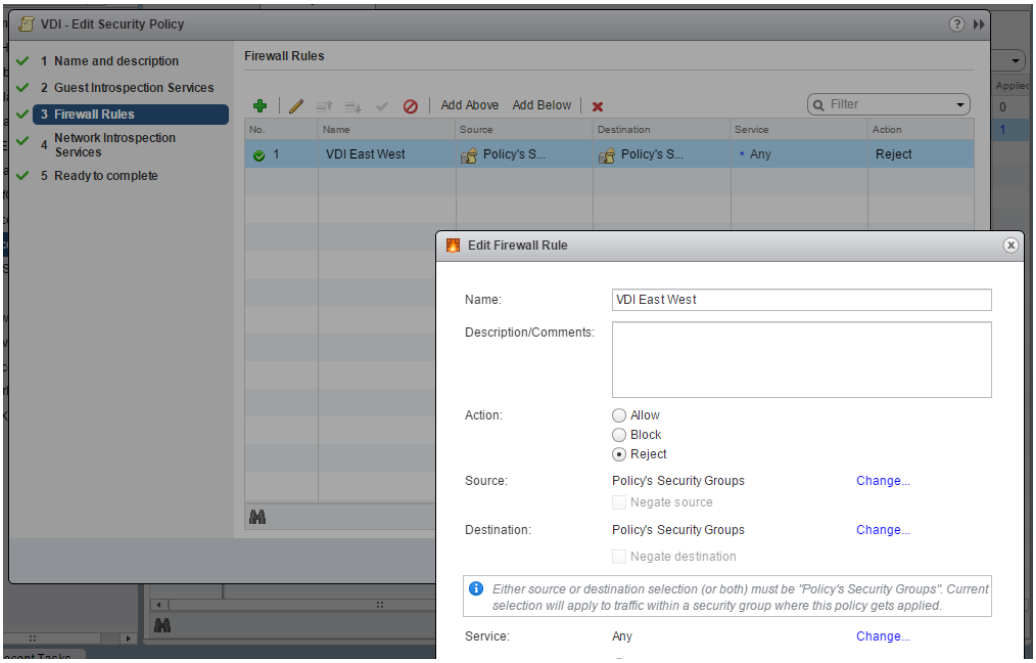
NSX distributed firewall (DFW) functionality was constructed to protect the E-W traffic in this environment, yet left in the "Allow" action state for the control scenario 1a. Using the service composer, Security Group SG-VDI had was configured to include the exploited Kali, the Win8.1 and Server 2003R2 VMs…



…by application of membership to a security policy VDI…

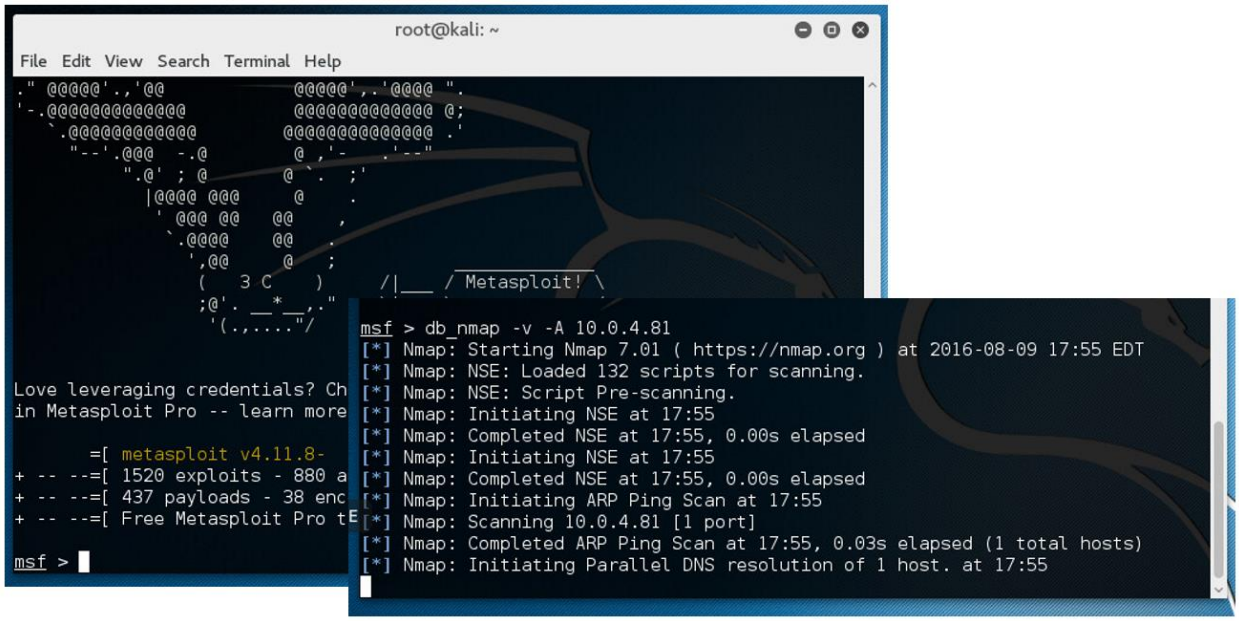


…which contained the following firewall rule to be applied to the E-W traffic patterns and was subsequently used to NSX Reject (and Block) DFW action in benchmark 1b (after being used in Allow action for 1a):

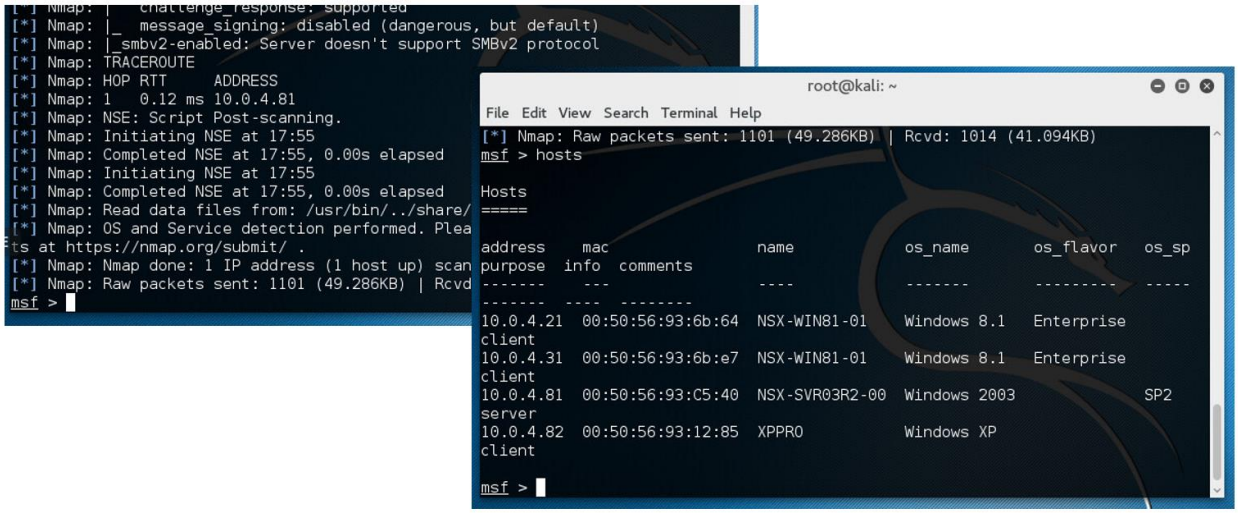


Shown in the results table above, the 1a behavior matched the expected result for unimpeded recon and exploit, which we see in the following series of screenshots, which show the actual recon and SMB exploit events as performed on the nsx-svr03r2-01 VM:

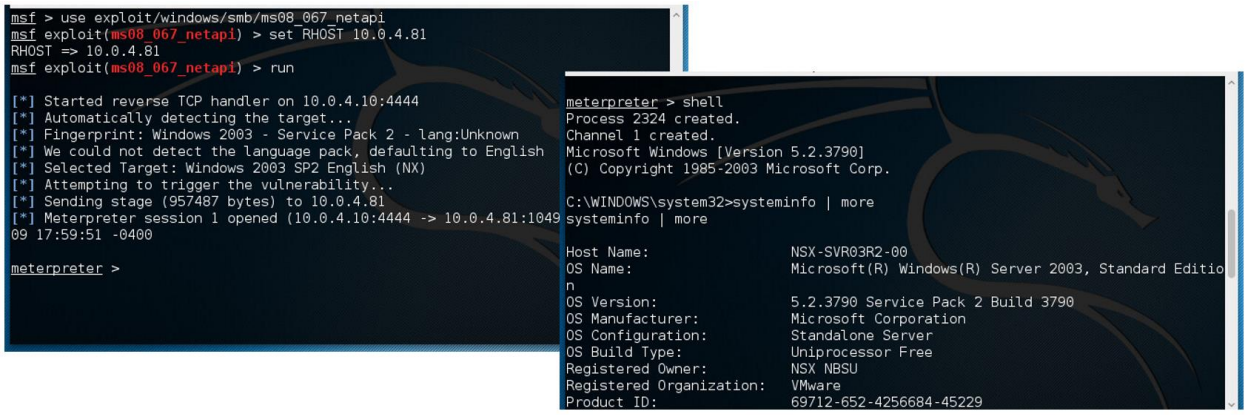
Kali Linux launch of msfconsole, issuing the initial DB_NMAP of target, which is the first Recon step…



… resulting in an addition to the MS_DB of the hosts, and after Recon step 2, the auxiliary SMB_VERSION tool, gets refined to include fine-grained details of the host profile(s):

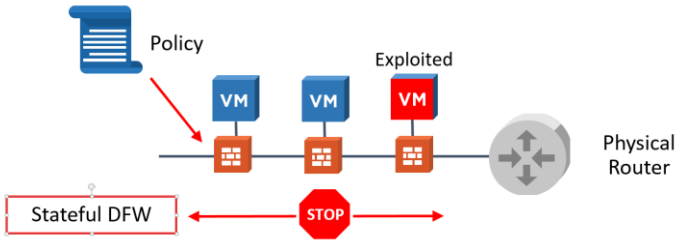


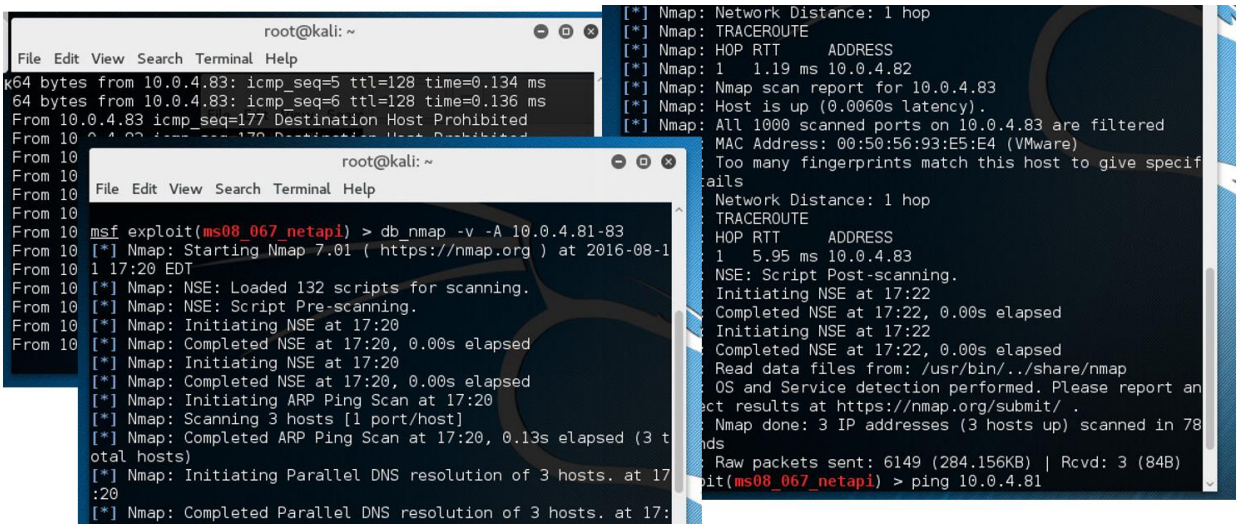
With intact target host information in the MS_DB, we can proceed with the SMB Exploit against the vulnerable Windows target machine…



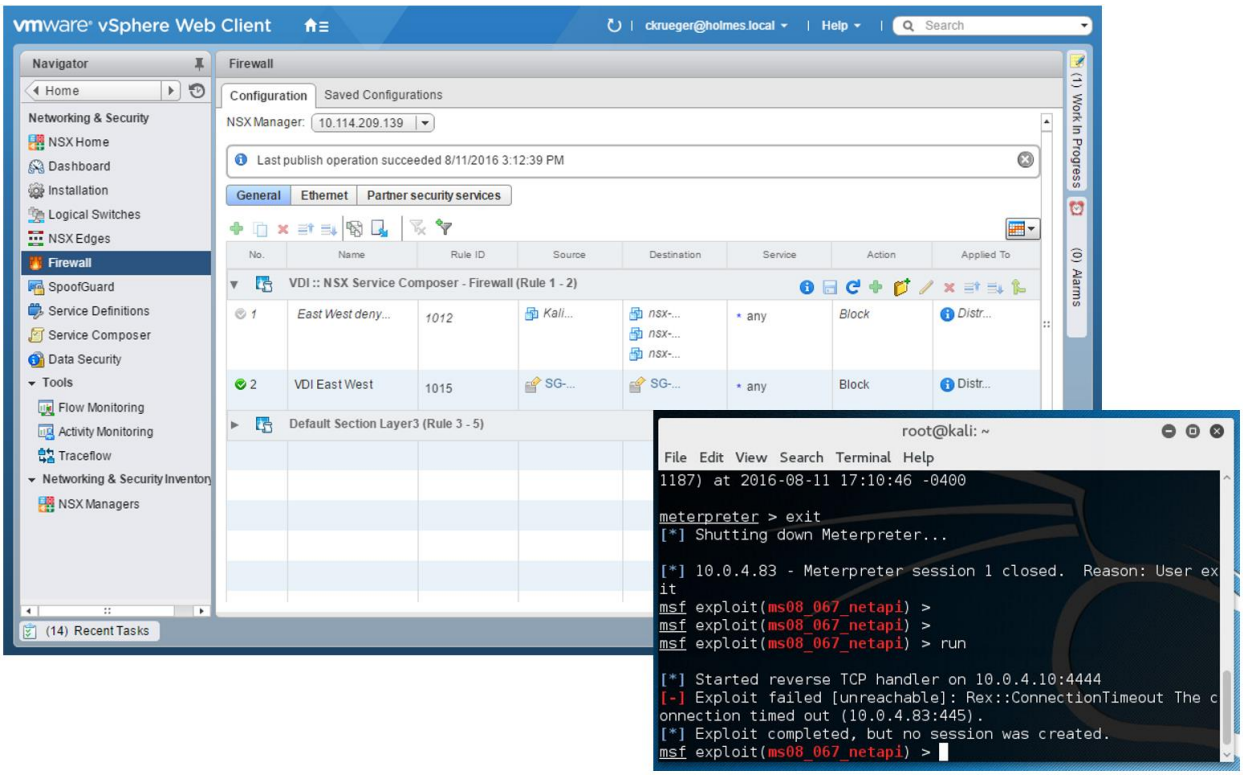
… where we now have unfettered access to through the meterpreter console. Exploited success in 1a, followed by similar success in our subsequent JavaARA and Magento exploits.

Scenario 1b begins with the modification of our firewall rule to Reject, which immediately eliminates all access to the target machines with this “Destination Host Prohibited” ping response and similar impacts to the reconnaissance DB_NMAP and Aux SMB_Version MetaSploit activities:





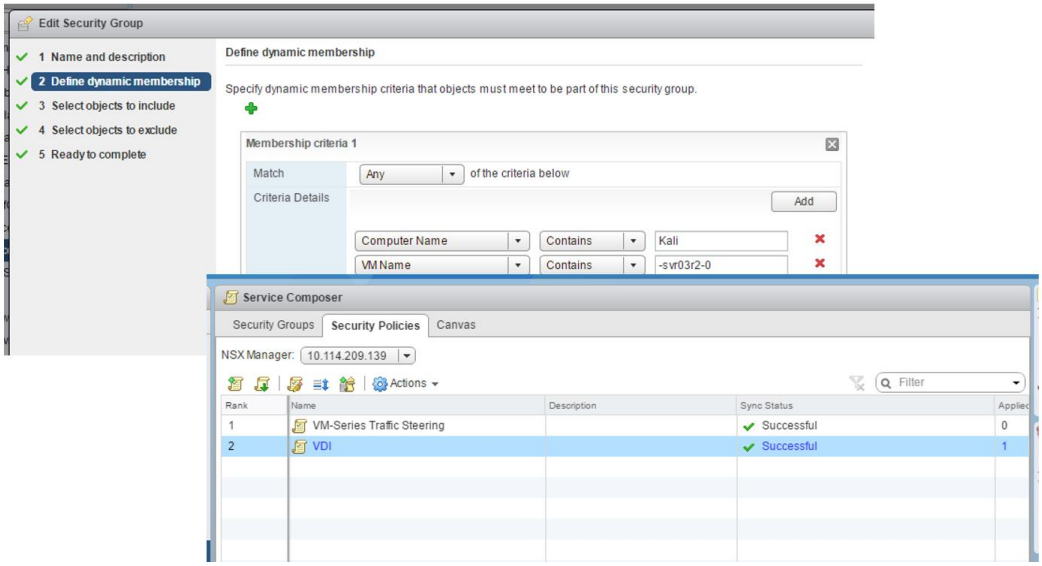
All subsequent exploits are impossible without useful recon. Should a HOST_DB table be constructed or otherwise created, the exploit similarly fails, without E-W access to the target:



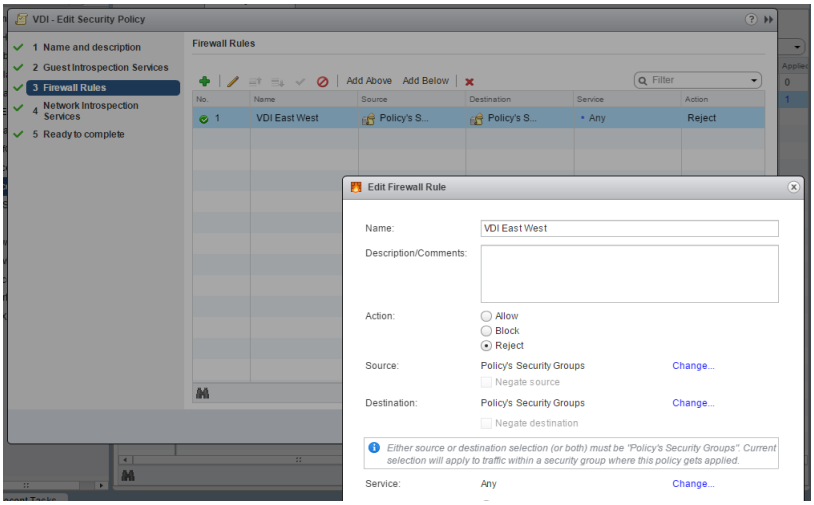
Design Patterns 2a and 2b – Distributed Segmentation with Network (VLAN) Isolation

As a logical extension of the 1a/1b flat network pattern, in design pattern 2, we are creating a Layer 3 network with the addition of the NSX Edge Services Gateway (ESG) and providing routing services to an additional pair of networks. The physical server is on the dVS DB network, while an additional pair of servers (lower left, nsx-win81-01 and nsx-svr03r2-03) are on the dVS App Network.

Using the NSX Service Composer, we further used the Security Group created for pattern 1a/1b and extended a rule to the Layer 3 domain to support the additional testing of the physical server connection via the Edge Services Gateway (ESG). Security Group configuration is illustrated in the following screenshots:

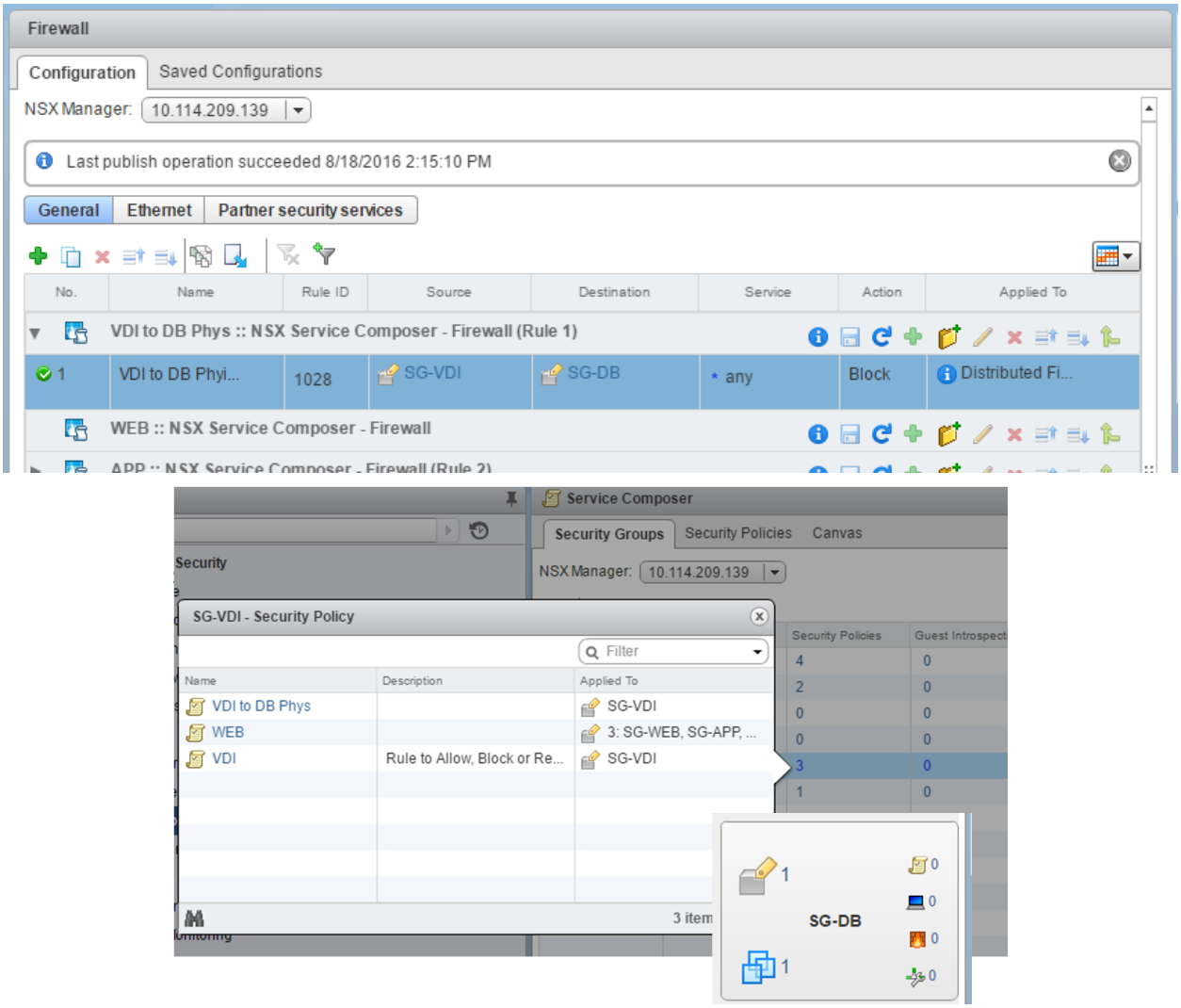


Application of the security policy firewall rule was configured using the Service Composer, also in a similar fashion to the earlier design pattern.

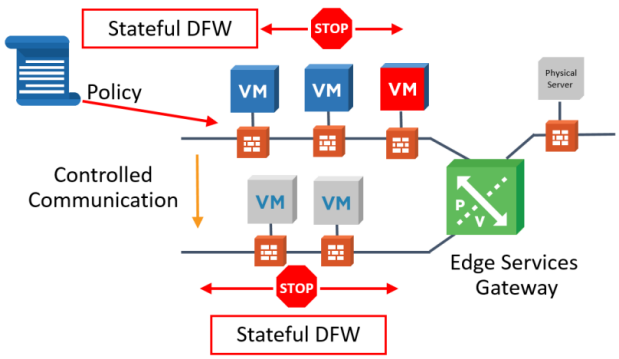


The NSX ESG is used to create a VMware VXLAN to VLAN conduit, permitting the physical server to reside on dedicated physical switching outside of the VMware SDN, with support for the native VLANs on the (Cisco Nexus) switch.

Physical server DFW/ESG configuration is by way of an additional rule constructed to restrict the traffic between the DB network containing the physical server and the VDI network by creating a “nested policy” SG-DB, which contained a firewall rule to restrict access. The “block” version of that rule is depicted in this screenshot, with a Canvas View from the Service Composer which follows:

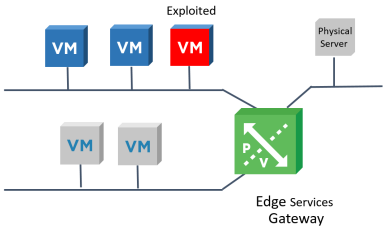


In the 2b test, NSX is enabled, again using the Service Composer, with additional rules imposed within the service groups (now 3 of them, VDI, APP and DB) to create the intended NSX protected design pattern:



Note: the use of Stateful Distributed Firewall (DFW) on all five VMs and on the connection to the DB network via the ESG. Although the actual physical machine lies outside of the NSX data/DFW plane, the ingress/egress to the outboard switch is within the control of NSX, and policy-based rules with IP 5-tupple stateful firewall actions may apply.

Results from our testing of the 2a/2b design patterns are in this table:



2: Distributed Segmentation with VLANS	Recon (nmap)	Recon (aux SMB)	SMB Exploit	JavaARA Exploit	Magento Exploit
Control 2a					
NSX 2b	2b-i	2b-ii			
Phys Control 2a					
Phys NSX 2b					

As with the 1a/1b designs, expected NSX protection under 1b scenario was achieved with special comments on the recon activities.

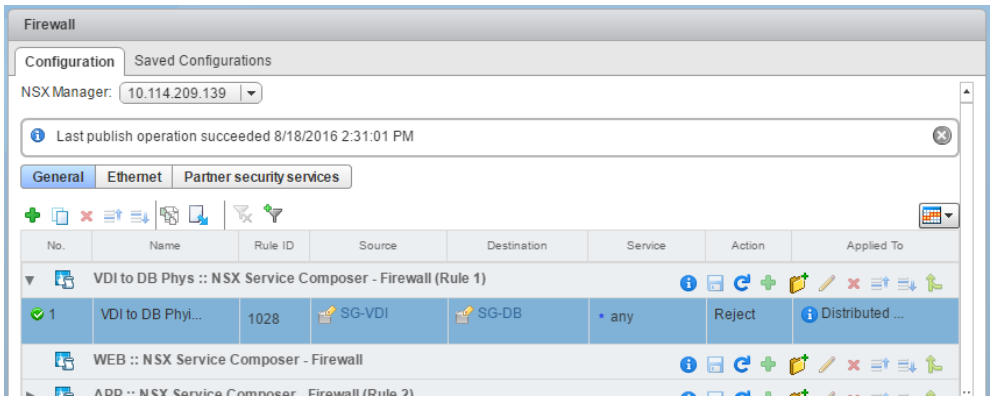
2b-i) Caveat: DB_NMAP activity differs based on the NSX DFW actions of Block versus Reject. In the Block action, which will "Block silently…", the rule permits the DB_NMAP to detect the presence of the target due to traceroute "hits", although all port scanning fails. Reject denies this profiling due to the return code being "network administratively prohibited" and traceroute failing with matching errors for all tested IPs. Recon, when performed by a typical intruder, is performed on a wide range of IPs, and the Reject action gives them no details about any targets and would be preferred.

2b-ii) Caveat: Similar to DB_NMAP, the Aux SMB_Version will fail, with no host entries in the MS_DB to begin with. Should a manually prepared host entry be present, both Block and Reject actions allow no refinement of target identity information, and exploitation will fail.

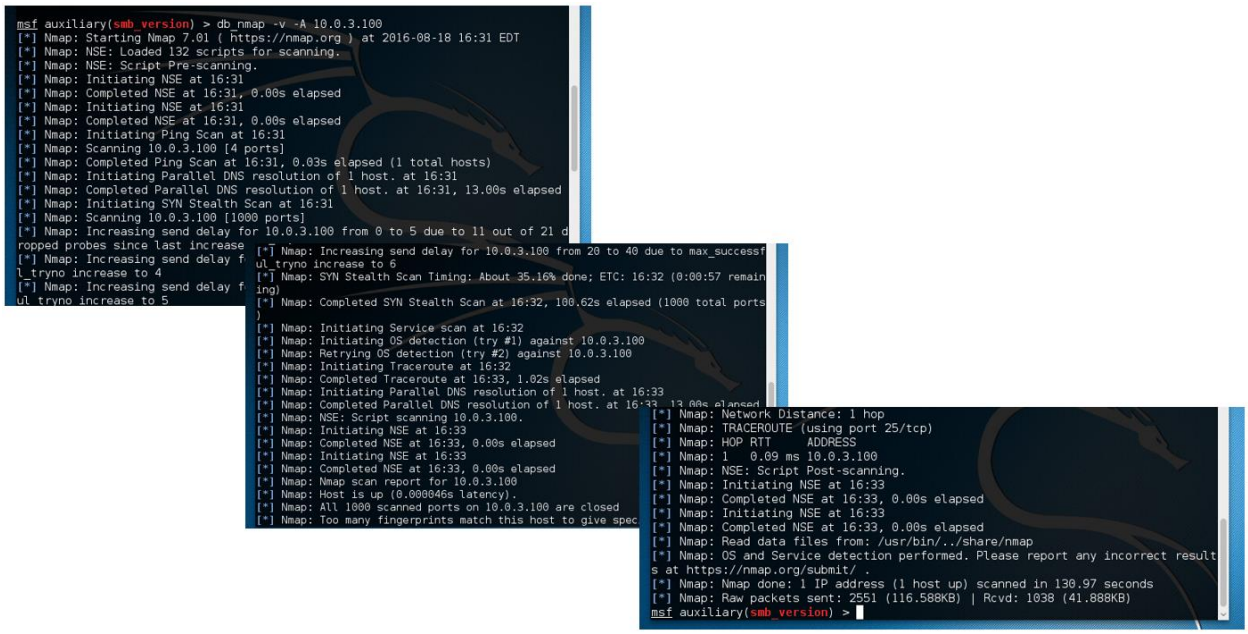
Screenshots of the Block and Reject behavior begin with an examination of the DB_NMAP activity response on the MetaSploit console in this example of Block DFW rule response, which illustrates a [host down] finding:

```
msf auxiliary(smb_version) > db_nmap -v -A 10.0.3.100
[*] Nmap: Starting Nmap 7.01 ( https://nmap.org ) at 2016-08-18 16:16 EDT
[*] Nmap: NSE: Loaded 132 scripts for scanning.
[*] Nmap: NSE: Script Pre-scanning.
[*] Nmap: Initiating NSE at 16:16
[*] Nmap: Completed NSE at 16:16, 0.00s elapsed
[*] Nmap: Initiating NSE at 16:16
[*] Nmap: Completed NSE at 16:16, 0.00s elapsed
[*] Nmap: Initiating Ping Scan at 16:16
[*] Nmap: Scanning 10.0.3.100 [4 ports]
[*] Nmap: Completed Ping Scan at 16:16, 3.03s elapsed (1 total hosts)
[*] Nmap: Nmap scan report for 10.0.3.100 [host down]
[*] Nmap: NSE: Script Post-scanning.
[*] Nmap: Initiating NSE at 16:16
[*] Nmap: Completed NSE at 16:16, 0.00s elapsed
[*] Nmap: Initiating NSE at 16:16
[*] Nmap: Completed NSE at 16:16, 0.00s elapsed
[*] Nmap: Read data files from: /usr/bin/../share/nmap
[*] Nmap: Note: Host seems down. If it is really up, but blocking our ping probe
s, try -Pn
[*] Nmap: Nmap done: 1 IP address (0 hosts up) scanned in 3.90 seconds
[*] Nmap: Raw packets sent: 8 (304B) | Rcvd: 0 (0B)
msf auxiliary(smb_version) >
```

Whereas the Reject response, configured with this DFW Policy action …



… provides DB_NMAP feedback that shows the target does exist, but cannot be probed for details of port response, as shown in this complete trace here:



Design Patterns 3a/3b – Distributed Segmentation with Network (VLAN) Isolation and Service Insertion

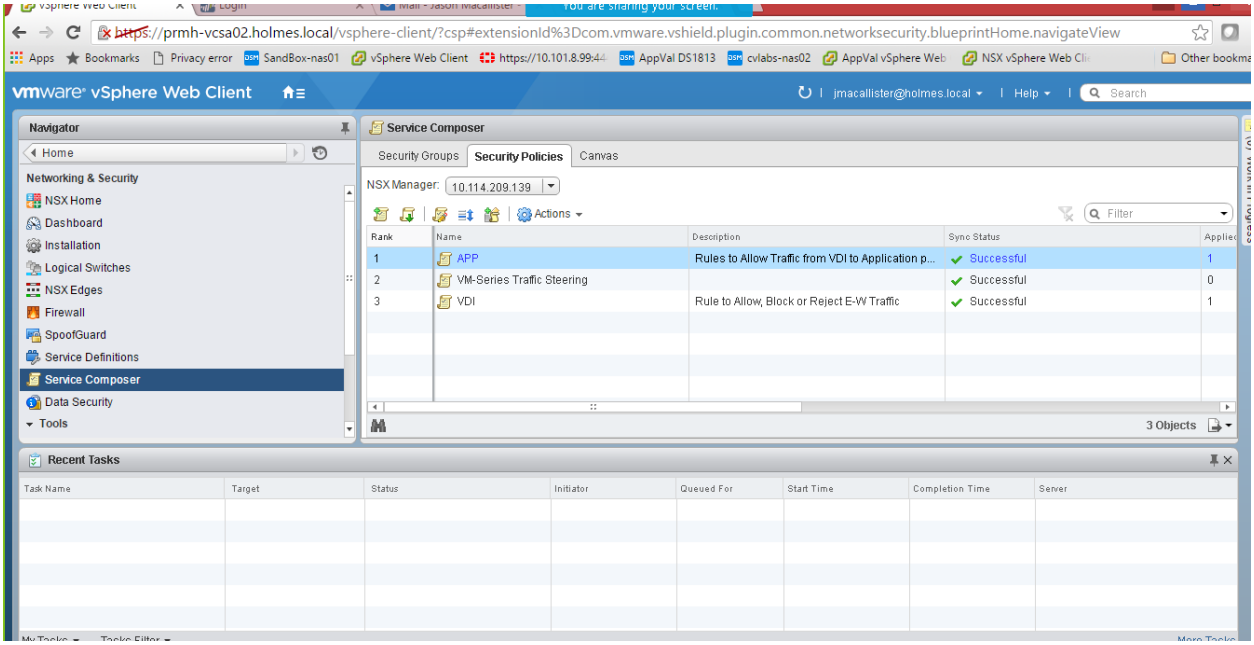
"Service Insertion" furthers the logical utility of NSX to support L4-L7 inspection and leveraging third-party firewalls and appliances, design patterns 3a/3b make use of the traffic steering features of NSX to hand-off the complex inspection and action steps of those higher ISO layers.

In these design patterns, we use the Palo Alto Networks (PAN) VM-Series Panorama Firewall via service insertion to inspect the Layer 7 HTTP transactions performed by the JavaARA and Magento exploits. This L7 inspection detects and responds to threats that could not be protected when more restrictive rules prohibiting E-W and N-S transit, as is typical in real-world network service scenarios required by lines of business, cannot be used.

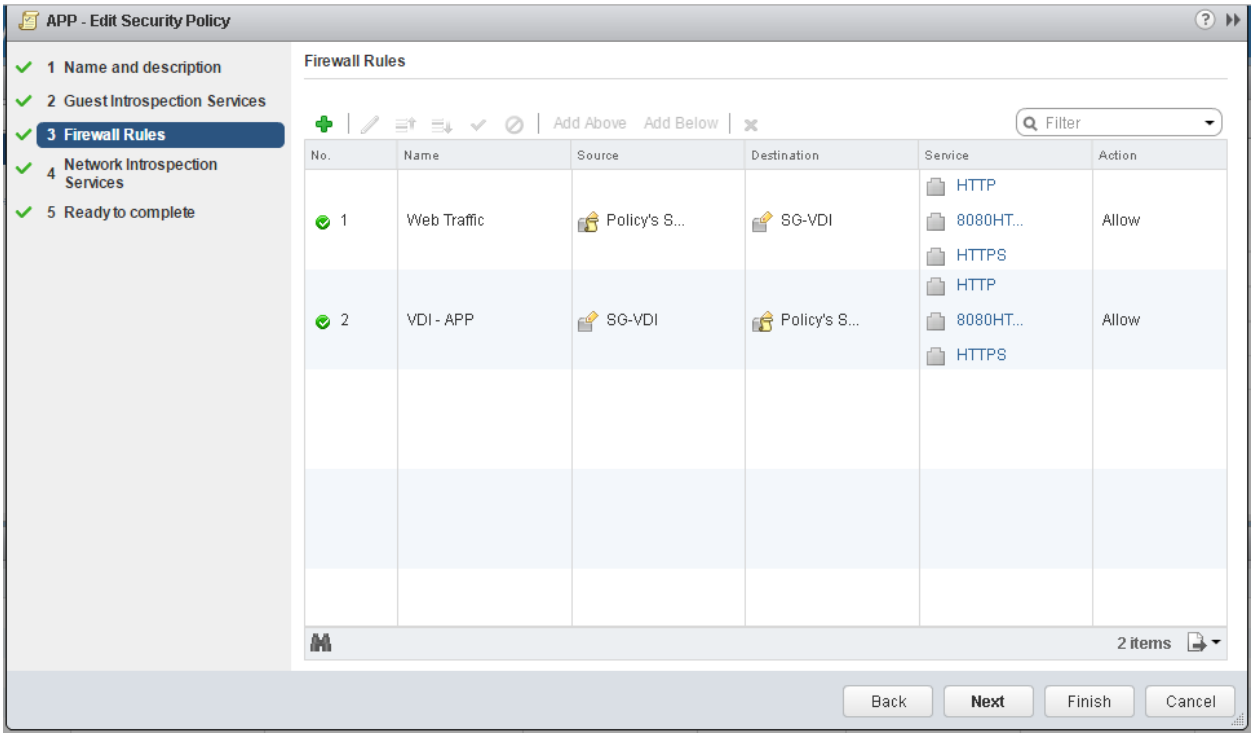
JavaARA and Magento exploits simulate traditional Java web/JAR transactions and storefront application delivery. These two examples illustrate browser-based vulnerabilities that can only be protected by Layer 7 inspection, detection and response outside of the Layer 2-4 capacity of NSX.

The control case, 3a, where service insertion is disabled and E-W rules are in the allow state, generated identical results to the patterns 1a and 2a, with JavaARA and Magento exploits being successful.

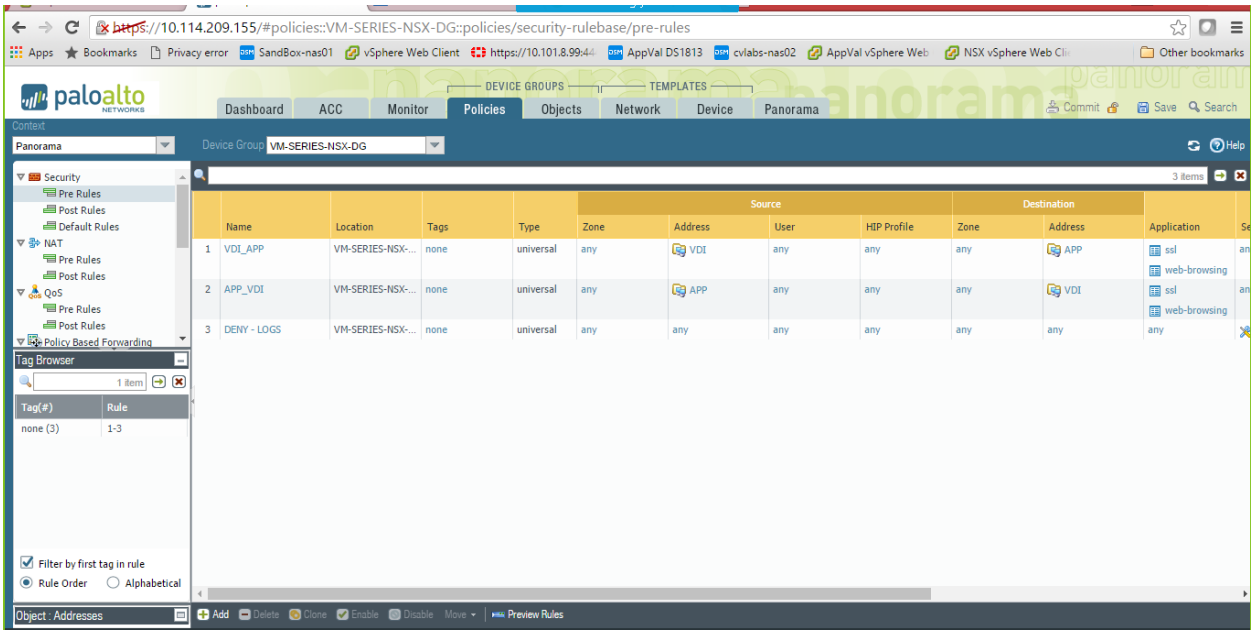
Design pattern 3b followed in our testing and was facilitated by initially configuring service insertion to steer traffic to the Panorama Firewall, shown in the following configuration screenshots:

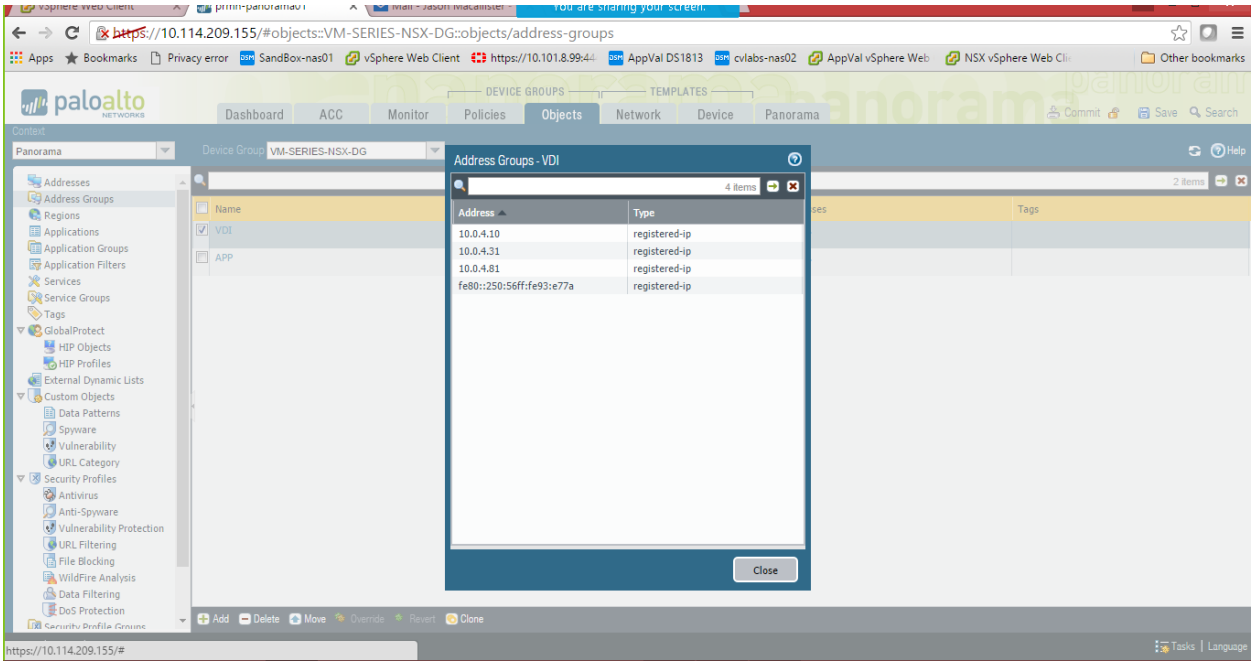


Allowed security policies (note the port 8080 for JavaARA support) are shown here:

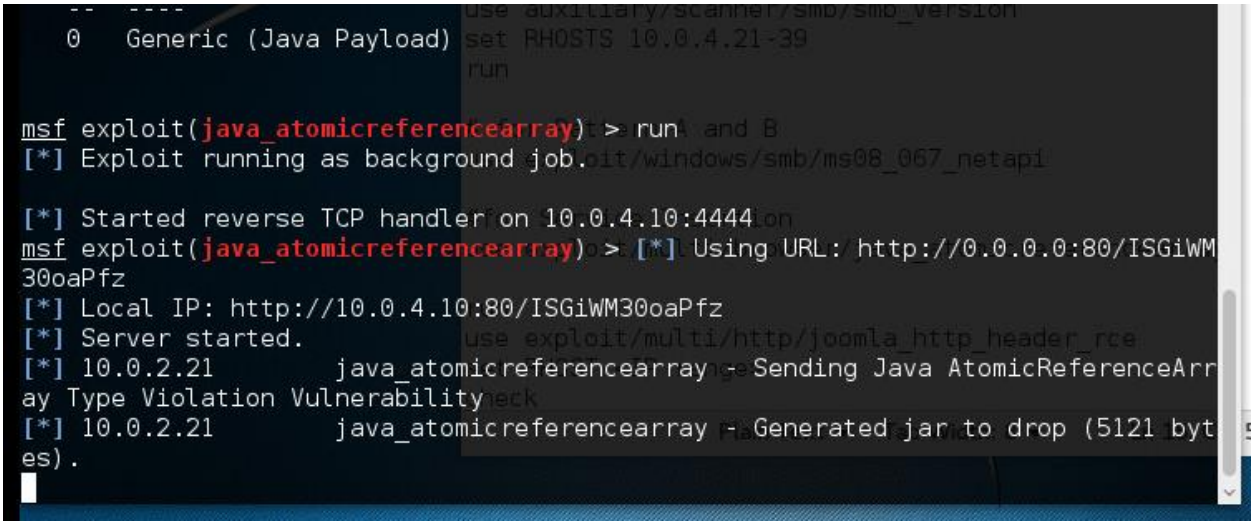


Palo Alto Panorama screens summarize the setting for the security policy rule creation for the L7 firewall:

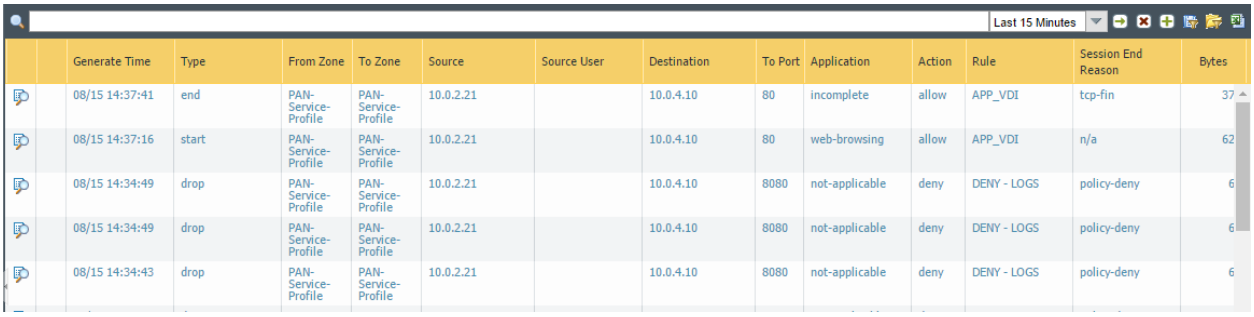




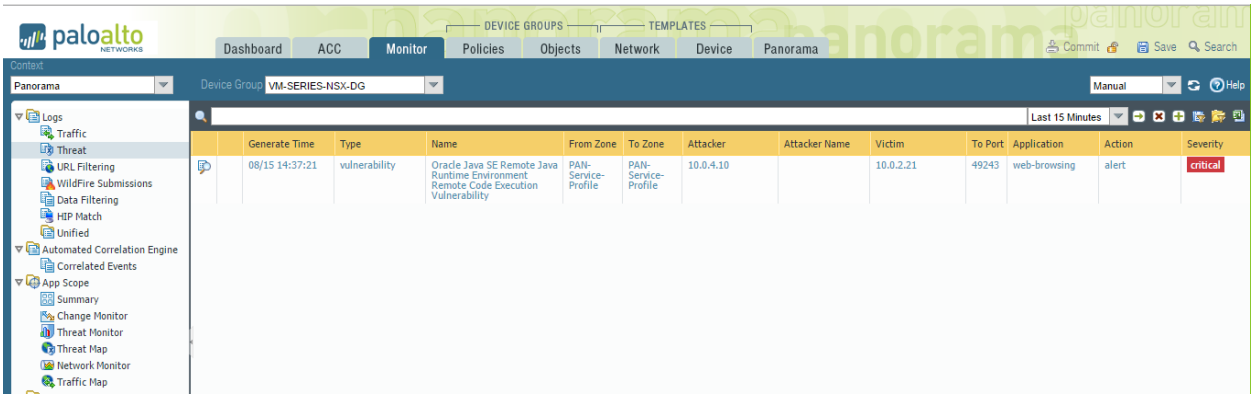
And then the execution of the JavaARA attack, shown here, on the MetaSploit console…



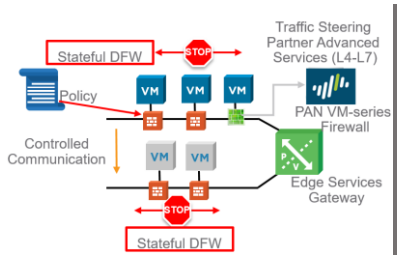
…generating this action log on Panorama…



…resulting in the capture and, in this particular configuration, the action “Alert” which was re-configured to action “Deny” for our actual testing.



Our summary results follow:



3: Distributed Segmentation with Net (VLAN) Isolation and Service Insertion	Recon (nmap)	Recon (aux SMB)	SMB Exploit	JavaARA Exploit	Magento Exploit
Control 3a					
NSX 3b	3b-i	3b-i			3b-ii

Caveats **3b-i)** Use of "Block" and "Reject" create differing DB_NMAP responses. With the Block action, which will "Block silently…", the action permits the DB_NMAP to detect the presence of the target, due to traceroute "hits", although all port scanning fails. Reject actions develop no DB_NMAP or auxiliary SMBversion data.

Caveat **3b-ii)** The Magento Unserialize exploit was not able to generate an action "Alert" nor to "Deny" the transaction, due to a missing signature in the Panorama signature database. Discussions with Panorama engineering confirmed that the signature could be constructed easily and integrated into the PHP detection logic of the firewall. We concluded that the small-business nature of a typical Magento storefront application customer has precluded construction of this exploit signature in the default Panorama packaging.

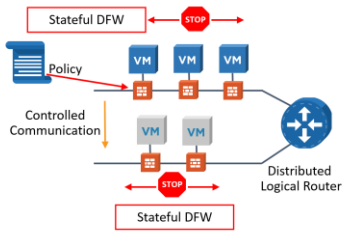
Design Patterns 4a/4b – Distributed Segmentation with Network Overlay Isolation

Use of non-VDI NSX Network overlays for the dVS networks in design patterns 4a/4b mimic the testing performed in the 2a/2b scenario exactly, albeit without the physical server connection. Instead of the Edge Services Gateway, the NSX Logical Router is used.

In the 4b scenario, where NSX DFW rules were enabled, VDI E-W and N-S between App and VDI networks had these firewall restrictions in place:

3	VDI East West	1015	SG-...	SG-...	any	Block	Distr...
4	North South Acc...	1026	SG-...	SG-...	any	Block	Distr...

The results were identical and screenshots are redundant to the 2a/2b testing. Here are the results:



4: Distributed Network with Network Overlay Isolation	Recon (nmap)	Recon (aux SMB)	SMB Exploit	JavaARA Exploit	Magento Exploit
Control 4a					
NSX 4b	4b-i	4b-ii			

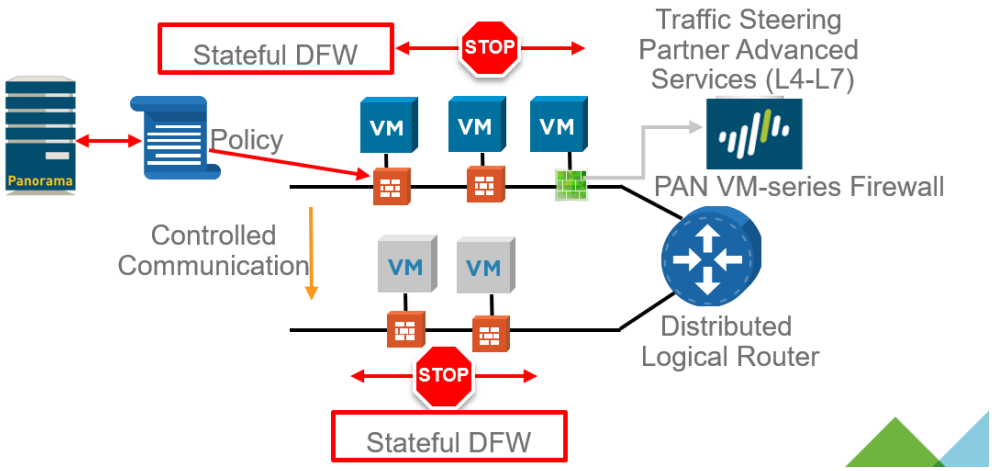
As with the 2a/2b designs, expected NSX protection under 4b scenario was achieved with special comments on the recon activities.

4b-i) Caveat: DB_NMAP activity differs based on the NSX DFW actions of Block versus Reject. The Block action, which will "Block silently…", the action permits the DB_NMAP to detect the presence of the target, due to traceroute "hits", although all port scanning fails.

4b-ii) Caveat: Similar to DB_NMAP the Aux SMB_Version will fail, with no host entries in the MS_DB to begin with. Should a manually prepared host entry be present, both Block and Reject actions allow no refinement of target identity information, and exploitation will fail.

Design Patterns 5a/5b – Distributed Segmentation with Network Overlay Isolation and Service Insertion

Also an analog to pattern 3a/3b, yet using the Network Overlay vDS, patterns 5a/5b re-visit the use of Palo Alto Network's Panorama VM-Series firewall, with identical exploits for JavaARA and Magento, as depicted in this diagram:



Methods and results are identical to our 3a/3b, with slightly different caveats under the Recon sections, as they related to recon across networks at Layer 3 (N-S):

5: Distributed Network with Network Overlay Isolation and Service Insertion	Recon (nmap)	Recon (aux SMB)	SMB Exploit	JavaARA Exploit	Magento Exploit
Control 5a					
NSX 5b	5b-i	5b-ii			5b-iii

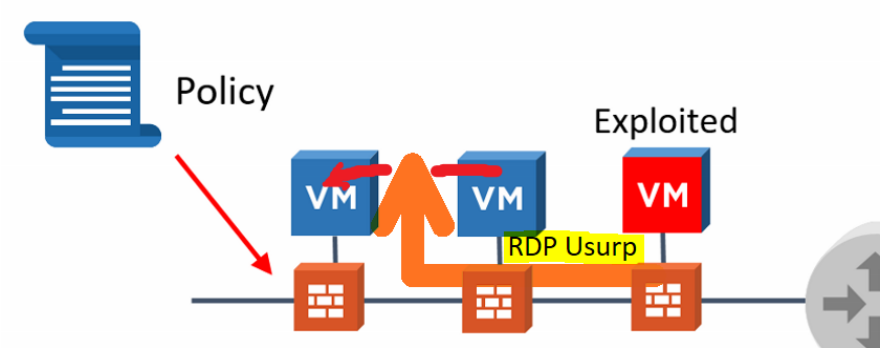
Caveats **5b-i and 5b-ii)** Use of "Block" and "Reject" create differing DB_NMAP response. With the Block action, which will "Block silently…", the action permits the DB_NMAP to detect the presence of the target, due to traceroute "hits", although all port scanning fails. Reject actions develop no DB_NMAP or auxiliary SMBversion data.

Caveat **5b-iii)** The Magento Unserialize exploit was not able to generate an action "Alert" nor to "Deny" the transaction, due to a missing signature in the Panorama signature database. As with 3b-ii, PAN engineering confirmed that the signature could be constructed easily and integrated into the PHP detection logic of the firewall.

Statefulness Validation – Attempt to Usurp MS-RDP Session to Imitate a Man in the Middle (MiM) Attack

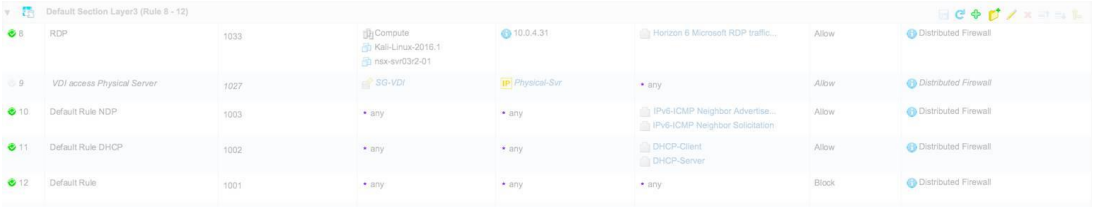
A key feature of the NSX DFW is the capacity to provide for enforcement of packet stateful transactions. NSX's capacity to support enforcement of proper sequencing and state-connectivity for TCP packets that transit in a stateful nature, such as Microsoft Remote Desktop Protocol (MS-RDP using TCP port 3389).

Stateful enforcement can ensure legitimate network flows for critical service protocols similar to MS-RDP, including SSH, TLS, etc.

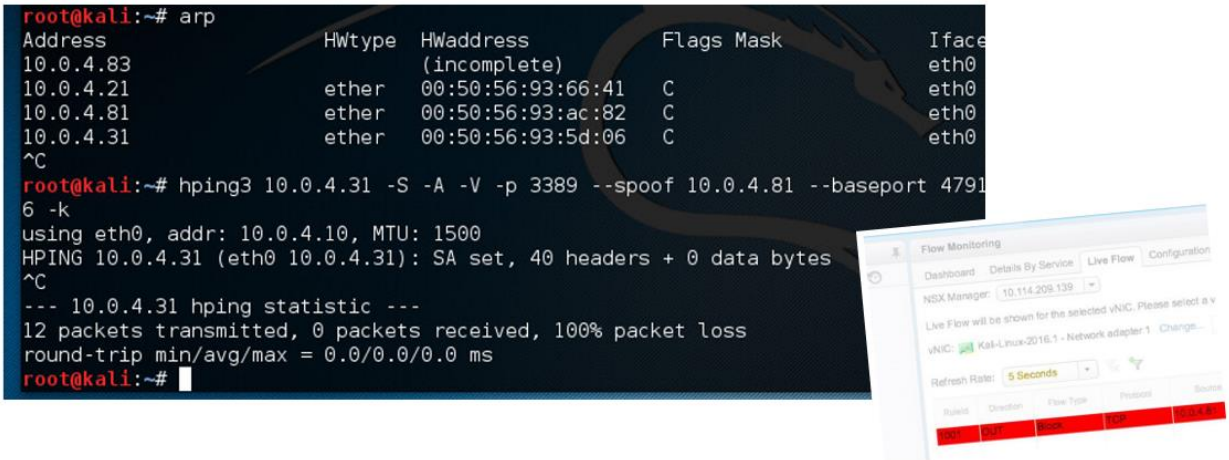


In our test scenario, we attempted to usurp a legitimate RDP session already established between two VMs (-win81-01 and –svr03-01) with our Kali host. We initiated an RDP session from the Windows Server to the console of the Windows 8.1 VM by supplying proper credentials with the Microsoft Remote Desktop application.

Previously, we crafted a rule on the DFW, which would allow E-W transit (Called Horizon 6 Microsoft RDP in our example) and enabled it on the firewall:



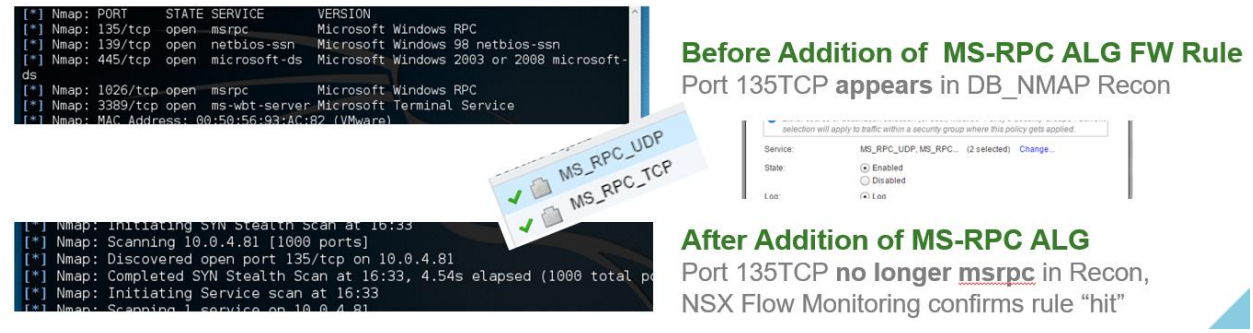
While the RDP session was underway, we attempted to spoof the IP/MAC of the Windows Server and inject traffic on behalf of that machine, in a Man in the Middle (MiM) fashion, with this Linux hping3 command, which failed:



Benchmark of ALG Disruption of Undesired MS-RPC Activity

Application Level Gateways are specialty proxies for a variety of common services to support combined data and control connections and expected stateful order of those transactions for key protocols including: FTP, CIFS, ORACLE TNS, MS-RPC and SUN-RPC.

We validated the NSX DFW with design pattern 1b – Flat Network w/ Physical Router using the following DB_NMAP reconnaissance event, and creation of a firewall rule to Permit MS-RPC traffic (which enables the ALG). Our testing produced this result:



Note in the “Before…” trace, DB_NMAP located “1026/tcp open msrpc Microsoft Windows RPC”, the dynamic port that was created when MS-RPC port 135 was activated during the scan. After the addition of the ALG rule, the dynamic MS-RPC spawned port cannot be detected.

CONCLUSION

Many security products do a terrific job of making promises in their marketing material – only to be found critically lacking when they are actually put into service in a customer's real infrastructure. In the information security space, customers who are looking for the proverbial "silver bullet" are often mistaken when they believe these advertisements.

The underlying purpose of our review of the VMware NSX 6.2.x network virtualization platform was to sort out the facts from the hype, in an actual "micro audit", where we could present representative E-W/N-S threats against typical network topologies (patterns) and scientifically measure the results. As stated in our "Objectives" section, we wanted to know if NSX functionally satisfied: the NIST SP 800-126 recommendations R1-R4; the defined precepts of micro-segmentation; and, could real-world threats be prevented, when launched using actual hacking/penetration testing tools. Here are our findings:

We found VMware NSX did satisfy our three objectives – based on these particulars**:**

- NIST SP 800-125B VM-FW-R1, "In virtualized environments with VMs running delay-sensitive applications, virtual firewalls should be deployed… instead of physical firewalls…" NSX fully satisfies this recommendation.
- NIST SP VM-FW-R2, "…kernel-based virtual firewalls should be deployed instead of subnet-level virtual firewalls…" NSX provides the DFW and ESG components which operate at native hardware speeds, due to their tight integration with the hypervisor.
- NIST SP VM-FW-R3, "…it is preferable if the firewall is integrated with a virtualization management platform…" NSX's intrinsic integration into the vCenter controls and instrumentation and direct RESTful-API and vRealize Operations suite(s) exceeds the intent of this recommendation.
- NIST SP VM-FW-R4, "…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…" VMware NSX is intrinsically policy-based, and through tools like the Service Composer, with its native integration into VMware tags, object names and other rich meta-data are useful to actual management and implementation. NSX is a prime example of how a product could satisfy this recommendation.
- NSX also fully satisfies the five precepts of the specific definition of micro-segmentation: It is distributed, stateful and topology agnostic (with the exception of not operating on all SDNs); it supports centralized, ubiquitous policy control of distributed services; it has granular, unit-level controls, which are implemented by high-level policy objects; embodies a network overlay-based isolation and segmentation model; and has policy-driven service insertion with traffic steering to enable third party solution for L4-L7 firewalling, IDS/IPS and guest introspection.
- In measurement of actual threat vectors, which simulated the three primary attack types (Worm, Browser exploit and application based), the VMware NSX platform did mitigate E-W and N-S attacks when launched by actual Penetration Testing/MetaSploit toolkits.
- In addition to and in conjunction with third-party solutions, specifically the Palo Alto Networks VM-series Panorama firewall, service insertion effectiveness was proven on the JavaARA exploit, providing for detection and prevention of Malware payload delivery in that complex scenario.
- Through the Application Level Gateway (ALG) features of the DFW, we were able to thwart the DB_NMAP reconnaissance by strictly enforcing the transaction rules for the MS-RPC call. This L4 response by the distributed firewall contributes to an improved resilience against E-W and N-S threats.

- The distributed firewall delivered stateful enforcement of proper network flows as demonstrated in our simulated Man in the Middle (MiM) attack using a spoofed MS-RDP example. Although the MS-RDP example tested port 3389 actions, this DFW action can easily be applied to other network flows for other ports and protocols.

Can NSX be used as part of a regulatory compliant infrastructure?

In addition to being asked to review Information Security technical products on a frequent basis, we are often asked about the suitability of those products for use in regulatory compliance environments, such as PCI DSS, HIPAA, FedRAMP, SOC, CJIS, DISA STIG (see: http://ir.vmware.com/overview/press-releases/press-release-details/2016/Newly-Released-STIG-Validates-VMware-NSX-Meets-the-Security-Hardening-Guidance-Required-for-Installment-on-Department-of-Defense-DoD-Networks/default.aspx), etc. Such requests have been made regarding VMware NSX.

Beginning in 2011, VMware and Coalfire have jointly developed and published an overview of how VMware products may be used to become compliant with many of these particular regulations. In the VMware Compliance Reference Architecture Framework (RAF) series of whitepapers (found on VMware Solutions Exchange at https://solutionexchange.vmware.com/store/products/vmware-compliance-cyber-risk-solutions#.VRGBT5PF9Fg), information is available to inform customers about using VMware products to create a "compliant capable" solution for their particular business requirement. The first document in each regulation area is the Product Applicability Guide (PAG), depicting the specific relationship between a VMware product and the regulatory controls used for compliance with that regulation. Additional documents in the RAF series address architecture and validation of particular "reference architecture" examples that may be helpful to the regulatory compliance objectives of the customer.

Although we must always disclaim generic suitability of any product for regulatory compliance (*customers attain compliance through a GRC program, not through the use of a specific products*), we can confirm that a number of high-profile customers have decided to use NSX to assist in delivering such workloads securely.

Coalfire and VMware have a number of customers using NSX as part their GRC program's technical solution. For example, service provider Armor (formerly FireHost) is delivering a multi-tenant hosting solution (PaaS) specifically for PCI DSS customers and will provide customers a PCI DSS Attestation of Compliance (AOC), which includes their network infrastructure that utilizes a micro-segmented firewall service built on VMware NSX. In another example, solution provider Rackspace has a PCI service provider offering, which has been recently profiled in a case study that may be found at: http://blogs.vmware.com/vcloud/2016/09/rackspace-meets-new-pci-dss-compliance-vmware-nsx-network-virtualization.html

Does VMware NSX deliver a useful and relevant cybersecurity benefit to customers?

In our micro-audit study of the effectiveness of NSX to prevent east-west and north-south exploits from real-world threats, we attempt to answer this significant question, and to, in general, conclude if the marketing promises are met by product reality. Our answers follow in the Coalfire Opinion section below.

COALFIRE OPINION

It is the opinion of the authors, as a result of the testing performed on VMware NSX 6.2.4, that the product **does** provide significant cybersecurity benefits in typical and expected customer use-cases, with the following comments, assumptions and expectations:

- Implementation and deployment follows best-practices including use of VMware supplied guidance to properly configure, harden and integrate NSX into the customer's architecture.
- The customer evolves their security model beyond basic perimeter and zone-based security design to incorporate aspects of zero trust into their particular scenario. Specific analysis and evaluation of their use-cases to incorporate E-W network transit needs into their security model is highly recommended.
- When protection beyond ALG, stateful enforcement and basic L4 services provided by NSX components is desirable, third-party service insertion products are used in conjunction with NSX.
- The customer will make use of the NSX Service Composer and well-designed tags, objects and policies to create a specific implementation methodology to ease deployment of the micro-segmentation security model they have developed.
- Periodic penetration testing by internal security teams and third-party scan providers should be used to verify the effectiveness of the NSX deployment.
- Security network training is provided to the VMware systems administration and operations team, as they will be in a position to perform many operations traditionally relegated to the cybersecurity and networking teams.
- Role Based Access Control (RBAC) has been properly configured and is in force at the time of the deployment.
- Other best practices not specifically stated that are customary to "in house" network configuration and management are also employed by the NSX consumer and are aligned with the particulars of their specific Information Security risk management framework and posture. These non-technical issues will undermine any technically excellent deployment.

ABOUT THE AUTHOR AND CONTRIBUTORS

Chris Krueger | **Author** | Principal, Cloud and Virtualization, Coalfire Labs, 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.

Jason Macallister | **Contributor** | Senior Consultant, Coalfire Labs, Coalfire Systems
Mr. Macallister consults on Information Security and regulatory compliance topics as they relate to advanced infrastructure, emerging technology and VMware products.

Dan McInerney | **PenTest Technical** | Consultant, Coalfire Labs, Coalfire Systems
Dan provided essential MetaSploit selections, construction of the Kali Linux test server and consultation to design and deliver the exploited Design Patterns used in this publication.

Wade Holmes | **VMware Project Sponsor** | Sr. Technical Product Manager NSBU, VMware
NSX Product Manager with responsibility for network and security thought leadership.

THE NATION'S CYBER RISK MANAGEMENT AND COMPLIANCE LEADER

With more than 15 years in IT security and compliance and ASV-certified since inception, Coalfire is a leading provider of IT advisory services, helping organizations comply with global financial, government, industry, and healthcare mandates while helping build the IT infrastructure and security systems that will protect their businesses from security breaches and data theft.