Lab 3: Penetration Testing and Intrusion Detection Systems

Josh Lieberg

Submitted To: Tony Wan

Date Submitted: 11/18/2024

Date Due: 11/18/2024

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The objective of this lab was to deploy and test two different intrusion detection systems (IDS). Testing was conducted on new machines by exploiting vulnerabilities in the target systems. A unique aspect of this lab was its two-phase structure. These phases consisted of identifying and exploiting vulnerabilities using tools such as Nmap, Nessus, and Metasploit, followed by the implementation and testing of Snort and Suricata as part of the IDS.

Phase one focused on analyzing and exploiting vulnerabilities in the target virtual machines (VMs). Metasploitable and Ultimate LAMP VMs were deployed as targets, while a newly deployed Kali Linux VM served as the attacking system. This configuration enabled vulnerability scanning and exploitation of the target systems.

Phase two introduced an additional component to the architecture: an intrusion detection system (IDS) implemented via Security Onion, which included Snort and Suricata. This IDS was deployed in network intrusion detection mode to monitor and analyze all traffic in the DMZ. Attacks simulated in phase one were repeated in this phase to assess the IDS's capabilities. As expected, the IDS successfully captured malicious activity in its logs. To optimize monitoring, sensors were strategically placed, and all rule sets were updated to ensure accurate traffic analysis.

This lab demonstrated the importance of both offensive and defensive approaches in cybersecurity. The results showcased the effectiveness of IDS systems in detecting network intrusions and identifying vulnerabilities within the infrastructure. Recommendations and insights gained throughout the lab are detailed in the following report.

BUSINESS CASE

Computer Industries has decided to enhance its security systems by subjecting their own systems to penetration testing and collecting the data from an IDS. The objective was to identify all vulnerabilities on newly implemented virtual machines. These machines were an Ultimate LAMP VM and a Metasploitable VM. By doing this, Computer Industries was able to strengthen the infrastructure's defenses against any threat that could potentially cause harm.

The enhancement was conducted in two distinct phases that were designed to simulate real world attack scenarios. In phase one, titled penetration testing, Computer Industries deployed a Kali Linux VM that acted as the attacking system. The tools deployed on this machine were Nmap, Nessus, and Metasploit. Computer Industries was then able to perform several other operations including operating system (OS) fingerprinting, vulnerability analysis, and exploitation on the Ultimate LAMP and Metasploitable VMs. This phase showed Computer Industries the vulnerabilities the systems possessed and how they could be exploited by real attackers.

Phase two, titled intrusion detection system implementation, introduced Snort and Suricata which are contained in Security Onion. These tools help monitor and analyze traffic within the DMZ. These systems were configured to log and alert malicious activity using the same attacks from the previous phase. To ensure correct functionality, Computer Industries needed to update all rules sets for Snort and Suricata. These updates ensured that the systems possessed robust detection capabilities. A logical diagram including the new additions can be found in the results and one illustrating the previous architecture before the new format was implemented during the enhancement is included below:

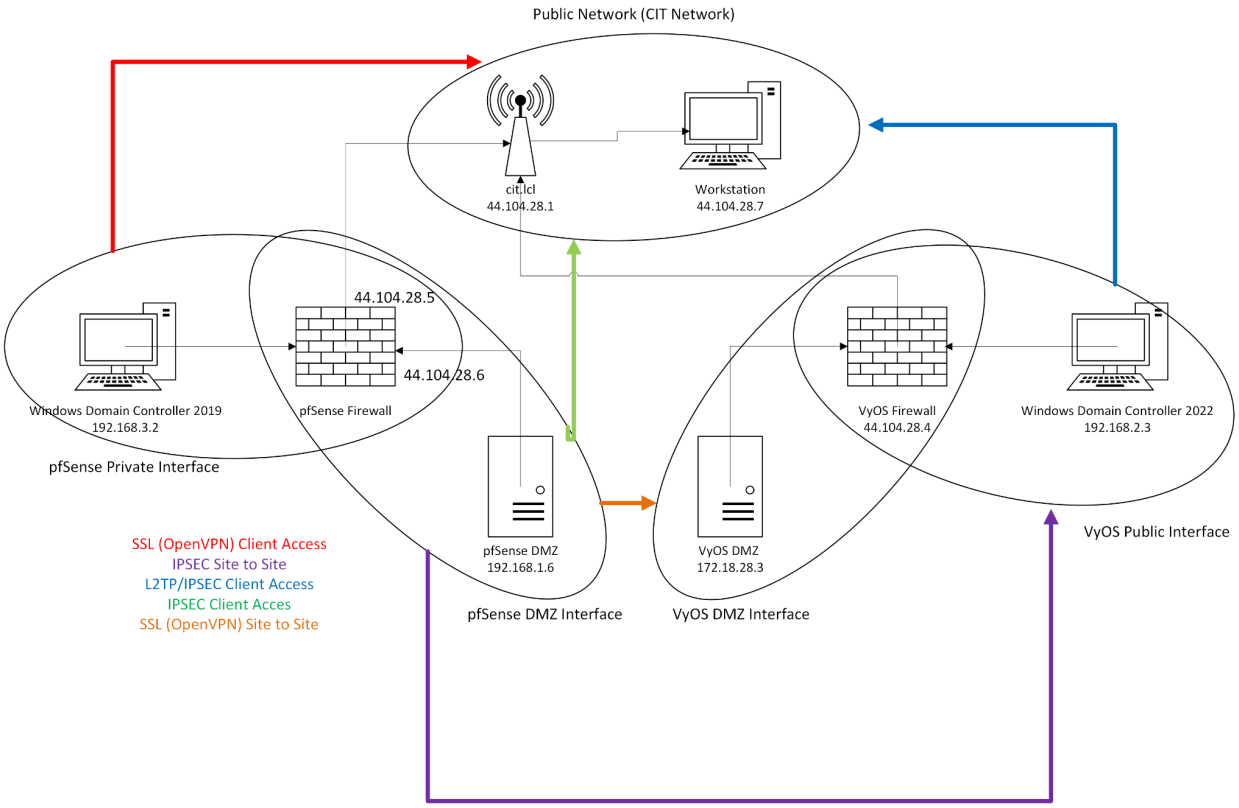

Figure 1: Logical Diagram

With the completion of this enhancement, Computer Industries has gained a comprehensive understanding of the vulnerabilities the infrastructure faces. The company also learned about the effective IDS is at monitoring malicious activity. By integrating IDS tools and performing staged real world attacks, Computer Industries created a new foundation for ongoing system defenses and continuous improvement.

PROCEDURES

This procedure section goes phase by phase for the objectives completed during lab two, Microsoft Windows Administration. The troubleshooting techniques used can be found in Appendix A. In this section, the **buttons** used will be in bold, typed in computer instructions are in `Courier New`, *options* selected/pressed will be italicized, and steps requiring menu navigation will be represented by the pipe symbol. In rare occurrences, there are sometimes *options* interpreted as a **button**, which is represented by both italicized and bolded ***words***.

Table 1. Formatting Key

Representation	Format in Report
Button	**Button**
Options	*Options*
Text Entered in Computer	`Courier Text`
Menu Navigation	*Start* | *Programs* | *MS Office* | *Word*

Deployed Kali OVF

The steps below outline the steps taken to create the Kali Linux virtual machine. This machine will be used to run certain tools within our environment to find and exploit vulnerabilities.

1. Opened *File Explorer* | Typed `\\rtfm\ISO\Linux\Kali\Old` | Copied *kali-linux-2017.3-vm-amd64* to Desktop
2. Navigated to https://studentvc.cit.lcl/ | Right clicked on *CNIT455G28* | *Deploy OVF Template…* | *Local File* | *Upload Files…* | *kali-linux-2017.3-vm-amd64* | **Open** | Named the VM `cnit45500.g28.KaliOVF` | **Next** | *stvmsrv22haas.cit.lcl* | **Next** | **Next** | *Thin Provision* | *Haas Storage Cluster* | **Next** | **Next** | **Finish**

Installed Nessus and run with nmap

Nessus is a tool used to scan either hosts or entire networks for vulnerabilities. Another tool, nmap, is also able to scan either hosts or networks for potential vulnerabilities such as open ports. These steps outline how both tools were used in the environment of our business.

1. Opened Kali VM
2. Navigated to www.tenable.com/dowloads
3. Selected the most recent version of Nessus
4. Selected **Download**
5. Set up an email to create an account
6. Entered a username and password to create an admin account
7. Logged into the web UI of Nessus
8. Selected *Scans* | *Create New Scan*

9. Set `Net Scan` for the name
10. Set `192.168.1.0/24` as the target of the scan
11. Selected the scan that was just created
12. Selected **Launch**
13. Opened a terminal
14. Typed `nmap 192.168.1.0/24` to scan the network using nmap

Deployed Metasploitable OVF

The Metasploitable machine was created with pre-existing vulnerabilities. By using this machine as a target, logs are able to be generated through the network traffic and sent to the Security Onion management console to be viewed.

1. Opened File Explorer
2. Then typed `\\rtfm\ISO\Metasploitable\Metasploitable2 OVF`
3. Copied all files within this folder to the desktop
4. Navigated to https://studentvc.cit.lcl/ | Right clicked on CNIT455G28 | Deploy OVF Template… | Local File | Upload Files…
5. Selected all files that were downloaded from the Metasploitable folder within rtfm
6. Selected **Next** | Named the VM `CNIT455.G28.Metasploitable` | **Next** | *Haas Storage Cluster* | **Next** | *stvmsrv22.cit.lcl* | **Next** | **Next** | *Thin Provision* | *8 GB Ram* | *40 GB Hard Disk* | **Next** | **Next** | **Finish**

Deployed Ultimate Lamp OVF

The Ultimate Lamp machine was created with pre-existing vulnerabilities. By using this machine as a target, logs are able to be generated through the network traffic and sent to the Security Onion management console to be viewed.

1. Opened File Explorer
2. Then typed `\\rtfm\ISO\Ultimate Lamp\UltimateLAMP-OVF`
3. Copied all files within this folder to the desktop
4. Navigated to https://studentvc.cit.lcl/ | Right clicked on *CNIT455G28* | *Deploy OVF Template…* | *Local File* | *Upload Files…*
5. Selected all files that were downloaded from the Metasploitable folder within rtfm
6. Selected **Next** | Named the VM `CNIT455.G28.UltimateLamp` | **Next** | *Haas Storage Cluster* | **Next** | *stvmsrv22.cit.lcl* | **Next** | **Next** | *Thin Provision* | *8 GB Ram* | *40 GB Hard Disk* | **Next** | **Next** | **Finish**

Created IDS

The IDS installed within our environment is Security Onion 2. Within Security Onion natively is Suricata. Snort community rules were also added into the management console, both providing different logs of the attacks that took place on the network.

1. Navigated to https://studentvc.cit.lcl
2. Right clicked *CNIT455G28* and selected *Create a new virtual machine* | **Next**
3. Set the name for the machine as `CNIT455.G28.SecOnion` | **Next**
4. Selected *stvmsrv22haas.cit.lcl* as the compute resource | **Next**
5. Selected the *Haas Storage Cluster* as the storage for the virtual machine | **Next** | **Next**

6. Set the Guest OS Family to *Other* and the Guest OS Version to *Other (64 bit)*, then selected **Next**
7. Selected *CD/DVD Drive* | *rtfm.iso* | *Security Onion* | *securityonion-2.4.110-20241010.iso* | **OK**
8. Set 4 CPU Cores, 24 GB of RAM, and 200 GB of hard disk
9. Thin provisioned the virtual machine
10. Added an additional NIC and set both NICs to DMZ-A port group
11. Selected **Next** and then **Finish** to create the VM

Configured the IDS

Before being able to be used as an IDS, the Security Onion machine was configured with two NICs. This allowed for certain machines to access the web user interface to view logs created as well as receive network traffic over its monitoring interface.

1. Opened Security Onion 2 machine that was created
2. Selected *Install Security Onion 2.4.110* | `Yes`
3. Entered a username and password for the machine
4. Pressed **Enter** to reboot the machine then logged in with the created username and password
5. Selected *Yes* | *Install* | *STANDALONE*
6. Typed `AGREE` | *Standard* | **Enter** | **Enter** | **Enter** | **Enter** | **Enter**
7. Set the IP address as `192.168.1.69/24` and `192.168.1.1` as the gateway address
8. Entered `44.2.1.44` and `44.2.1.45` for the DNS servers then pressed **Enter** | **Enter** | Enter

9. Selected the other NIC as the monitor interface | **Enter**
10. Entered group28@acme.com for the email address and set a password | **Enter** | **Enter**
11. Set the allowed IP range to view the web UI as `192.168.1.0/24` | **Enter** | **Enter** | **Tab** | **Enter**
12. After the configuration of the machine was set, ran `sudo soup` command to install all updates

Added Snort Rules to Security Onion

Through the addition of Snort Community rules, an extra layer of logs was created when attacks were generated. These rules allow for a more granular view of the attacks that took place.

1. Opened DMZ-A Alma Linux machine
2. Navigated to https://192.168.1.69
3. Entered the login information created during the setup process
4. Migrated to *Administration* and then *Configuration*
5. Clicked *Options* | *Show Advanced Settings* | *customRulesets* | *soc* | *config* | *server* | *modules* | *suricataengine* | *customrulesets*
6. In the Current Grid Value box, added the following ruleset:

   ```
   {"community":true,"license":GPLv2","ruleset":"snort-community","target-file":"community.rules","url":"https://www.snort.org/downloads/community/community-rules.tar.gz"}
   ```

7. Selected **Options** | **Synchronize Grid** to update the ruleset
8. Suricata was already installed in the initial creation of Security Onion

Created Port Mirroring Session

In order for the IDS to obtain logs of the network traffic, a monitoring session was created. This ensures that all traffic that flowed over the DMZ network segment was logged in the IDS and viewable by network security administrators.

1. Migrated to https://studentvc.cit.lcl
2. Navigated to the Switch section of the cluster
3. Selected the distributed switch for our group number
4. Selected *Configure* | *Port Mirroring* | *New…* | *Distributed Port Mirroring* | *Next* | *Next*
5. Chose the target machines as sources for the session, then clicked Next
6. Chose the Security Onion machine as the destination for the session, then clicked **Next** | **Finish**
7. In the port group section, selected *DMZ-A* | *Policies* | *Edit* | *Security*
8. Changed the *Promiscuous mode* setting from *Reject* to *Accept* and clicked **OK**

Conducted Metasploitable Exploit

An exploit of the Metasploitable machine was researched and conducted over the network. The exploit used was a vsftp backdoor. This allowed for the testing of the IDS to ensure that logs were created.

1. Opened a terminal within the Kali VM
2. Typed `msfconsole`
3. Typed `search vsftpd`
4. Entered `use exploit/unix/ftp/vsftpd_234_backdoor`
5. Entered `RHOST 192.168.1.25`

6. Typed `exploit`
7. Migrated to the Security Onion Console (SOC) on the DMZ-A Alma machine
8. Selected *Detections* | *Options* | *Suricata* | *Full Update*
9. After the update, looked through logs to see both suricata and snort detections from the attack

Conducted Ultimate Lamp Exploit

An exploit of the Ultimate Lamp machine was researched and conducted over the network. The exploit used was a denial of service attack. This allowed for the testing of the IDS to ensure that logs were created.

1. Opened a terminal within the Kali VM
2. Typed `nmap script dOS 192.168.1.200 -sV`
3. Migrated to the Security Onion Console (SOC) on the DMZ-A Alma machine
4. Selected *Detections* | *Options* | *Suricata* | *Full Update*
5. After the update, looked through logs to see both suricata and snort detections from the attack

RESULTS

The team at Computer Industries has successfully conducted penetration testing on their own systems. Data was retrieved by the IDS machine, which proves to be fully functional. The objective here was to ensure our systems were secure and that the IDS machine would see malicious traffic on the company's network. Building off of the preexisting architecture, the pfSense machine sits in front of all of the new machines added in this phase, which include a Kali Linux machine, Metasploitable machine, Ultimate LAMP machine, and a Security Onion machine which acts as the IDS. The upgrade was split into a two-phase enhancement to its infrastructure, which again was focused on penetration testing and the deployment of intrusion detection systems (IDS). This initiative provided invaluable insights into the network's vulnerabilities and strengthened its defenses against real-world attack scenarios.

The first phase focused on identifying and exploiting vulnerabilities within the network. A Kali Linux virtual machine was deployed as the attacking system, equipped with tools like Nmap and Nessus. These tools facilitated operating system fingerprinting, vulnerability analysis, and exploitation on target systems such as Metasploitable and Ultimate LAMP. By simulating real-world attack scenarios, the team gained critical insights into the network's weaknesses and the potential methods attackers could employ to compromise it.

Building on the knowledge gained in Phase One, Phase Two introduced Snort and Suricata—which are both popular IDS solutions. These tools were deployed using Security Onion and updated with the latest rule sets to ensure proper detection capabilities. Placed within the DMZ, the IDS sensors were configured to monitor mirrored traffic from the target systems.

During this phase, attacks similar to those performed in Phase One were launched against the targets to evaluate the effectiveness of the IDS solutions. The systems successfully logged and alerted on malicious activity, which showed their ability to detect and analyze network threats. Figure 2 below illustrates the new network architecture implemented at Computer Industries after the completion of the two phase enhancement project. In this network architecture, the target machines all sit in the DMZ underneath the pfSense firewall. The Kali Linux sits outside of the DMZ but can still attack the target machines.

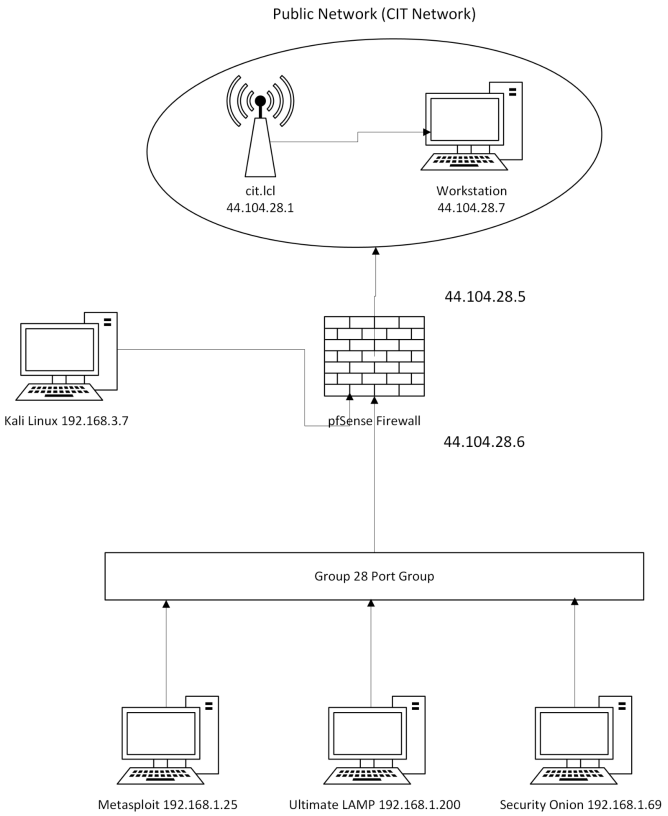

Figure 2: Logical Diagram

CONCLUSIONS AND RECOMMENDATIONS

All objectives in this enhancement project were successfully achieved. During phase one, the Kali Linux VM was able to exploit both the Ultimate LAMP and Metasploitable VMs. This showcased how vulnerable Computer Industries infrastructure was and the immediate need for this project. In phase two, the implementation of Security Onion with Suricata and Snort allowed for all malicious and non malicious activities to be detected and logged. Seeing all the logged activity in the infrastructure showed Computer Industries the enhancement was successful and all machines were properly set up and fulfilled their intended purposes.

The preestablished pfSense firewall was correctly configured to allow the new operations occurring in the network. These changes ensured that ICMP echo requests were blocked. Additionally, all external access points were sealed off ensuring that no unauthorized external access was allowed. Combining the success attacks on vulnerabilities with the functional IDS highlighted Computer Industries successes. It also displayed the importance of a robustly configured network and how an IDS makes an infrastructure proactive instead of reactive.

Recommendations

Recommendation One: Keep Snort and Suricata Together

Due to the volatility of the kernel the VMs run on, it is recommended to keep Snort and Suricata together so resources can be optimized to the highest potential. Creating two VMs that serve the same purpose would double the CPU, memory, and storage so it is recommended to keep them together. This will also reduce costs in a real world environment as there is no need for additional physical or virtual infrastructures.

Recommendation Two: Take Multiple Snapshots

Since attacking is part of the lab, it is recommended to prepare for troubleshooting. Taking proactive measures, such as snapshots, will ensure data is not lost if errors occur when making significant changes. Taking a snapshot allows a VM to be quickly restored to that state it was in when the snapshot was taken. By utilizing this feature provided in vSphere, troubleshooting can be streamlined since progress can always be restored.

Recommendation Three: Test In Between Phases

Testing in the environment is critical as it ensures the phases are completed successfully. If phase one is set up incorrectly and not functioning appropriately, issues will without a doubt appear in phase two. Making sure phase one is one hundred percent sound before moving on to phase two is of utmost importance. Completing the prior phase creates a solid foundation that Security Onion can mount itself to and render itself useful. Trying to bypass portions of phase one and go to phase two will result in more errors that are difficult to troubleshoot.

BIBLIOGRAPHY

Cisco. (n.d.). Snort rules and IDS software download. https://www.snort.org/downloads/#rule-downloads

CIT (n.d.). *Lab Report Template*

Configuration—Interface Configuration | pfSense Documentation. (n.d.). Retrieved from https://docs.netgate.com/pfsense/en/latest/config/interface-configuration.html

Download Suricata Suricata. https://suricata.io/download/

Hadden, J. (n.d.). Exploit vulnerabilities of lamp based web applications in ... https://ptolemy.berkeley.edu/projects/truststc/education/reu/10/Papers/HaddenJ_paper.pdf

Installation. Installation - Security Onion Documentation 2.4 documentation. (n.d.). https://docs.securityonion.net/en/2.4/installation.html

LAMP – AMPPS. (n.d.). AMPPS. Retrieved from https://ampps.com/lamp/

Metasploitable 2 exploitability guide. Metasploitable 2 Exploitability Guide | Metasploit Documentation. (n.d.). https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/

Metasploit | Penetration Testing Software, Pen Testing Security. (n.d.). *Metasploit.* Retrieved from https://www.metasploit.com/

Mitchel, J., & Novotny, J. (2024, May 9). *How to Emulate a SYN Flood Attack With Kali Linux*. Linode. Retrieved November 18, 2024, from https://www.linode.com/docs/guides/emulate-syn-flood-attack-with-kali-linux/

Rawles, P (n.d.). *CNIT 45500 Lab 3 - Penetration Testing and Intrusion Detection Systems*

Security Onion Solutions. (n.d.). *SOS.* Retrieved from https://securityonionsolutions.com/software

slowhttptest. (n.d.). Kali Linux. Retrieved November 18, 2024, from https://www.kali.org/tools/slowhttptest/

Snort 3 and Security Onion 2. https://blog.securityonion.net/2021/02/snort-3-and-security-onion-2.html

APPENDIX A: PROBLEM SOLVING

Problem 1: KaliOVF Apt issues

Problem Description: After importing a Kali Linux OVF file into a virtualized environment, the system encountered issues with the apt-get command as well as web browser resolving. Specifically, attempts to run apt-get update resulted in errors, while certain web pages resolved but displayed a blank white screen. These issues prevented the installation of necessary tools such as Nessus.

Possible Solutions: One potential solution was to verify and ensure correct network settings on the virtual machine and retest network connectivity. This could involve checking the adapter type on vSphere and the network configuration inside the Kali VM. Another solution involved testing DNS resolution to confirm DNS was working properly. A third solution was to investigate proxy or firewall settings that might be interfering with internet access or blocking specific ports required by apt-get.

Attempted Solutions: The first solution was attempted by restarting the network service on the Kali VM. This approach did not resolve the issue, as the errors with apt-get persisted. The second solution focused on DNS troubleshooting, including testing connectivity to public DNS servers like Google's 8.8.4.4 using ping. The problem still persisted. The third solution involved checking the firewall configurations on both the host machine and the virtual machine. Nothing was found to be out of the ordinary.

Final Solution: Unfortunately, there was no final solution to this problem. Further investigation is required to identify any underlying issues, including a potential review of logs for any misread error messages, testing with different configurations, or seeking external assistance for more

advanced troubleshooting. A brand new KaliVM was deployed onto the architecture and worked fine.

Problem 2: Security Onion Initial Internet Connectivity

Problem Description: After configuring the Security Onion instance on the architecture, the system could not access the internet to perform updates to be fully functional. Despite completing the install, the device could not reach the public internet.

Possible Solutions: One potential solution was to verify the network configuration on the Security Onion instance, ensuring it had a valid gateway and DNS settings for outbound connectivity. Misconfigured network settings could block internet access. Another solution was to check the pfSense firewall rules and NAT settings. If outbound NAT was not configured correctly, the Security Onion instance might not be able to route traffic to the internet. A final solution involved verifying that no proxy settings or advanced filtering rules on the machine were blocking internet access.

Attempted Solutions: The first solution was attempted by reviewing and manually reconfiguring the Security Onion instance's network interface, including assigning a valid gateway and DNS server. While the configuration appeared correct, this did not resolve the internet connectivity issue. The second solution involved testing with a default allow-all rule on the pfSense firewall for troubleshooting purposes. However, the problem still persisted. The third solution focused on reviewing the pfSense firewall’s rules and settings. Outbound NAT rules were initially absent for the Security Onion instance, so NAT was manually configured to allow the instance’s traffic to pass through the firewall to the internet.

Final Solution: The issue was resolved by simply implementing an outbound NAT rule on the pfSense firewall for the Security Onion instance. This allowed the system to translate its private

IP to the public-facing IP of the firewall, enabling internet access for updates and other necessary operations. After applying this NAT rule, the Security Onion instance successfully connected to the internet, and updates were performed without issue.

Problem 3: Security Onion Disrupted Internet

Problem Description: A few days after resolving a previous internet connectivity issue on the Security Onion instance by configuring NAT on the pfSense firewall, the system experienced another problem where internet access dropped unexpectedly. Despite the NAT rule being intact, the instance was unable to reach external repositories or perform updates.

Possible Solutions: One potential solution was to restart the network services on the Security Onion instance again, as there might be a simple glitch in the network stack which could cause connectivity issues. Another solution was to verify the network adapter settings in the virtual machine to ensure they were still correctly configured and functioning. Virtual network adapters can sometimes encounter issues that disrupt connectivity. A final solution involved reviewing the firewall logs on pfSense to determine whether traffic from the Security Onion instance was being blocked or misrouted, even with the NAT rule in place.

Attempted Solutions: The first solution was attempted by restarting the network service on the Security Onion instance by running `sudo systemctl restart networking`. However, this did not resolve the issue, and the system remained unable to access the internet. The second solution involved replacing the NAT rule that was configured above. After replacing the NAT rule on pfSense, the issue still persisted. The third solution focused on replacing the virtual network adapter entirely. The existing adapter was removed from the VM, and a new adapter was added with the same IP address configuration as the previous one.

Final Solution: The internet connectivity issue was resolved by removing the network adapter from the Security Onion virtual machine and replacing it with a new one while keeping the same IP address as before. Doing so resolved the problem, which suggests that the old network adapter may have encountered an issue within the virtualized environment. Once replaced, the Security Onion instance regained stable internet access and was able to perform updates successfully.

APPENDIX B: VIRTUAL MACHINE NETWORK SETTINGS CONFIGURATION

Appendix B gives the IP Address, Subnet mask, Default Gateway, Preferred DNS, Alternate DNS settings, and Domain Controller assignments for each VM computer. Appendix B also gives the IP address and Subnet mask of the interfaces attached to pfSense.

Table 1: Virtual Machine Networking Configuration Settings

PC/Interface	IP address	Subnet Mask	Gateway	Pref. DNS	Alt. DNS
pfDMZ	192.168.1.6	255.255.255.0	192.168.1.1	44.2.1.44	44.2.1.45
pfSense	44.104.28.5	255.255.255.0	192.168.1.1	192.168.1.5	192.168.2.2
privA	192.168.3.2	255.255.255.0	192.168.3.1	127.0.0.1	
privB	192.168.2.3	255.255.255.0	192.168.3.1	44.2.1.44	44.2.1.45
VyOS	44.104.28.4	255.255.255.0	192.168.3.1	44.2.1.44	44.2.1.45
VYosDMZ	172.18.28.3	255.255.255.0	172.18.28.1	44.2.1.44	44.2.1.45
WINDOWSPublic	44.104.28.7	255.255.255.0	192.168.3.1	44.2.1.44	44.2.1.45
Metasploit	192.168.1.25	255.255.255.0	192.168.1.1	44.2.1.44	44.2.1.45
KaliVM	192.168.3.7	255.255.255.0	192.168.3.1	44.2.1.44	44.2.1.45

UltimateLAMP	192.168.1.200	255.255.255.0	192.168.1.1	44.2.1.44	44.2.1.45
SecOnion	192.168.1.69	255.255.255.0	192.168.1.1	44.2.1.44	44.2.1.45
Windows10	192.168.5.3	255.255.255.0	192.168.5.1	44.2.1.44	44.2.1.45

APPENDIX C: PFSENSE FIREWALL RULE CONFIGURATION

Table 1: WAN Interface Rules

Action	Protocol	Source	Port	Destination	Port	Gateway
Pass	IPv4 ICMP	*	*	*	*	*
Pass	IPv4 TCP/UDP	*	*	44.104.28.6	80	*
Pass	IPv4 TCP/UDP	*	*	44.104.28.6	20-21	*
Pass	IPv4 TCP/UDP	*	*	WAN address	80	*
Pass	IPv4 TCP/UDP	*	*	192.168.1.6	80	*
Pass	IPv4 TCP/UDP	*	*	192.168.1.6	20-21	*
Block	IPv4 *	*	*	*	*	*

Table 2: OpenVPN

Action	Protocol	Source	Port	Destination	Port	Gateway
Pass	IPv4 *	*	*	*	*	*

Table 3: OPT1 Interface Rules

Action	Protocol	Source	Port	Destination	Port	Gateway
Pass	*	*	*	OPT Address		*
Pass	IPv4 TCP	OPT1	*	LAN	*	*
Pass	IPv4 *	OPT1	*	*	*	*
Pass	IPv4 TCP/UDP	*	*	OPT1	21000-21999	*
Pass	IPv4 TCP/UDP	OPT1	*	*	53	*
Pass	IPv4 TCP/UDP	OPT1	*	*	80	*
Pass	IPv4 TCP/UDP	OPT1	*	*	443	*

Pass	IPv4 TCP/UDP	OPT1	*	*	20-21	*
Pass	IPv4 TCP/UDP	OPT1	*	*	*	*
Pass	IPv4 TCP/UDP	OPT1	*	*	123	*
Pass	IPv4 TCP/UDP	OPT1	*	*	25	*
Block	IPv4 *	*	*	*	*	*

Table 4: LAN Interface Rules

Action	Protocol	Source	Port	Destination	Port	Gateway
Pass	IPv4 *	*	*	LAN	*	*
Pass	IPv4 TCP	*	*	*	*	*
Pass	IPv4 *	*	*	*	*	*
Pass	IPv4 *	PRIVATE	*	*	*	*
Pass	IPv6 *	PRIVATE	*	*	*	*
Block	IPv4 *	*	*	*	*	*

Table 5: NAT Outbound Rules

Interface	Source	Port	Destination	Port	NAT Address	Port
WAN	192.168.1.6/32	*	*	*	44.104.28.6/32	*
WAN	192.168.3.0/24	*	WAN	*	WAN	*
WAN	192.168.3.0/24	*	*	*	44.104.28.6	*
WAN	192.168.1.0/24	*	WAN	*	WAN	*
WAN	192.168.1.0/24	*	*	*	44.104.28.6	*

Table 2: IPsec

Action	Protocol	Source	Port	Destination	Port	Gateway

Pass	IPv4 *	*	*	*	*	*