Danny Choi

CYSE 301: Cybersecurity Technique and Operations

Assignment 3: Sword vs. Shield

In this assignment, you will act as an attacker to identify the vulnerabilities in the LAN network and a defender to apply proper countermeasures. You need to provide a screenshot for each task below.

Task A: Sword - Network Scanning (20+ 20 = 40 points) Power on the listed VMs and complete the following steps from the **External Kali** (you can use either nmap or zenmap to complete the assignment)

- External Kali
- pfSense
- Ubuntu
- Windows Server 2022

Make sure you didn't add/delete any firewall policy before continuing.

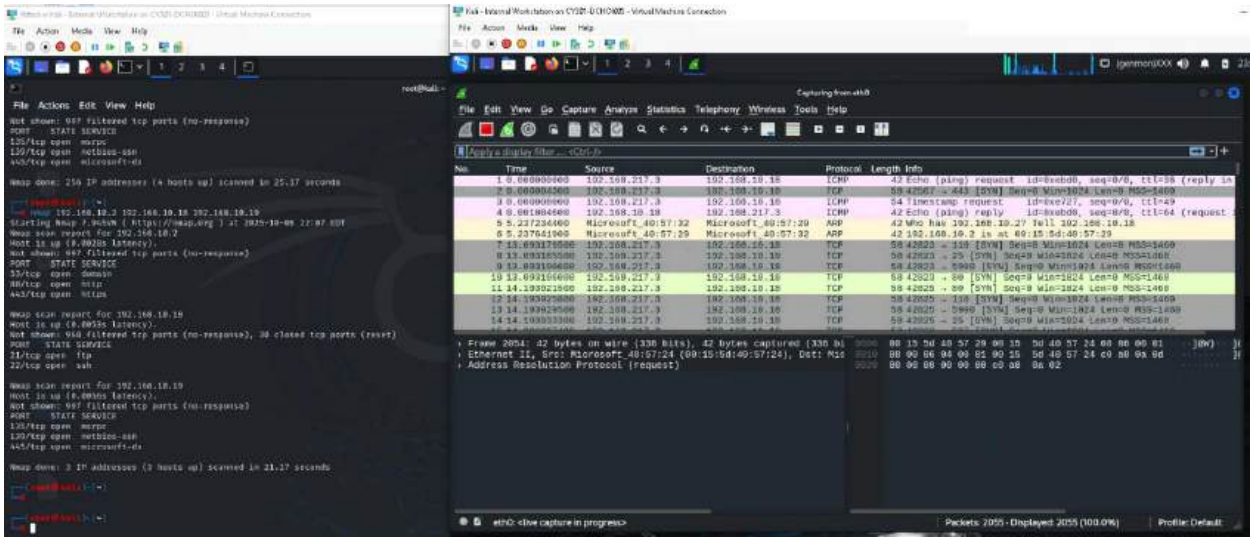
1. Use Nmap to profile the basic information about the **subnet** topology (including open ports information, operation systems, etc.) You need to get the **service** and **backend software** information associated with each opening port in each VM.

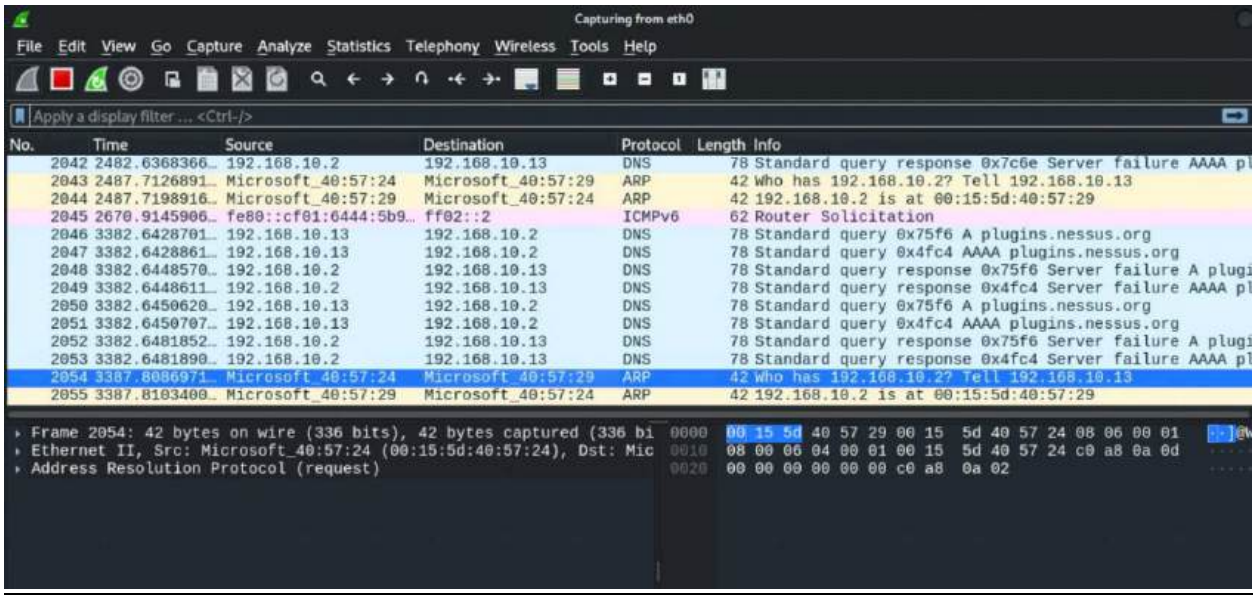
Used Nmap to scan subnet 192.168.10.0/24

```
└─# nmap 192.168.10.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-10-06 21:46 EDT
Nmap scan report for 192.168.10.2
Host is up (0.0031s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE
53/tcp  open  domain
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.10.18
Host is up (0.010s latency).
Not shown: 968 filtered tcp ports (no-response), 30 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh

Nmap scan report for 192.168.10.19
Host is up (0.0045s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Nmap done: 256 IP addresses (3 hosts up) scanned in 25.14 seconds
```

Ran nmap -Sv -p (and the ports from above) to get the service name and backend software/version for the 3 IPs.

```
┌──(root㉿kali)-[~]
└─# nmap -sV -p 21,22,53,80,135,139,443,445 192.168.10.2 192.168.10.13 192.168.10.18
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-10-06 23:17 EDT
Nmap scan report for 192.168.10.2
Host is up (0.0096s latency).

PORT    STATE    SERVICE      VERSION
21/tcp  filtered ftp
22/tcp  filtered ssh
53/tcp  open     domain       (generic dns response: REFUSED)
80/tcp  open     http         nginx
135/tcp filtered msrpc
139/tcp filtered netbios-ssn
443/tcp open     ssl/http     nginx
445/tcp filtered microsoft-ds
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-serv
ice :
SF-Port53-TCP:V=7.94SVN%I=7%D=10/6%Time=68E48654%P=x86_64-pc-linux-gnu%r(D
SF:NSVersionBindReqTCP,E,"\0\x0c\0\x06\x81\x05\0\0\0\0\0\0\0\0");

Nmap scan report for 192.168.10.13
Host is up (0.011s latency).

PORT    STATE  SERVICE      VERSION
21/tcp  closed ftp
22/tcp  closed ssh
53/tcp  closed domain
80/tcp  closed http
135/tcp closed msrpc
139/tcp closed netbios-ssn
443/tcp closed https
445/tcp closed microsoft-ds

Nmap scan report for 192.168.10.18
Host is up (0.0069s latency).

PORT    STATE    SERVICE      VERSION
21/tcp  open     ftp          vsftpd 3.0.5
22/tcp  open     ssh          OpenSSH 8.9p1 Ubuntu 3ubuntu0.7 (Ubuntu Linux; protocol 2.0)
53/tcp  filtered domain
80/tcp  filtered http
135/tcp filtered msrpc
139/tcp filtered netbios-ssn
443/tcp filtered https
445/tcp filtered microsoft-ds
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 3 IP addresses (3 hosts up) scanned in 30.71 seconds
```





2. Run Wireshark in Internal Kali VM while External Kali is scanning the network. Discuss the traffic pattern you observed. What do you find? **Please write a 200-word essay to discuss your findings.**

I ran Nmap, and it returned 3 IP addresses, which makes sense as I had Ubuntu, pfSense, and Windows Server 2022 VMs running (excluding the Internal and Attacker Kali VMs). I proceeded to have those 3 specific IPs on Nmap again while running Wireshark on Internal Kali. I found that the only IP that responded was 192.168.10.18 (Ubuntu), and the other 2 did not, suggesting that the other VM IPs did not respond most likely due to their firewall configurations blocking or filtering incoming probes.

While running Wireshark during the network scan, the patterns that I observed were the ICMP pings from External Kali (192.168.217.3) to the different VMs to check if the host is alive. Followed by the TCP handshake to common ports like 21, 22, 80, 443, with ports responding with SYN/ACK packets indicating that those ports are open, or RST suggesting that they were closed. ARP request was also seen as the host, attempted to discover the MAC addresses to be able to communicate and send packets on the LAN. The Nmap scan result highlights the observed patterns, as the target IPs of the VM, showcased the different ports open as well as other ports. I was able to observe various methods such as ARP request, ICMP ping requests from source/destination IPs and the sending of different packets like TCP/UDP to see how Nmap probes the network. These traffic patterns demonstrates the method that Nmap utilizes to map out the topography discovering live hosts and available services. These scans are crucial as they demonstrate the need for understanding network structure to prevent or better our security posture as well as understanding potential vulnerabilities.

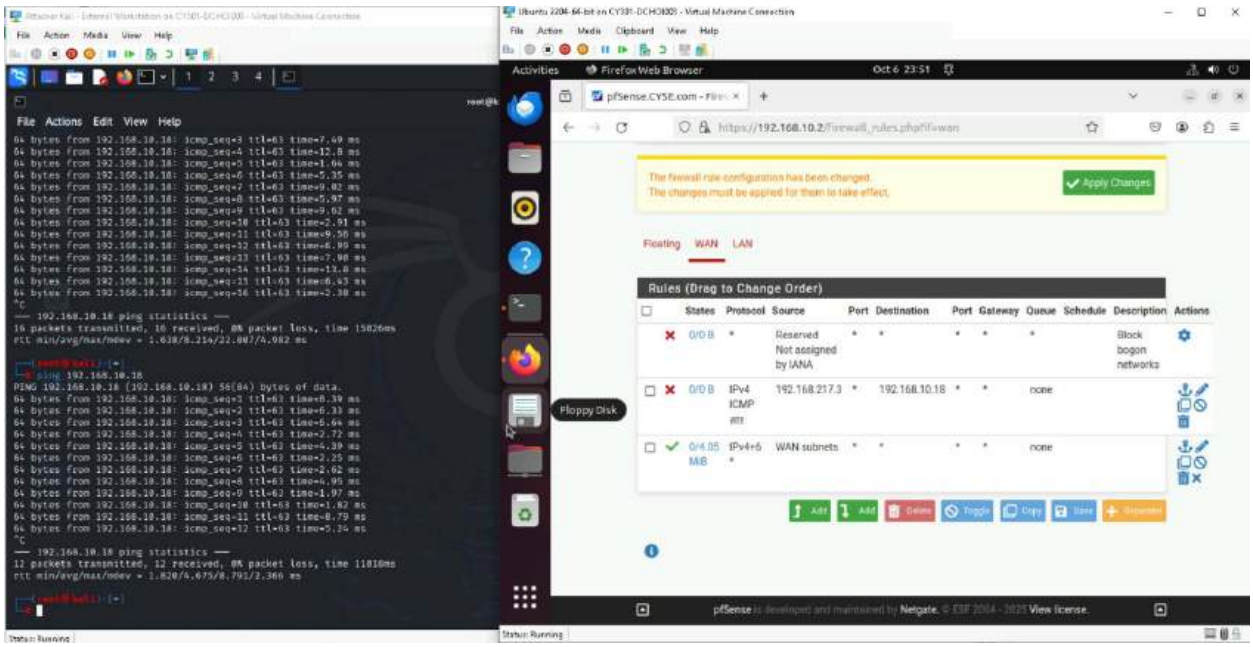
Task B: Shield – Protect your network with a firewall (10 + 10+ 20 + 20 = 60 points)

In order to receive full credits, you need to fill the table (add more rows if needed), implement the firewall rule(s), show me the screenshot of your firewall table, and verify the results.

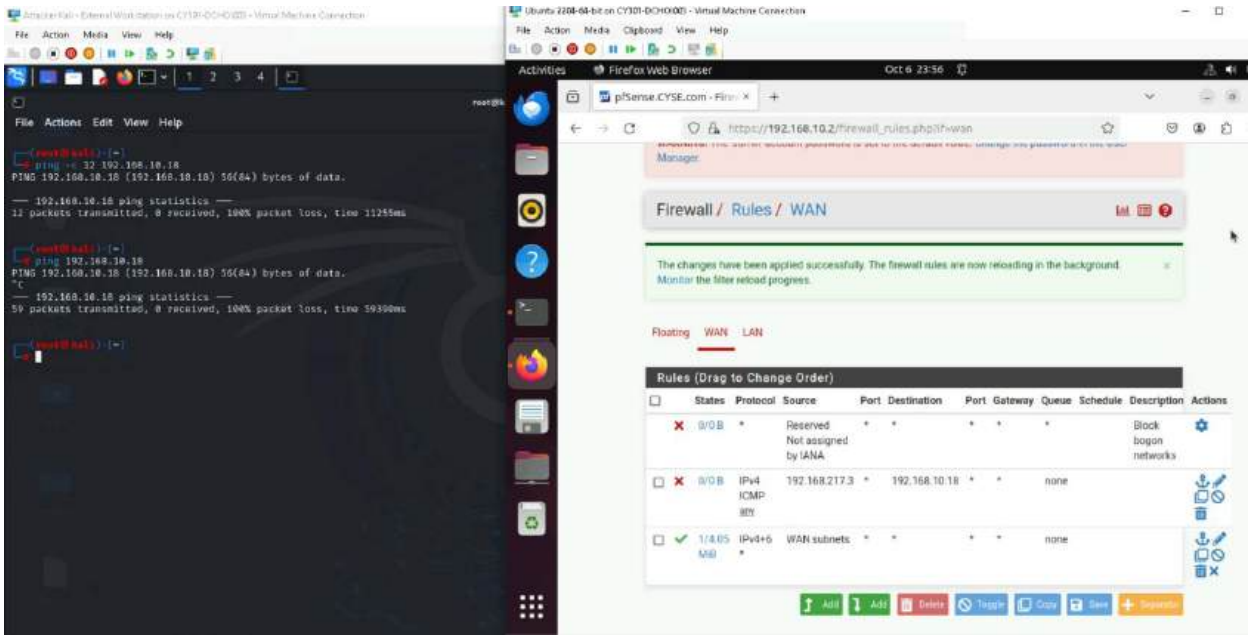
1. Configure the pfSense firewall rule to block the ICMP traffic from External Kali to Ubuntu VM.

Rule #	Direction	Interface	Action	Source IP	Destination IP	Protocol (port # if appliable)
2	Inbound	WAN	Block	192.168.217.3	192.168.10.18	ICMP

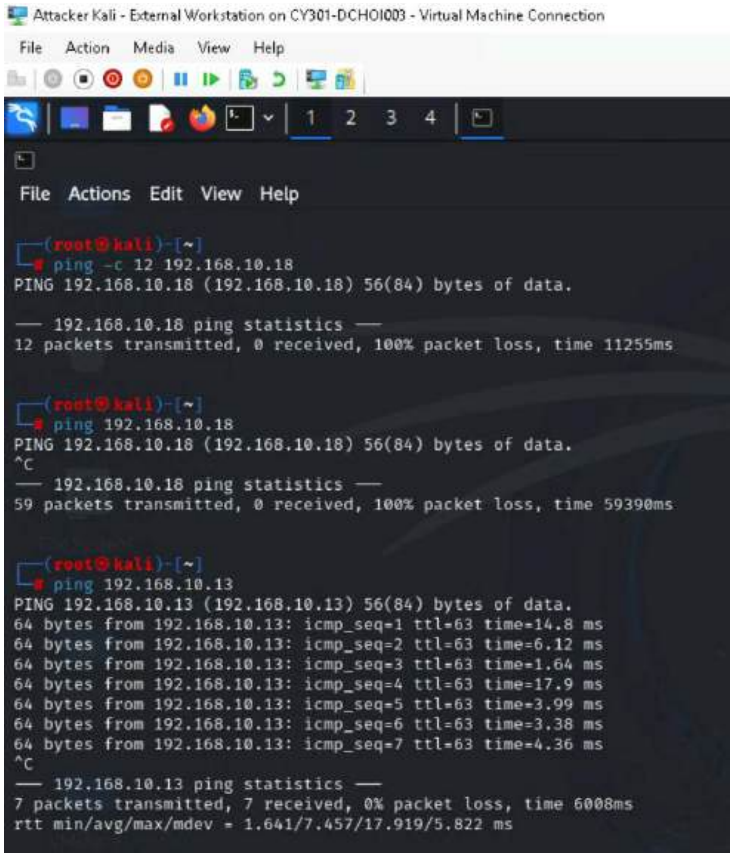
Before applying configuration:



After applying configuration:



Pinged internal Kali, to test that only Ubuntu was blocked. Forgot to close out internal Kali from previous section, and didn't realize that it seems like only 4 VMs are able to run concurrently. Windows Server 2022 closed and Internal Kali remained opened. Cleared the rule before testing, so ended up testing Internal Kali.

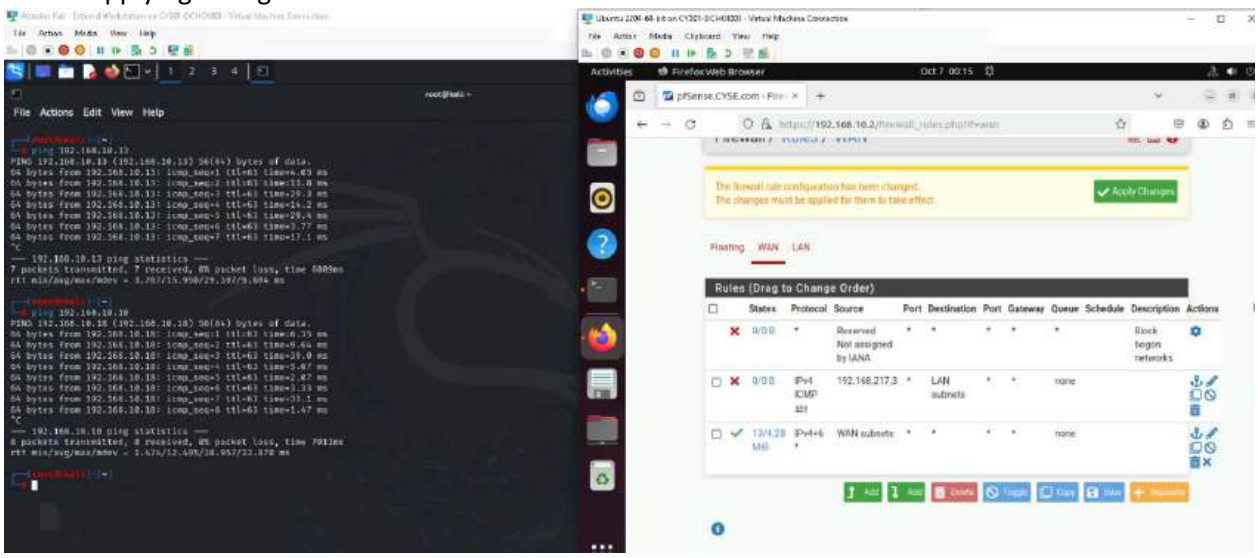


2. Clear the previous firewall policies and configure the pfSense firewall to block all ICMP traffic from External Kali to the LAN side.

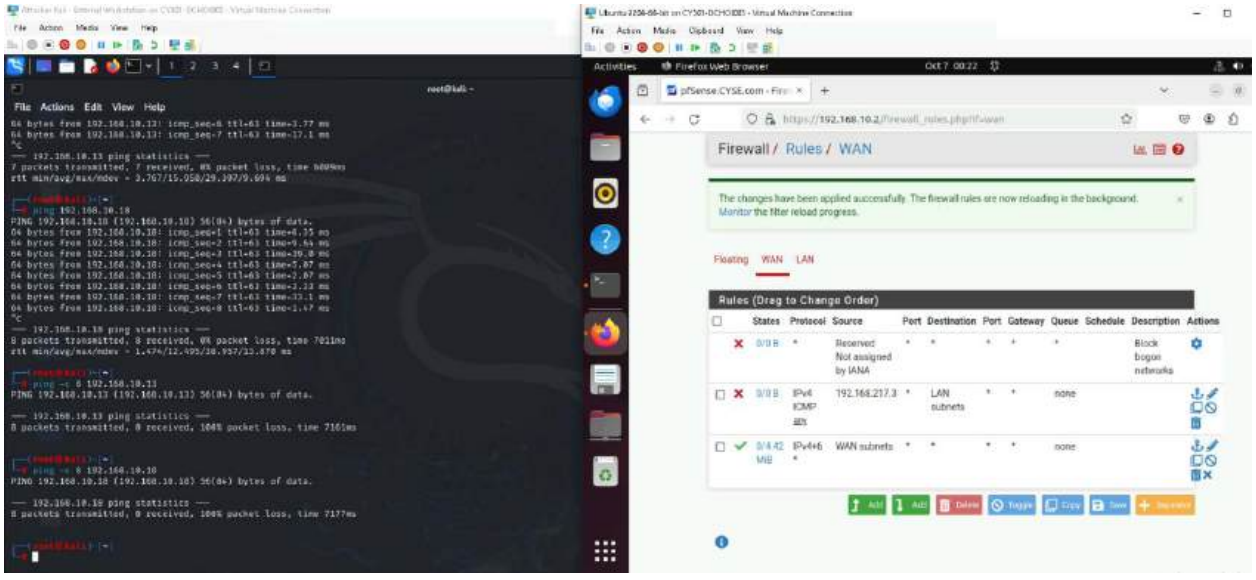
Rule #	Direction	Interface	Action	Source IP	Destination IP	Protocol (port # if appliable)
2	Inbound	WAN	Block	192.168.217.3	LAN Subnet	ICMP

For Destination IP, the IP is 192.168.10.0/24.

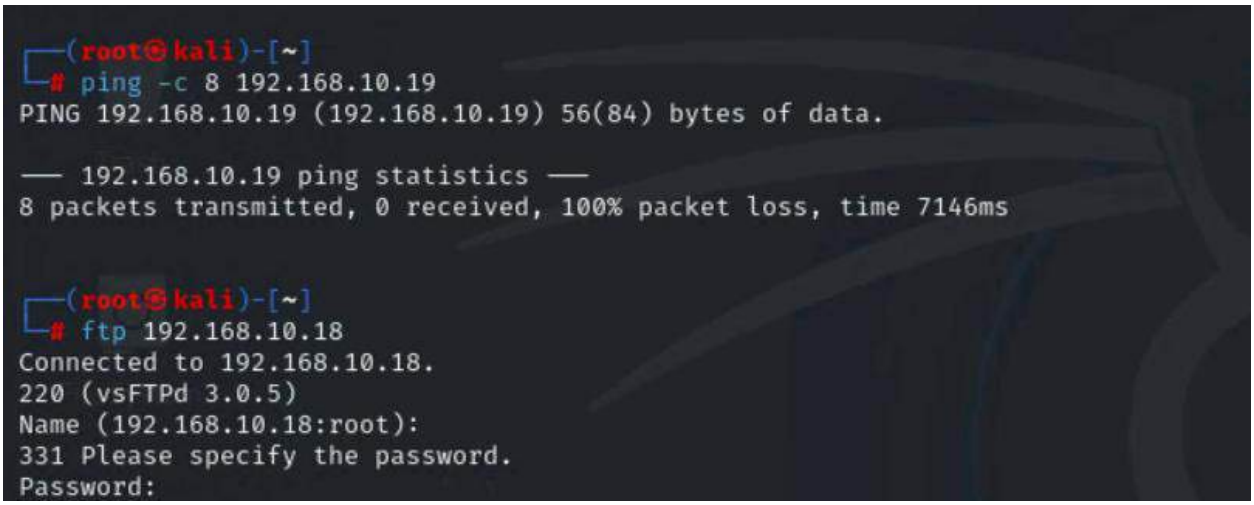
Before applying configuration:



After applying configuration:



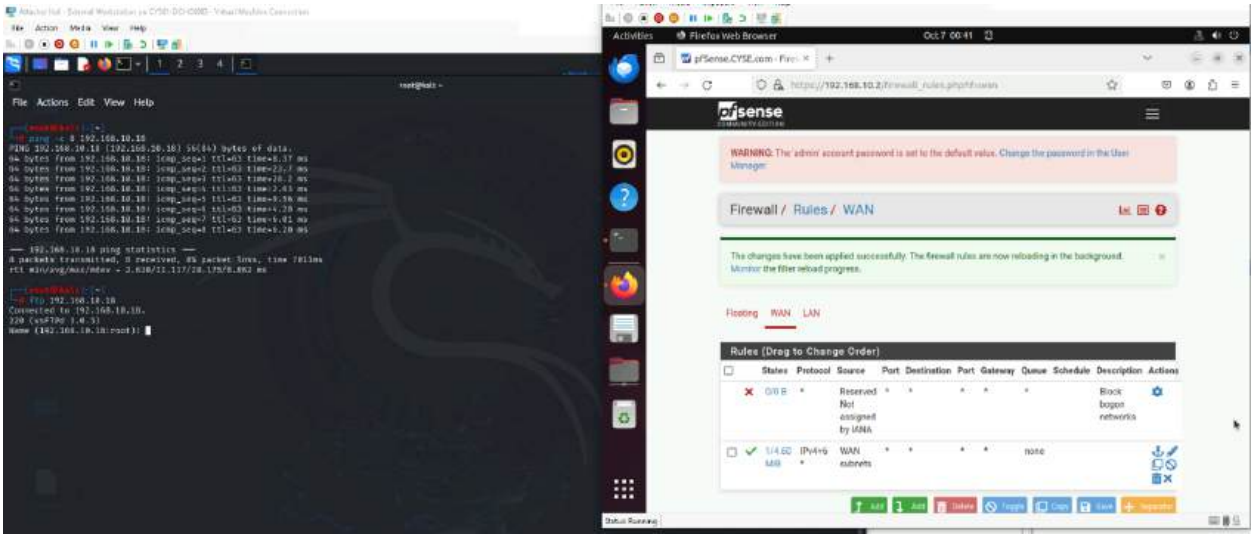
Tried pinging Windows Server 2022 to no success. Was able to establish a connection to Ubuntu FTP.



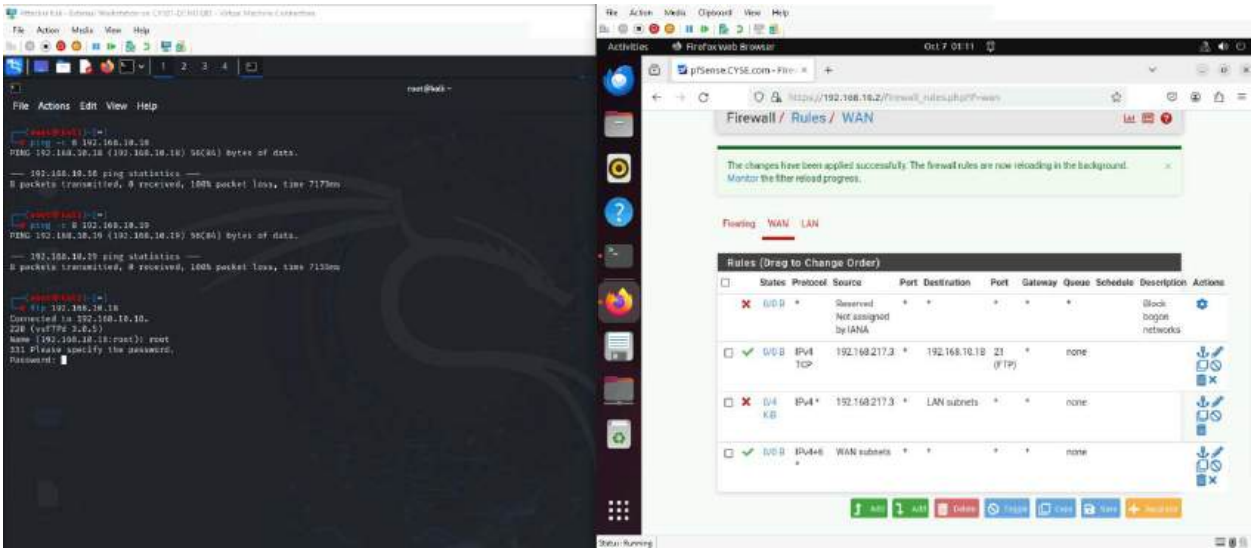
3. Clear the previous firewall policies and configure the pfSense firewall to block ALL traffic from External Kali to the LAN side, except for the FTP protocol towards Ubuntu.

Rule #	Direction	Interface	Action	Source IP	Destination IP	Protocol (port # if appliable)	Port #
2	Inbound	WAN	Pass	192.168.217.3	192.168.10.18	TCP	21
3	Inbound	WAN	Block	192.168.217.3	LAN Subnet	ANY	N/A

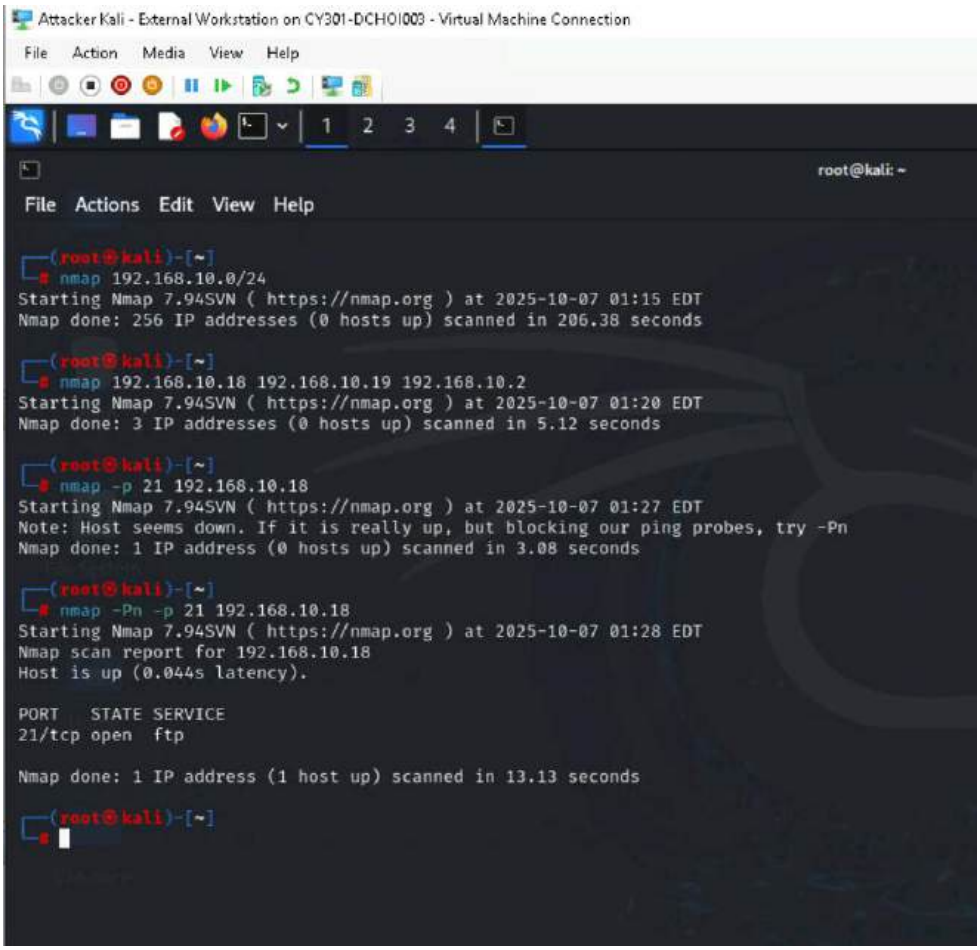
Before Configurations: Able to ping Ubuntu and connect to FTP.



After configurations: Pinged Windows Server and Ubuntu to no success, but was able to connect to FTP.

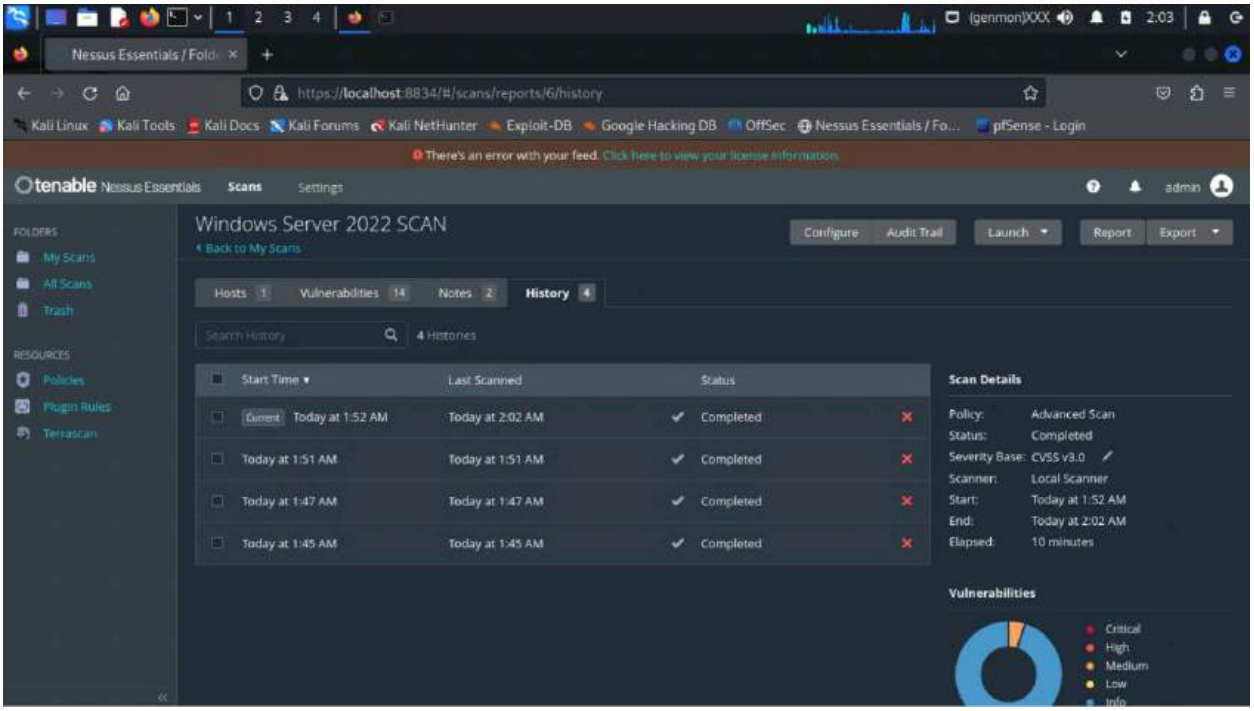


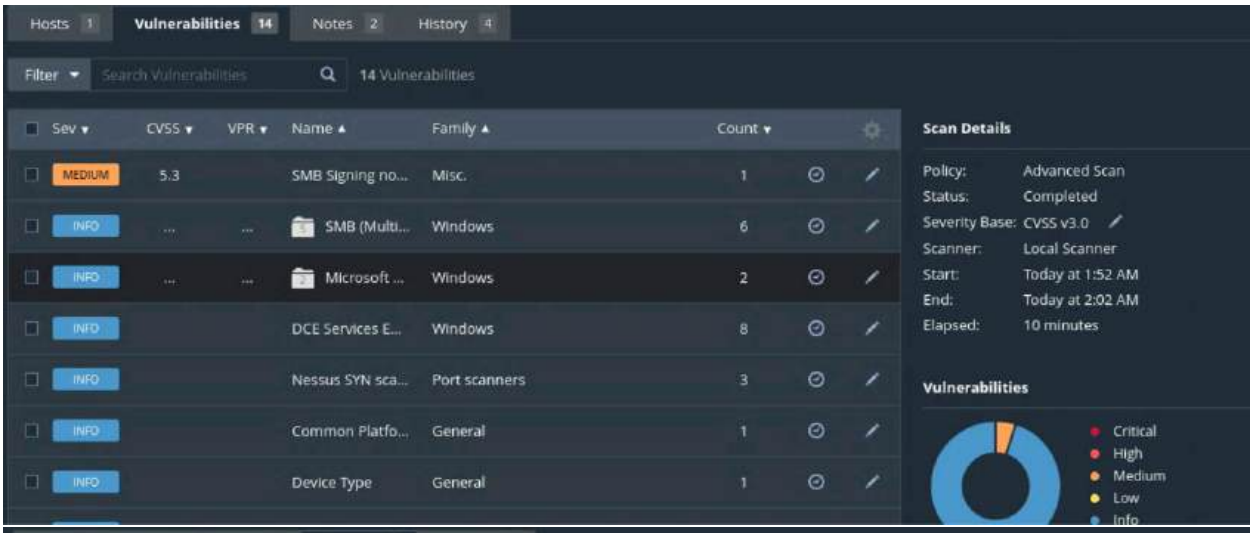
4. <u>Keep the firewall policies you created in Task B.3 and repeat Task A.1. What's the difference?</u>

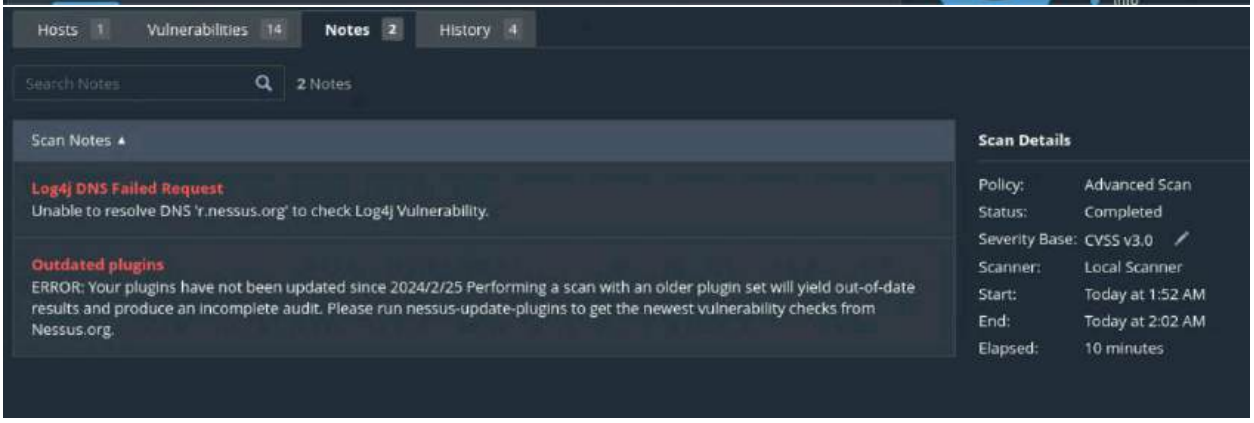


When conducting a repeat of Task A.1, it showed all the hosts as being down, which makes sense due to our firewall rule number 3. This simply means that no responses were received from the host. I wanted to ensure that FTP was up and had an isolated test done. The issue revolves around the fact that FTP is a TCP-based protocol, and typically, for pings, it uses ICMP, and Nmap uses ICMP by default. I narrowed it down so Nmap, when conducting its scan, would probe the default port protocol, which in this case is TCP. The main difference between before and after adding the configuration to pfSense was that the host was up prior to adding the configuration and down afterward. This just demonstrates pfSense working as intended to.

Extra credit (15 points): Use NESSUS to enumerate the security vulnerabilities of Microsoft Windows Server 2022 VM in the CCIA network.




Hosts 1
Vulnerabilities 14
Notes 2
History 4
Filter
Search Vulnerabilities
14 Vulnerabilities
Sev
CVSS
VPR
Name
Family
Count
MEDIUM 5.3 SMB Signing no... Misc. 1
INFO ... ... SMB (Multi... Windows 6
INFO ... ... Microsoft ... Windows 2
INFO DCE Services E... Windows 8
INFO Nessus SYN sca... Port scanners 3
INFO Common Platfo... General 1
INFO Device Type General 1
Scan Details
Policy: Advanced Scan
Status: Completed
Severity Base: CVSS v3.0
Scanner: Local Scanner
Start: Today at 1:52 AM
End: Today at 2:02 AM
Elapsed: 10 minutes
Vulnerabilities
Critical
High
Medium
Low
Info


Hosts 1
Vulnerabilities 14
Notes 2
History 4
Search Notes
2 Notes
Scan Notes
Log4j DNS Failed Request
Unable to resolve DNS 'r.nessus.org' to check Log4j Vulnerability.
Outdated plugins
ERROR: Your plugins have not been updated since 2024/2/25 Performing a scan with an older plugin set will yield out-of-date results and produce an incomplete audit. Please run nessus-update-plugins to get the newest vulnerability checks from Nessus.org.
Scan Details
Policy: Advanced Scan
Status: Completed
Severity Base: CVSS v3.0
Scanner: Local Scanner
Start: Today at 1:52 AM
End: Today at 2:02 AM
Elapsed: 10 minutes