CYSE 301: Cybersecurity Technique and Operations

Assignment 3: Sword vs. Shield

By: George Smith

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.

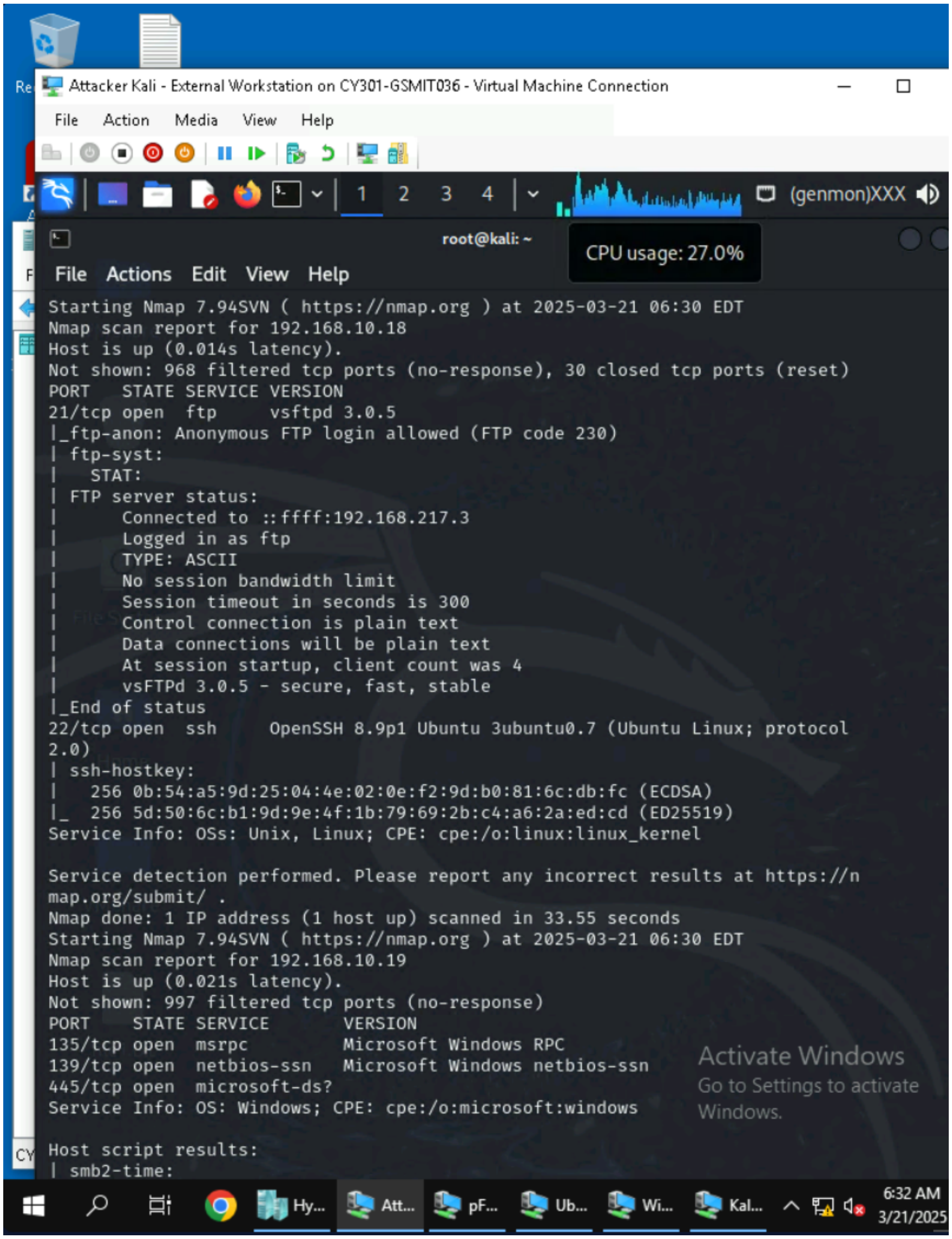
Task A: Sword - Network Scanning (20+ 20 = 40 points)

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.


Attacker Kali - External Workstation on CY301-GSMIT036 - Virtual Machine Connection
File Action Media View Help
1 2 3 4
(genmon)XXX
root@kali: ~
CPU usage: 27.0%
File Actions Edit View Help
Activate Windows
Go to Settings to activate Windows.
Hy... Att... pF... Ub... Wi... Kal...
6:32 AM
3/21/2025

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-21 06:30 EDT
Nmap scan report for 192.168.10.18
Host is up (0.014s latency).
Not shown: 968 filtered tcp ports (no-response), 30 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.5
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.217.3
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 4
|      vsFTPd 3.0.5 - secure, fast, stable
|_End of status
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.7 (Ubuntu Linux; protocol
2.0)
| ssh-hostkey:
|   256 0b:54:a5:9d:25:04:4e:02:0e:f2:9d:b0:81:6c:db:fc (ECDSA)
|_  256 5d:50:6c:b1:9d:9e:4f:1b:79:69:2b:c4:a6:2a:ed:cd (ED25519)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 33.55 seconds
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-21 06:30 EDT
Nmap scan report for 192.168.10.19
Host is up (0.021s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds?
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
```

2. Run Wireshark in Internal Kali VM while External Kali is scanning the network. Discuss the traffic pattern you observed. What do you find? <u>**Please write a 200-word essay to discuss your findings.**</u>
   - **First due to the nature of the type of packets sent by Nmap we get a lot of ACK packets. The reason we get a lot of these ACK packets is because the Nmap type of scan we selected was a SYN version scan using defaulted scripts to get the other information. The SYN creates a three-way handshake and when the computer ACK it Nmap sends back that the port is open. Looking at the protocol type that is sent the most which is TCP it is because the Nmap scan we use goes through the TCP protocol. Some other common themes are that all ACK lengths are**

stable at 54 bytes. Diving a little deeper inside the TCP Protocol section we can see that the checksum values are unverified. This is probably due to the fact that just making the connection is all Nmap needs to spit back information for its results. As stated, before just making the handshake allows the user to know that the port is open. This is also confirmed in the conversation completeness section in the same area as before. Lastly, something I didn't notice until now is the multiple SMB2 packets to try and keep alive the message. I'm thinking this is in response to the incomplete conversion that reaches its TTL limit and before they kill it off, they try to reach one last response.

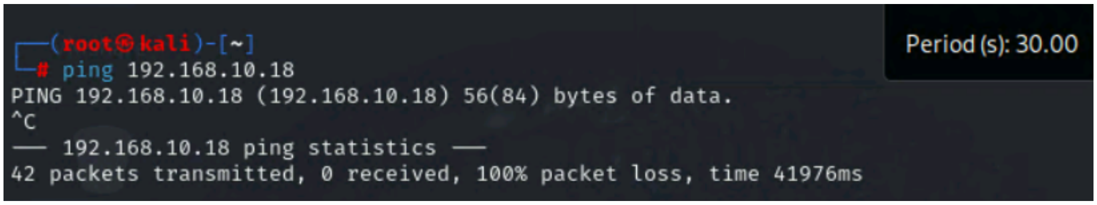
Task B: Shield – Protect your network with a firewall (10 + 10+ 20 + 20 = 60 points)

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 #	Interface	Action	Source IP	Destination IP	Protocol (port # if appliable)
1	WAN	Block	192.168.217.3	192.168.10.18	IMCP

[Add the screenshot here]

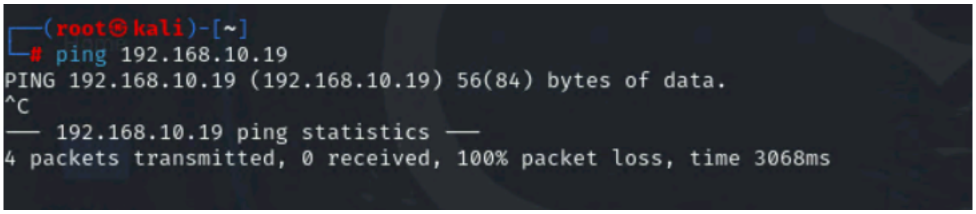


```
┌──(root㉿kali)-[~]
└─# ping 192.168.10.18
PING 192.168.10.18 (192.168.10.18) 56(84) bytes of data.
^C
── 192.168.10.18 ping statistics ──
42 packets transmitted, 0 received, 100% packet loss, time 41976ms
```

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

Rule #	Interface	Action	Source IP	Destination IP	Protocol (port # if appliable)
1	LAN	Block	192.168.217.3	Lan network	ICMP

[Add the screenshot here]



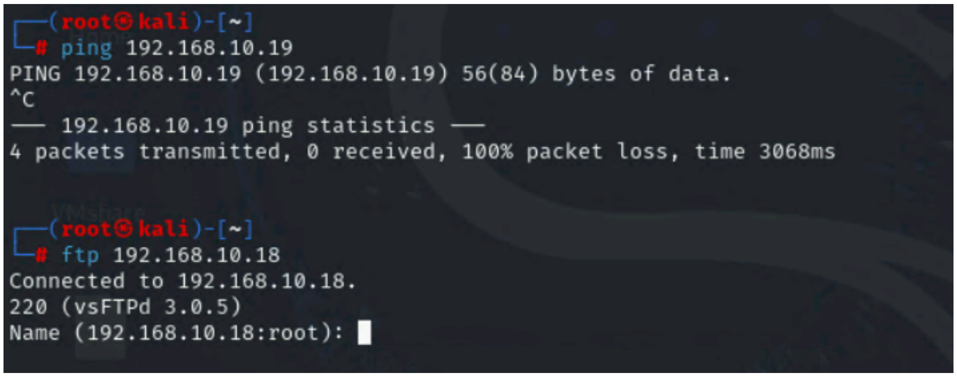
```
┌──(root㉿kali)-[~]
└─# ping 192.168.10.19
PING 192.168.10.19 (192.168.10.19) 56(84) bytes of data.
^C
── 192.168.10.19 ping statistics ──
4 packets transmitted, 0 received, 100% packet loss, time 3068ms
```

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 #	Interface	Action	Source IP	Destination IP	Protocol (port # if appliable)

2	LAN	Block	192.168.217.3	Lan Subnets	Any

[Add the screenshot here]



```
┌──(root㉿kali)-[~]
└─# ping 192.168.10.19
PING 192.168.10.19 (192.168.10.19) 56(84) bytes of data.
^C
── 192.168.10.19 ping statistics ──
4 packets transmitted, 0 received, 100% packet loss, time 3068ms


┌──(root㉿kali)-[~]
└─# ftp 192.168.10.18
Connected to 192.168.10.18.
220 (vsFTPd 3.0.5)
Name (192.168.10.18:root): █
```

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

It gives us a host that seems down instead of giving us information.