Q1. Notice the open ports on all 3 devices (the attacker notebook, the target notebook, and the target Linux VM). Does anything look suspicious, i.e., some ports that you are not aware of that are open on the VM or on your notebooks?

Answer:

```
ATTACKER_IP (macos-host) = 192.168.1.47
TARGET_HOST_IP (window-host) = 192.168.1.37
TARGET_VM_IP (linux-ubuntu-vm-on-window) = 192.168.1.38
```

- **Attacker notebook (Mac, 192.168.1.47):** Open ports 53, 5000, 7000, 7070. Ports 5000 and 7000 are Apple AirPlay services, expected on macOS. Port 7070 belongs to AnyDesk remote desktop, this is notable, since it exposes a remote access service. Port 53 is unusual for a client device and may be a local resolver.

- **Target host notebook (Windows, 192.168.1.37):** Nmap reported the host as down, likely due to Windows Firewall blocking incoming probes. No open ports were detected.

```
# Nmap 7.98 scan initiated Mon Sep  8 18:19:01 2025 as: nmap -T4 -A -v -oN scan-target-host.txt 192.168.1.37

Nmap scan report for 192.168.1.37 [host down]

Read data files from: /opt/homebrew/bin/../share/nmap

# Nmap done at Mon Sep  8 18:19:03 2025 -- 1 IP address (0 hosts up) scanned in 2.17 seconds
```

- **Target Linux VM (192.168.1.38):** Open ports 22 (SSH) and 80 (HTTP). These match the services we explicitly installed for the activity. No suspicious extra services were found.

```
# Nmap 7.98 scan initiated Mon Sep  8 18:09:34 2025 as: nmap -T4 -A -v -oN scan-target-vm.txt 192.168.1.38

Nmap scan report for 192.168.1.38

Host is up (0.020s latency).

Not shown: 998 closed tcp ports (conn-refused)

PORT   STATE SERVICE VERSION

22/tcp open  ssh     OpenSSH 9.6p1 Ubuntu 3ubuntu13.13 (Ubuntu Linux; protocol 2.0)

| ssh-hostkey:

|   256 d1:b4:25:c8:2c:2d:d3:a5:de:ad:59:d3:94:b5:6b:2e (ECDSA)

|_  256 b7:66:6b:b9:6a:3a:7a:de:70:47:26:fe:aa:27:32:fe (ED25519)

80/tcp open  http    Apache httpd 2.4.58 ((Ubuntu))

|_http-title: Apache2 Ubuntu Default Page: It works

|_http-server-header: Apache/2.4.58 (Ubuntu)

| http-methods:

|_  Supported Methods: OPTIONS HEAD GET POST

Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
Read data files from: /opt/homebrew/bin/../share/nmap

Service detection performed. Please report any incorrect results at https://
nmap.org/submit/ .

# Nmap done at Mon Sep  8 18:09:46 2025 -- 1 IP address (1 host up) scanned in
11.71 seconds
```

Conclusion: The Linux VM is clean and as expected, while the attacker Mac reveals AirPlay services (normal) and AnyDesk (potentially sensitive). The Windows host blocked the scan, so results are inconclusive but likely firewall behavior.¶

Q2. Look at the information provided by nmap about your OS's on all 3 devices. Is the information correct? Why is it or why is it not correct?

Answer:

Target VM (Ubuntu 192.168.1.38): Nmap correctly identified the system as Linux/Ubuntu. The result is reliable because SSH and Apache both revealed version banners.

Attacker notebook (Mac 192.168.1.47): Nmap did not explicitly state "macOS" in the OS info, but the open services (AirPlay, AirTunes, AnyDesk) are consistent with a Mac host. This makes the OS identification effectively correct.

Target host (Windows 192.168.1.37): Nmap could not determine the OS and reported the host as down. This is because Windows Firewall blocks ping and fingerprinting probes by default. Therefore, the OS detection was not correct in this case.

Q3. What do you think about the information you can get using nmap? Scary?

Answer:

Nmap provides a surprisingly detailed picture of a system from the outside. From a simple scan, I was able to learn which services were running, their versions, and even get hints about the operating system. This information could easily be abused by attackers to identify vulnerabilities and launch targeted exploits.

For example, my scan of the VM revealed the exact Apache and OpenSSH versions, and my Mac's scan revealed AnyDesk running — something that could be a potential entry point. It is "scary" because none of this required authentication; it is purely from probing the network. At the same time, the same capability is valuable for defenders, since it shows them what attackers can see and helps close unintended exposures.

Q4. Look at the access.log file for the web server in your Linux VM. What IP addresses do you see accessing the web server? Which devices do these IP addresses belong to?

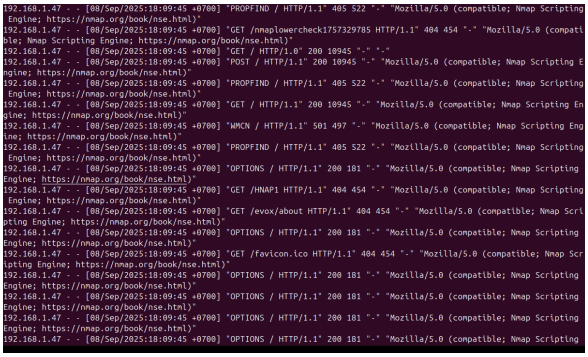
Answer:

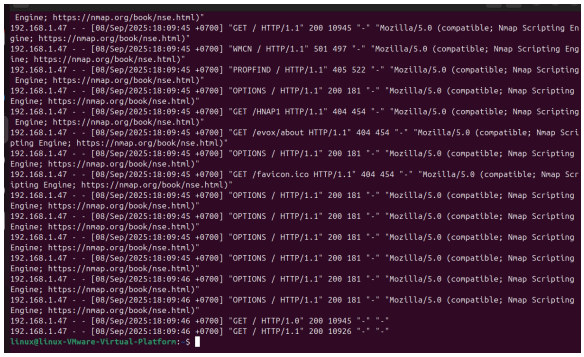
In the Apache access.log of the Linux VM, the only client IP observed was **192.168.1.47**.

- This IP belongs to the **attacker notebook (Mac)**.
- All the requests were generated by the nmap scripting engine during reconnaissance (e.g., OPTIONS, PROPFIND, GET /evox/about).

I did not see entries from the target host notebook (192.168.1.37) or from the VM itself (192.168.1.38). This confirms that the logs captured the attacker's scanning activity specifically.

```
linux@linux-VMware-Virtual-Platform:~$ sudo tail -n 50 /var/log/apache2/access.log
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "OPTIONS / HTTP/1.1" 200 181 "-" "Mozilla/5.0 (compatible; Nmap Scripting
Engine; https://nmap.org/book/nse.html)"
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "GET / HTTP/1.1" 200 10945 "-" "Mozilla/5.0 (compatible; Nmap Scripting En
gine; https://nmap.org/book/nse.html)"
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "PROPFIND / HTTP/1.1" 405 522 "-" "Mozilla/5.0 (compatible; Nmap Scripting
 Engine; https://nmap.org/book/nse.html)"
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "GET /nmaplowercheck1757329552 HTTP/1.1" 404 454 "-" "Mozilla/5.0 (compati
ble; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "GET / HTTP/1.0" 200 10945 "-" "-"
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "POST / HTTP/1.1" 200 10945 "-" "Mozilla/5.0 (compatible; Nmap Scripting E
ngine; https://nmap.org/book/nse.html)"
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "OPTIONS / HTTP/1.1" 200 181 "-" "Mozilla/5.0 (compatible; Nmap Scripting
Engine; https://nmap.org/book/nse.html)"
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "PAZD / HTTP/1.1" 501 497 "-" "Mozilla/5.0 (compatible; Nmap Scripting Eng
ine; https://nmap.org/book/nse.html)"
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "PROPFIND / HTTP/1.1" 405 522 "-" "Mozilla/5.0 (compatible; Nmap Scripting
 Engine; https://nmap.org/book/nse.html)"
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "GET /HNAP1 HTTP/1.1" 404 454 "-" "Mozilla/5.0 (compatible; Nmap Scripting
 Engine; https://nmap.org/book/nse.html)"
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "GET /favicon.ico HTTP/1.1" 404 454 "-" "Mozilla/5.0 (compatible; Nmap Scr
ipting Engine; https://nmap.org/book/nse.html)"
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "GET /evox/about HTTP/1.1" 404 454 "-" "Mozilla/5.0 (compatible; Nmap Scri
pting Engine; https://nmap.org/book/nse.html)"
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "OPTIONS / HTTP/1.1" 200 181 "-" "Mozilla/5.0 (compatible; Nmap Scripting
Engine; https://nmap.org/book/nse.html)"
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "GET / HTTP/1.1" 200 10945 "-" "Mozilla/5.0 (compatible; Nmap Scripting En
gine; https://nmap.org/book/nse.html)"
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "OPTIONS / HTTP/1.1" 200 181 "-" "Mozilla/5.0 (compatible; Nmap Scripting
Engine; https://nmap.org/book/nse.html)"
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "OPTIONS / HTTP/1.1" 200 181 "-" "Mozilla/5.0 (compatible; Nmap Scripting
Engine; https://nmap.org/book/nse.html)"
```





Q5. Find the nmap scan in the web server log. Copy the lines from the log file that were created because of the nmap scan.

Answer:

The following entries in the VM's Apache access.log were created by the nmap scan (all from attacker notebook 192.168.1.47, with Nmap Scripting Engine as the client):

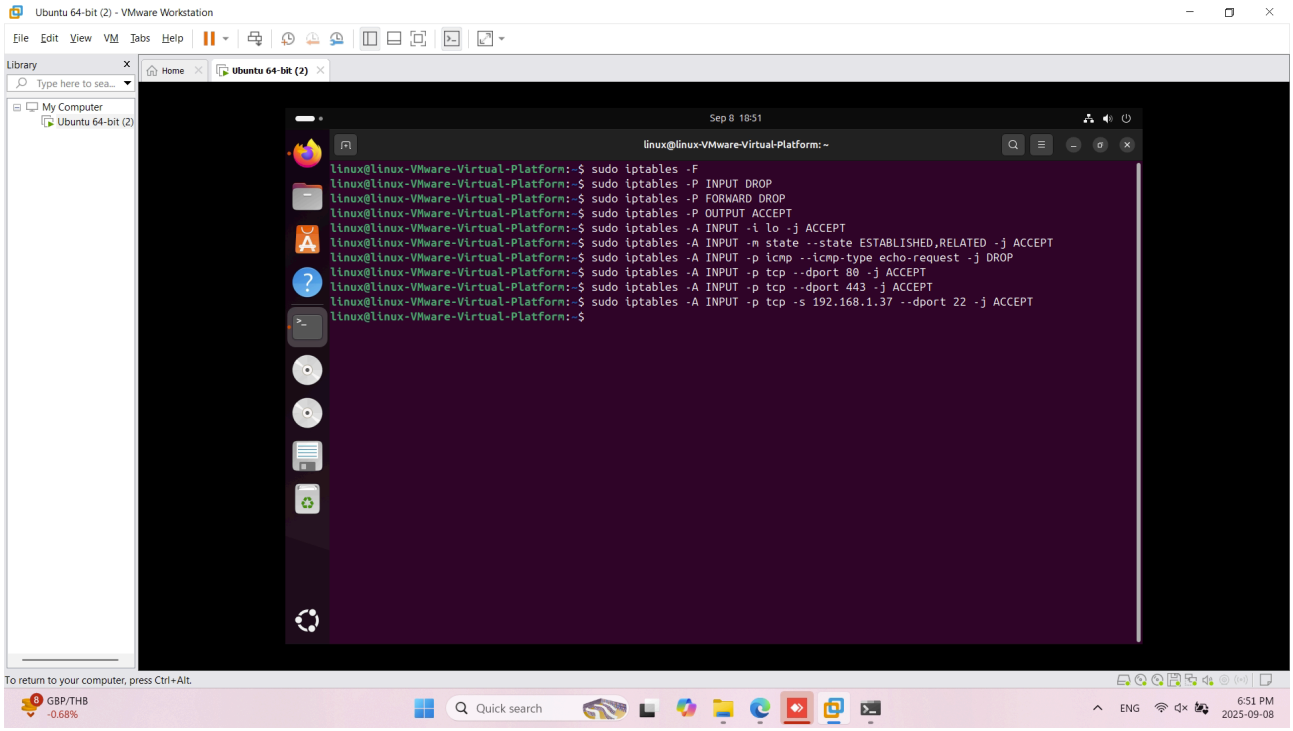
```
192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "OPTIONS / HTTP/1.1" 200 181 "-"
"Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"

192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "GET / HTTP/1.1" 200 10945 "-"
"Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"

192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "PROPFIND / HTTP/1.1" 405 522 "-"
"Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"

192.168.1.47 - - [08/Sep/2025:18:05:52 +0700] "GET /favicon.ico HTTP/1.1" 404 454
"-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/
nse.html)"

192.168.1.47 - - [08/Sep/2025:18:09:45 +0700] "GET /evox/about HTTP/1.1" 404 454
"-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/
nse.html)"
```

These requests clearly came from nmap's service/version detection phase, not from a normal browser.

Q6. After you successfully install your iptable rule(s), how do the reported results from your new nmap scan compare to your previous scan before using iptables? Look to see if OS detection, port open results, etc. have changed. Something(s) have definitely changed.

Answer:

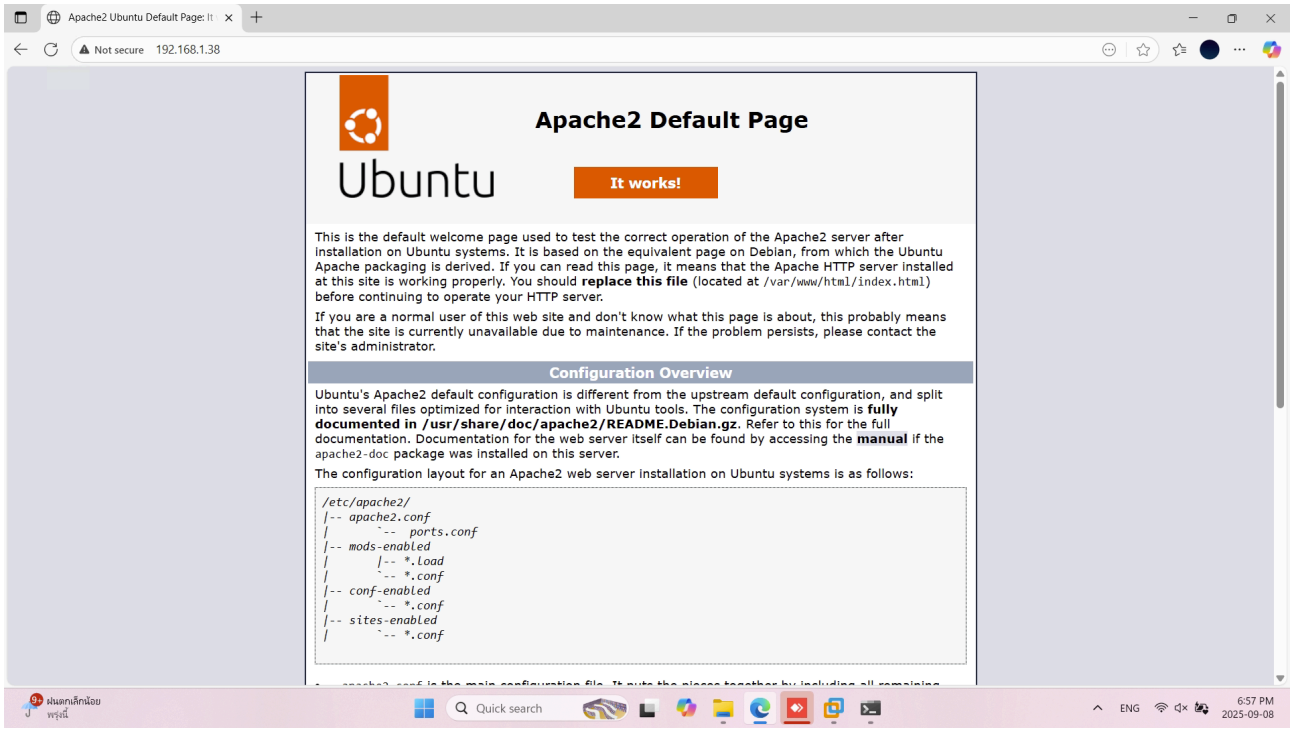
- iptables rules



- Test from attacker

```
(base) pupipatsingkhorn@Pupipats-MacBook-Air activity-03 % ping 192.168.1.38
PING 192.168.1.38 (192.168.1.38): 56 data bytes
Request timeout for icmp_seq 0
Request timeout for icmp_seq 1
Request timeout for icmp_seq 2
Request timeout for icmp_seq 3
Request timeout for icmp_seq 4
^C
--- 192.168.1.38 ping statistics ---
6 packets transmitted, 0 packets received, 100.0% packet loss
(base) pupipatsingkhorn@Pupipats-MacBook-Air activity-03 % ssh 192.168.1.38


^C
(base) pupipatsingkhorn@Pupipats-MacBook-Air activity-03 % nmap -T4 -A -v 192.168.1.38
Starting Nmap 7.98 ( https://nmap.org ) at 2025-09-08 18:53 +0700
NSE: Loaded 158 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 18:53
Completed NSE at 18:53, 0.00s elapsed
Initiating NSE at 18:53
Completed NSE at 18:53, 0.00s elapsed
Initiating NSE at 18:53
Completed NSE at 18:53, 0.00s elapsed
Initiating Ping Scan at 18:53
Scanning 192.168.1.38 [2 ports]
Completed Ping Scan at 18:53, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 18:53
Completed Parallel DNS resolution of 1 host. at 18:53, 0.50s elapsed
Initiating Connect Scan at 18:53
Scanning 192.168.1.38 [1000 ports]
Discovered open port 80/tcp on 192.168.1.38
Stats: 0:00:27 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 62.65% done; ETC: 18:53 (0:00:16 remaining)
Completed Connect Scan at 18:53, 42.71s elapsed (1000 total ports)
Initiating Service scan at 18:53
Scanning 1 service on 192.168.1.38
Completed Service scan at 18:53, 6.08s elapsed (1 service on 1 host)
NSE: Script scanning 192.168.1.38.
Initiating NSE at 18:53
Completed NSE at 18:54, 5.03s elapsed
Initiating NSE at 18:54
Completed NSE at 18:54, 0.08s elapsed
Initiating NSE at 18:54
Completed NSE at 18:54, 0.00s elapsed
Nmap scan report for 192.168.1.38
Host is up (0.017s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE  SERVICE VERSION
80/tcp  open   http    Apache httpd 2.4.58 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
| http-methods:
|_  Supported Methods: OPTIONS HEAD GET POST
|_http-server-header: Apache/2.4.58 (Ubuntu)
443/tcp closed https

NSE: Script Post-scanning.
Initiating NSE at 18:54
Completed NSE at 18:54, 0.00s elapsed
Initiating NSE at 18:54
Completed NSE at 18:54, 0.00s elapsed
Initiating NSE at 18:54
Completed NSE at 18:54, 0.00s elapsed
Read data files from: /opt/homebrew/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 54.64 seconds
(base) pupipatsingkhorn@Pupipats-MacBook-Air activity-03 %
```

- Test from host



After installing the iptables rules, the nmap scan results changed significantly compared to the original scan. Previously, the scan showed ports 22 (SSH) and 80 (HTTP) as open, with detailed service/version information and OS detection pointing to Ubuntu Linux.

After the firewall:

- The VM stopped responding to pings, so nmap reported the host as down unless I forced scanning with -Pn.
- Only port 80 (HTTP/Apache) appeared open.
- Port 22 (SSH) showed as *filtered* from the attacker notebook, because SSH was restricted to the host notebook's IP.
- OS detection was less accurate, since many of nmap's fingerprinting probes were dropped.

Conclusion: The firewall successfully reduced the system's exposure. From the attacker's perspective, the VM now looks like "just a web server," instead of a Linux host with both SSH and HTTP available.

Q7. Notice that nmap can still figure out you have Apache httpd running. Look at the access.log file for the web server in your Linux VM. Are the logs the same as in Part II?

Answer:

After enabling the firewall rules, nmap could still detect that Apache httpd was running on port 80. This is expected, since HTTP was deliberately left open. Looking at the access.log, I still observed entries from the attacker's IP (192.168.1.47), with user-agent strings from the Nmap Scripting Engine.

However, compared to Part II, the logs were simpler, they contained only web-related probes. In Part II, there were many diverse requests from nmap because it scanned multiple services (SSH and HTTP). In Part III, only HTTP requests were logged, confirming that iptables successfully blocked access to all other ports.

Q8. Explain whether or not you could prevent nmap from reaching the web server while still allowing legitimate clients to get service. Will a firewall be sufficient for this? Or do you need some other device? Please think critically about this.

Answer:

A simple firewall cannot block nmap while still allowing the general public to access a web server, because scans and legitimate HTTP requests both look the same at the packet level. If I only need a few trusted clients, I can solve this with firewall allowlists or VPN access. But for a public-facing service, I would need additional application-layer defenses (e.g., WAF, IPS, TLS client certs, rate-limiting) to distinguish scans from real traffic.

Q9. What are your firewall rules? Run iptables -L on your VM and enter the output here.

Answer:

```
linux@linux-VMware-Virtual-Platform:~$ sudo iptables -L
[sudo] password for linux:
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
DROP       icmp --  anywhere             anywhere             icmp echo-request
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:https
ACCEPT     tcp  --  192.168.1.37         anywhere             tcp dpt:ssh

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
linux@linux-VMware-Virtual-Platform:~$
```