## Cyber security project

**First part:**

**Task:** As a network developer, create and network a secure virtual network for penetration testing.

**Step 1:**

Create a virtual lab using virtual OS like = Windows server, Windows 10 & 11, Metasploitable 2, Ubuntu Linux and Pfsense.

**Reason for my choose of OS,**

**1. Windows server**: as a server it manages enterprise-level infrastructure, hosting applications, managing network and supports various server roles like domain controller, web server and file server. It also has high data storage.

**2. Windows 10 & 11**: They are day to day OS used for personal use and in almost all parts of industries and companies in the world today.

**3. Metasploitable 2**: it is a vulnerable OS designed with high exploits and vulnerabilities for pen testers.

**4. Ubuntu Linux:** as an open- source Linux-based OS with wide variety of purposes like powering servers, laptops, desktops, cloud platforms and **IOT** devices. (IOT: means internet of things).

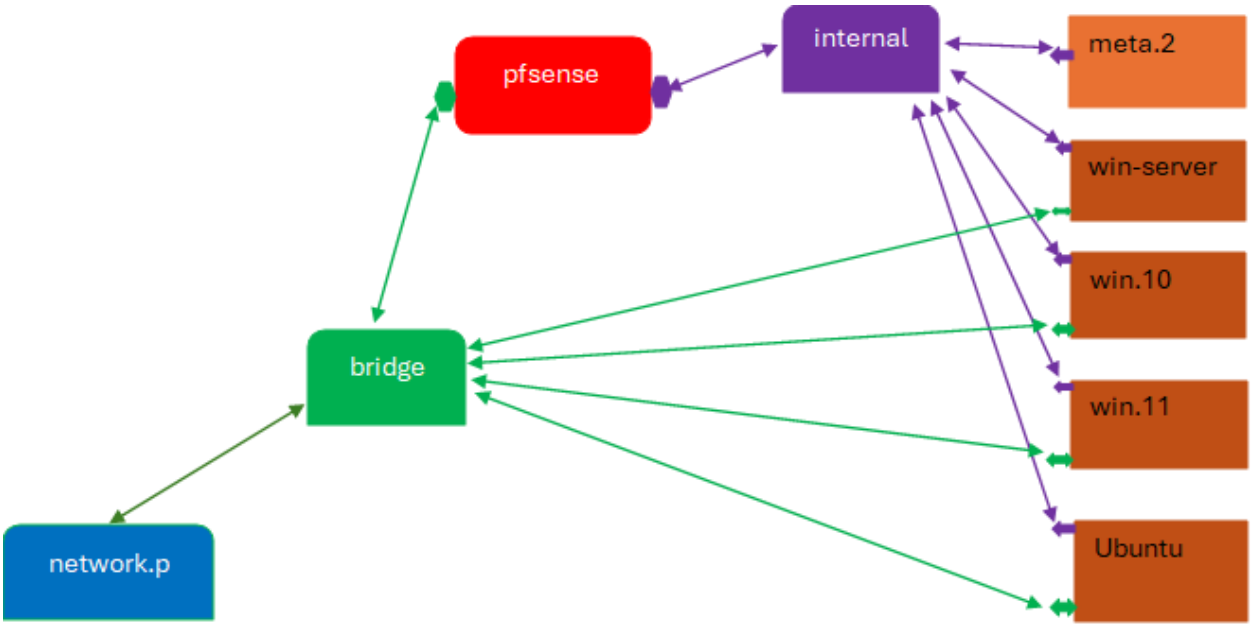
**5. Pfsense**: It will be configured as a firewall to shield the network.

The Network design: using bridge and internal network.

**a. Bridge network:** Helps the OS to get internet access independently from the internet provider.

**b. Internal network**: for only internal network communication.

**The Network design:**




1. **Pfsense:** as a firewall has two interfaces, one for the internal network and the other for external network and each interface with its own IP address. The internal network IP is used as a default gateway for OS in the internal network. The bridge interface IP is used by the pfsense to browse with the help of the Network provider.
2. **Metaspoiltable 2**: Has only on interface so we Connect its adapter to the internal network.
3. **The other OS (windows server, windows 10 & 11, Ubuntu):** Can power two adapters at once, so they are connected to the bridge and internal network interfaces.
   a. **Bridge network:** Enables the OS go to the network provider independently outside the internal network connection.
   b. **Internal network**: Binds the OS together under an internal LAN and it doesn't browse, that is why the bridge network is included in the network design.

**Setting of firewall, IDS/IPS on the internal network:**

**Make sure the OS adapters are on promiscuous mode**

1. Configuring pfsense as a firewall.

Steps.

a. login the pfsense dashboard from any OS in the internal network using the internal network default gateway IP address 192.168.1.1 from a browser.

b. on the browser tab, enter https://192.168.1.1 , when it loads you will get a warning interface of **(your network is not private),** scroll down and click on advanced or accept risk. The pfsense login page will come up.

c. Login to pfsense using admin as username and pfsense as password.

d. On the dashboard nav bar. Go to firewall, under it go to rules, once in rules go to LAN, (we use because we are configuring for a local area network) that's our internal network. On LAN there are some already predefined rules. You just must add your custom rule at the top of the default rules. [**why the top**: firewall rules are read from top to bottom, so no matter how good your custom rule is as long as it's not at the top it will be neglected].

e. Go to add = then customize your rule.

i. Action = pass, block or reject.

ii. Interface = LAN or WAN.

iii. Address family = IPV4, IPV6 or both.

iv. Protocol = ICMP, TCP, UDP .....etc.

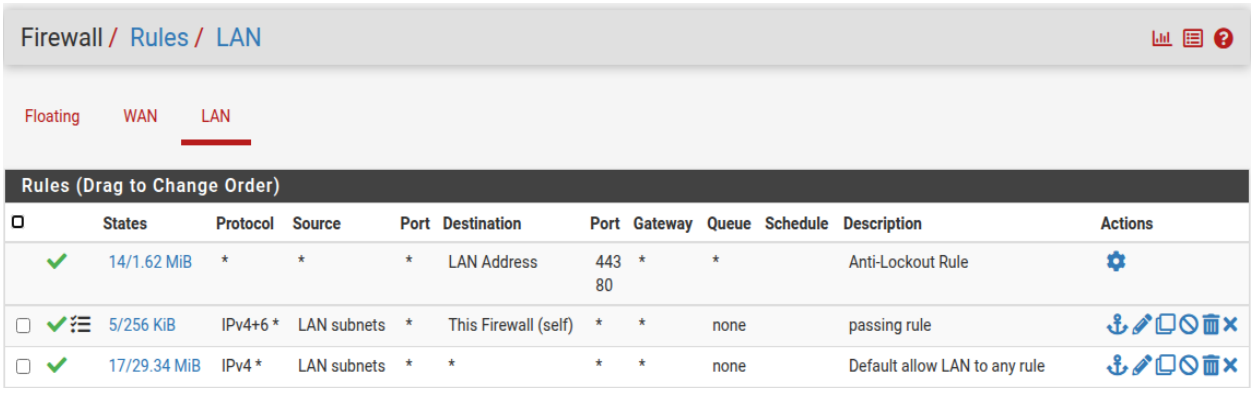
v. source = where the sender is coming from or attacker.

vi. Destination = where it is going to, that's our internal network.

vii. Log = information file or activities.

viii. Description = A text to tell what kind of rule you have written.

**E.g**. Of a custom firewall rule == pass, LAN, ICMP, IPV4+IPV6, LAN subnet, Any, this firewall, Any, passing rule.




ix. Save.

x. Apply change = To enable the customed rule.

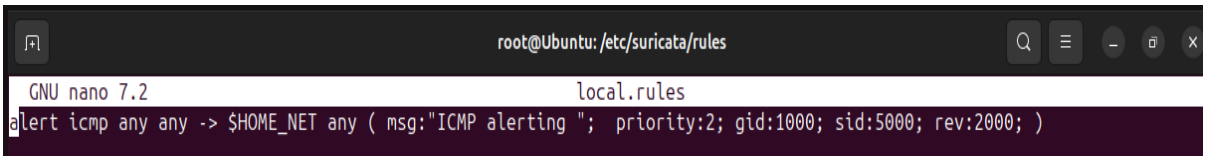
Test the rule after the change has been made. Your fire wall setup is now complete.

2. Configuring IDS [Intrusion detection system] using Suricata.

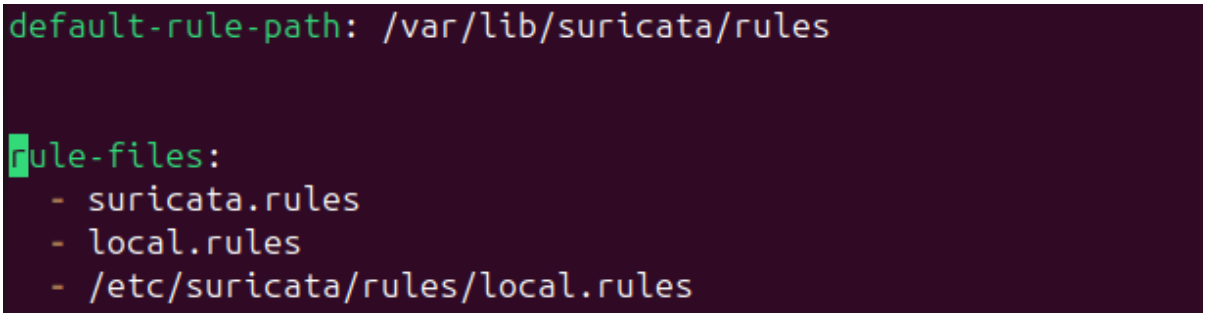
**Steps**.

a. Install Suricata in Ubuntu.
b. Locate Suricata. [whereis Suricata]
c. Locate the configuration file [ Suricata.yaml ] in the etc/suricata path.
d. Create a backup of the configuration file.
e. Edit the HOME_NET with the internal network IP address e.g. 192.168.1.0/24.
f.
```
vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[192.168.1.0/24]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
```
g. Search for af-packet and edit the ethernet e.g. enp0s8.
h.
```
# Linux high speed capture support
af-packet:
  - interface: enp0s8
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
```
i. Locate community id change from false to true, then save the edited file.
j.
```
# enable/disable the community id feature.
community-id: true
# Seed value for the ID output. Valid values are 0-65535.
community-id-seed: 0
```
k. Update Suricata to enable your changes. [Suricata-update].
l. Manually test Suricata, [Suricata –T –c  etc/suricata.yaml].
m.
```
17/4/2025 -- 09:09:55 - <Info> -- Writing rules to /var/lib/suricata/rules/suricata.rules: total: 58308; enabled: 42867;
 added: 738; removed 32; modified: 1611
17/4/2025 -- 09:09:55 - <Info> -- Writing /var/lib/suricata/rules/classification.config
17/4/2025 -- 09:09:55 - <Info> -- Testing with suricata -T.
17/4/2025 -- 09:10:32 - <Info> -- Done.
root@Ubuntu:/etc/suricata#
```
n. Trace log files, [ls –al /var/log/Suricata].
o. Enable Suricata testing interface, [tail –f fast.log].

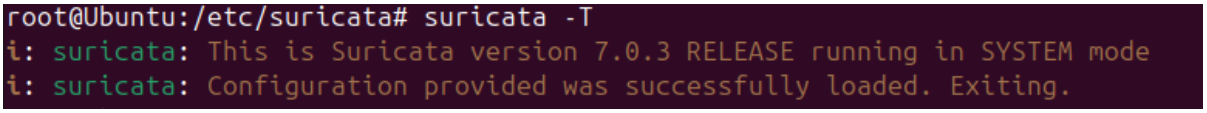
p. Once the testing interface is up go to Suricata website on kali Linux, on alerting under rules trigger copy this command, [curl https:// test .....] and run it on your kail Linux terminal. This is to ensure that Suricata is running well.
q. Back in your Ubuntu, a new log will be added to the Suricata testing logs with a massage that ends with bad traffic.
r. Set up a custom rule: locate the path to Suricata rules.
s. Make local.rules using touch command.
t.



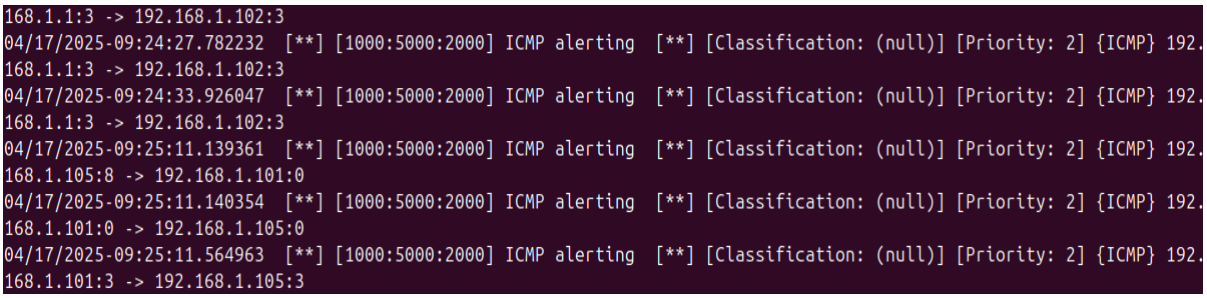

u. Back in Suricata.yaml add local.rules on the default rules path [rule-files].
v.



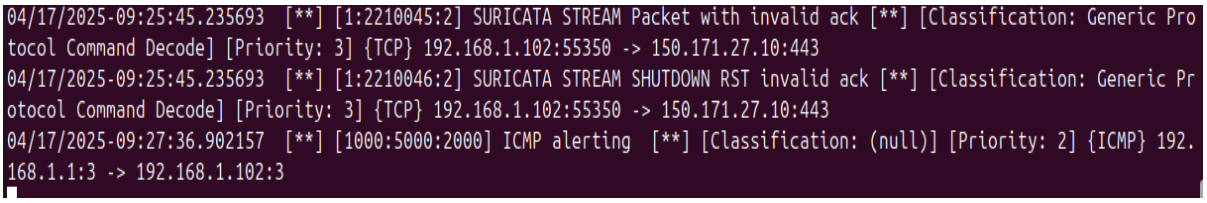

w. Add a line for the local.rules path under the rule-files.
x. Save, restart Suricata and test.
y.




z. On the console mood test your changes with [tail –f fast.log].
aa.




bb. Stop then process and retest just to confirm.
cc.




dd. Start, stop or restart Suricata with [systemctl (the action) Suricata].

Your IDS configuration is complete.

3. configuring IPS [intrusion prevention system] using Snort.

Steps.

a. Install snort.

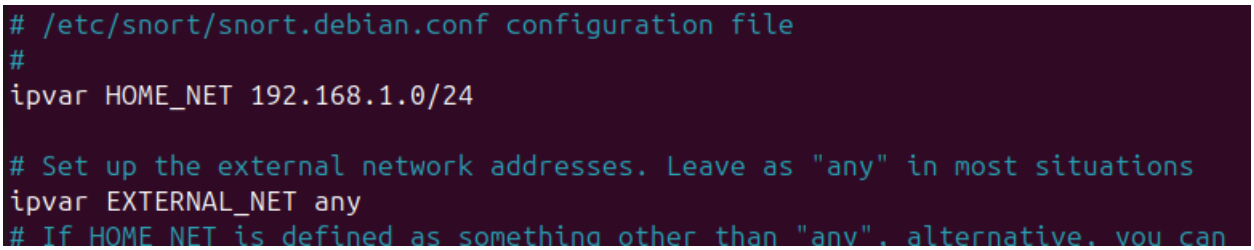
b. During the installation an interface will come, be careful not to miss it, it comes only once. Edit it and add your internal network IP address e.g. 192.168.1.0/24.

c. Locate snort. [whereis Snort].

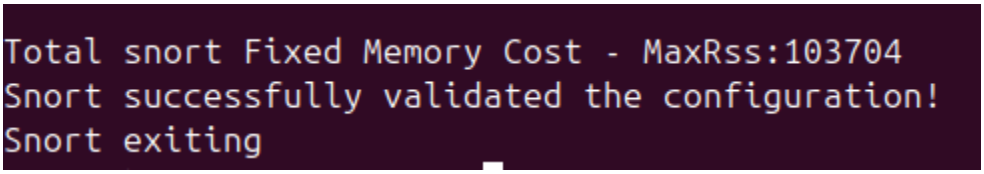
d. Follow the path of etc/snort to locate the configuration file, Snort.conf.

e. Create a backup for the configuration file.

f. Inside the configuration file edit HOME_NET by adding the same internal network IP address your used on the first interface that came up during the snort installation.

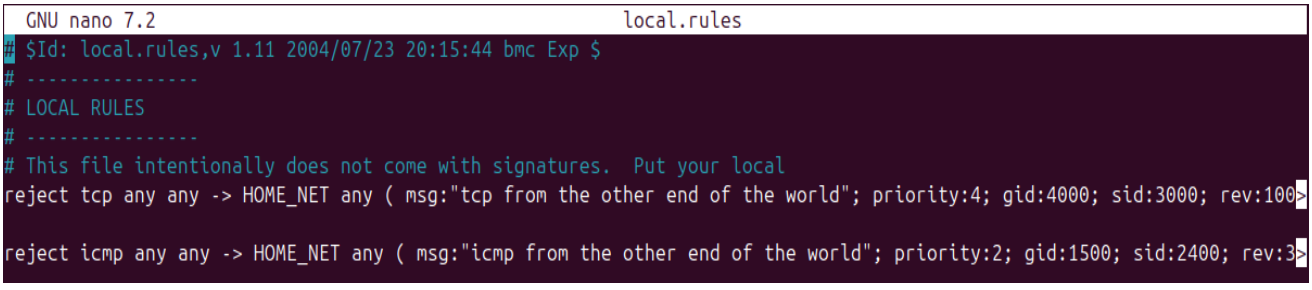



g. Test the snort interface using, [snort –T –i enp0s8 –c snort.conf].




h. To see predefined go to the rule path and list.

i. Inside local.rules customize your own IPS rules.



j. Enable console mode to make sure that snort was configured correctly, [snort –q –A console –c /etc/snort/snort.conf -i enp0s8].

```
04/22-09:59:27.815048  [**] [1500:2400:3400] icmp from the other end of the world [**] [Priority: 2] {ICMP} 192.168.1.10
5 -> 192.168.1.101
04/22-09:59:27.815261  [**] [1500:2400:3400] icmp from the other end of the world [**] [Priority: 2] {ICMP} 192.168.1.10
5 -> 192.168.1.101
04/22-09:59:27.815963  [**] [1500:2400:3400] icmp from the other end of the world [**] [Priority: 2] {ICMP} 192.168.1.10
1 -> 192.168.1.105
```

```
04/22-10:01:45.870840  [**] [4000:3000:1000] tcp from the other end of the world [**] [Priority: 4] {TCP} 192.168.1.101:
80 -> 192.168.1.105:45912
04/22-10:01:45.870934  [**] [4000:3000:1000] tcp from the other end of the world [**] [Priority: 4] {TCP} 192.168.1.101:
80 -> 192.168.1.105:45912
04/22-10:01:45.870987  [**] [4000:3000:1000] tcp from the other end of the world [**] [Priority: 4] {TCP} 192.168.1.105:
45912 -> 192.168.1.101:80
```

Intergate snort into pfsense from the pfsense  dashboard if you can.

Otherwise, your internal network and virtual lab is ready for pen testing.

**Second part**

**Task:** As a pen tester, run Pen testing session on this network (192.168.1.0/24).

**Steps 1**

**Active and passive information gathering.**

**Tools Nmap and Wireshark.**

**Active information gathering using Nmap.**

1. With my knowledge in nmap the first scan that comes to mind is a host discovery scan because our target is a network, and all networks have hosts run in them.

2. Nmap Host discovery scan: nmap (the target) -sn.

# 3. Report: nmap 192.168.1.0/24 -sn

4. Nmap scan report for pfSense.home.arpa (192.168.1.1)
5. Host is up (0.0017s latency).

6. MAC Address: 08:00:27:46:31:8B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
7. Nmap scan report for 192.168.1.101 Host is up (0.0011s latency).
8. MAC Address: 08:00:27:3F:44:71 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
9. Nmap scan report for 192.168.1.102 Host is up (0.0022s latency).
10. MAC Address: 08:00:27:37:13:6C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
11. Nmap scan report for 192.168.1.103 Host is up (0.00058s latency).
12. MAC Address: 08:00:27:C5:2D:32 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
13. Nmap scan report for 192.168.1.104 Host is up (0.0041s latency).
14. MAC Address: 08:00:27:FE:29:CF (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
15. Nmap scan report for 192.168.1.105 Host is up.

16. An aggressive scan on the network to run all nmap scans at once: nmap –T4 –A –V -Pn (the traget).

# 17. Report: nmap -T4 -A -v -Pn -oN nmap_aggressive.txt 192.168.1.0/24

18. Nmap scan report for 192.168.1.0 [host down] ..... till
19. Nmap scan report for 192.168.1.255 [host down]
20. **Nmap runs a broadcast on the target network from the first unknown host till the last**
21. Nmap scan report for pfSense.home.arpa (192.168.1.1)
22. Host is up (0.42s latency).
23. Not shown: 997 closed tcp ports (reset)
24. PORT STATE SERVICE VERSION
25. 53/tcp open tcpwrapped
26. 80/tcp open tcpwrapped
27. | Subject Alternative Name: DNS:pfSense-67341f27e86cd

28. MAC Address: 08:00:27:46:31:8B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
29. Network Distance: 1 hop TCP Sequence Prediction: Difficulty=263 (Good luck!)
30. IP ID Sequence Generation: All zeros

31. TRACEROUTE HOP RTT ADDRESS
32. 1 421.06 ms pfSense.home.arpa (192.168.1.1)
33. Nmap scan report for 192.168.1.101
34. Host is up (0.0029s latency).
35. Not shown: 998 closed tcp ports (reset)
36. PORT STATE SERVICE VERSION
37. 22/tcp open ssh OpenSSH 9.6p1 Ubuntu 3ubuntu13.8 (Ubuntu Linux; protocol 2.0)
38. | Subject Alternative Name: IP Address:127.0.0.1
39. | Issuer: organizationName=Wazuh
40. MAC Address: 08:00:27:3F:44:71 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

41. TRACEROUTE
42. HOP RTT ADDRESS
43. 1 2.88 ms 192.168.1.101
44. Nmap scan report for 192.168.1.102
45. Host is up (0.015s latency).
46. Not shown: 996 closed tcp ports (reset)
47. PORT STATE SERVICE VERSION
48. 135/tcp open tcpwrapped
49. MAC Address: 08:00:27:37:13:6C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
50. Device type: general purpose
51. Running: Microsoft Windows 10
52. OS CPE: cpe:/o:microsoft:windows_10
53. OS details: Microsoft Windows 10 1709 - 21H2
54. Network Distance: 1 hop TCP Sequence Prediction: Difficulty=255 (Good luck!)

55. *MAC: 08:00:27:37:13:6c (PCS Systemtechnik/Oracle VirtualBox virtual NIC)*

56. TRACEROUTE
57. HOP RTT ADDRESS
58. 1 14.54 ms 192.168.1.102
59. Nmap scan report for 192.168.1.103
60. Host is up (0.0019s latency).
61. Not shown: 977 closed tcp ports (reset)
62. PORT STATE SERVICE VERSION
63. 21/tcp open ftp vsftpd 2.3.4
64. *22/tcp open ssh OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)*
65. | *ssh-hostkey:*
66. |_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
67. 
68. |*http-title: Metasploitable2 - Linux*
69. |*http-server-header: Apache/2.2.8 (Ubuntu) DAV/2 111/tcp open tcpwrapped*
70. MAC Address: 08:00:27:C5:2D:32 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
71. Device type: general purpose
72. Running: Linux 2.6.X
73. OS CPE: cpe:/o:linux:linux_kernel:2.6
74. OS details: Linux 2.6.9 - 2.6.33
75. Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
76. | message_signing: disabled (dangerous, but default)
77. | nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: , NetBIOS MAC: (unknown)
78. | Names:
79. | METASPLOITABLE<00> Flags:
80. | METASPLOITABLE<03> Flags:
81. | METASPLOITABLE<20> Flags:

82. TRACEROUTE
83. HOP RTT ADDRESS

84. 1 1.91 ms 192.168.1.103
85. Nmap scan report for 192.168.1.104
86. Host is up (0.011s latency).
87. Not shown: 994 closed tcp ports (reset)
88. PORT STATE SERVICE VERSION
89. 22/tcp open tcpwrapped
90. |_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
91. 135/tcp open tcpwrapped
92. 139/tcp open tcpwrapped
93. MAC Address: 08:00:27:FE:29:CF (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
94. Device type: general purpose
95. Running: Microsoft Windows 2022
96. OS CPE: cpe:/o:microsoft:windows_server_2022
97. OS details: Microsoft Windows Server 2022
98. Uptime guess: 0.021 days (since Thu Mar 20 14:15:30 2025)
99. Network Distance: 1 hop TCP Sequence Prediction: Difficulty=261 (Good luck!)

100. Nmap simple vulnerability script scan.

# nmap --script=vuln 192.168.1.0/24

101. Nmap scan report for pfSense.home.arpa (192.168.1.1)
102. Host is up (0.49s latency).
103. | *VULNERABLE:*
104. | *Slowloris DOS attack*
105. | *State: LIKELY VULNERABLE*
106. | *IDs: CVE:CVE-2007-6750*
107. | *Slowloris tries to keep many connections to the target web server open and hold them open as long as possible. It accomplishes this by opening connections to the target web server and sending a partial request. By doing so, it starves the http server's resources causing Denial of Service.*
108. MAC Address: 08:00:27:46:31:8B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

109. Nmap scan report for 192.168.1.101

| *VULNERABLE:*
| *Authentication bypass by HTTP verb tampering*
| *State: VULNERABLE (Exploitable)*
| *This web server contains password protected resources vulnerable to authentication bypass*
| *vulnerabilities via HTTP verb tampering. This is often found in web servers that only limit access to the common HTTP methods and in misconfigured htaccess files.*
MAC Address: 08:00:27:3F:44:71 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.1.102
Host is up (0.00081s latency).

Nmap scan report for 192.168.1.103
Host is up (0.00043s latency).
| VULNERABLE:
| vsFTPd version 2.3.4 backdoor
| State: VULNERABLE (Exploitable)
| IDs: BID:48539 CVE:CVE-2011-2523
| vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04.
| Disclosure date: 2011-07-03

| smtp-vuln-cve2010-4344:
|_ The SMTP server is not Exim: NOT VULNERABLE
| *VULNERABLE:*
| *SSL POODLE information leak*
| *State: VULNERABLE*
| *IDs: BID:70574 CVE:CVE-2014-3566*
| *The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other products, uses nondeterministic CBC padding, which makes it easier for man-in-the-middle attackers to obtain cleartext data via a padding-oracle attack, aka the "POODLE" issue.*

| *VULNERABLE:*

| *Diffie-Hellman Key Exchange Insufficient Group Strength*
| *State: VULNERABLE*
| *Transport Layer Security (TLS) services that use Diffie-Hellman groups of insufficient strength, especially those using one of a few commonly shared groups, may be susceptible to passive eavesdropping attacks.*

| *ssl-poodle:*
| *VULNERABLE:*
| *SSL POODLE information leak*
| *State: VULNERABLE*
| *IDs: BID:70574 CVE:CVE-2014-3566*
| *The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other*
| *products, uses nondeterministic CBC padding, which makes it easier for man-in-the-middle attackers to obtain cleartext data via a padding-oracle attack, aka the "POODLE" issue.*
| *References:*
| *https://www.openssl.org/~bodo/ssl-poodle.pdf*
MAC Address: 08:00:27:C5:2D:32 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Host script results:
Nmap scan report for 192.168.1.104
Host is up (0.00055s latency).
PORT STATE SERVICE
22/tcp open ssh

Nmap version detection scan.

# nmap -sV --script=vuln 192.168.1.0/24

155. Nmap scan report for pfSense.home.arpa (192.168.1.1)
156. Host is up (0.17s latency).
157. MAC Address: 08:00:27:46:31:8B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

158. Nmap scan report for 192.168.1.101
159. Host is up (0.00059s latency).
160. VERSION
161. Ubuntu 3ubuntu 13.8
162. MAC Address: 08:00:27:3F:44:71 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

163. Nmap scan report for 192.168.1.102
164. Host is up (0.00044s latency).
165. VERSION
166. Micor soft Windows10 2.3.4
167. MAC Address: 08:00:27:37:13:6C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

168. Nmap scan report for 192.168.1.103
169. Host is up (0.000432s latency).
170. VERSION
171. Metasploitable2 2.3.4
172. MAC Address: 08:00:27:C5:2D:32 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

173. Nmap scan report for 192.168.1.104
174. Host is up (0.00051s latency).
175. VERSION
176. Windows Server 8.1
177. MAC Address: 08:00:27:FE:29:CF (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Full vulnerability port scan with Nmap...

# nmap -p 1-65535 --script=vuln 192.168.1.0/24

Nmap scan report for pfSense.home.arpa (192.168.1.1)
Host is up (0.0020s latency).
Not shown: 65532 filtered tcp ports (no-response)
PORT STATE SERVICE
No vulnerable port.
MAC Address: 08:00:27:46:31:8B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.1.101
Host is up (0.0022s latency).
PORT STATE SERVICE
No vulnerable port.
MAC Address: 08:00:27:3F:44:71 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.1.102
Host is up (0.0022s latency).
PORT STATE SERVICE
No vulnerable port.
MAC Address: 08:00:27:37:13:6C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Host script results:

196. |_smb-vuln-ms10-054: false
197. Nmap scan report for 192.168.1.103
198. Host is up (0.0011s latency).
199. Not shown: 65505 closed tcp ports (reset)
200. PORT STATE SERVICE
201. 21/tcp open ftp
202. | ftp-vsftpd-backdoor:
203. | VULNERABLE:
204. | vsFTPd version 2.3.4 backdoor
205. | State: VULNERABLE (Exploitable)
206. | IDs: BID:48539 CVE:CVE-2011-2523

207. | smtp-vuln-cve2010-4344:
208. |_ The SMTP server is not Exim: NOT VULNERABLE

209. | *ssl-poodle:*
210. | *VULNERABLE:*
211. | *SSL POODLE information leak*
212. | *State: VULNERABLE*
213. | *IDs: BID:70574 CVE:CVE-2014-3566*

214. | ssl-dh-params:
215. | VULNERABLE:
216. | Anonymous Diffie-Hellman Key Exchange MitM Vulnerability
217. | State: VULNERABLE

218. | Transport Layer Security (TLS) Protocol DHE_EXPORT Ciphers Downgrade MitM (Logjam)
219. | State: VULNERABLE
220. | IDs: BID:74733 CVE:CVE-2015-4000

| *distcc-cve2004-2687:*
| *VULNERABLE:*
| *distcc Daemon Command Execution*
| *State: VULNERABLE (Exploitable)*
| *IDs: CVE:CVE-2004-2687*

| ssl-ccs-injection:
| VULNERABLE:
| SSL/TLS MITM vulnerability (CCS Injection)
| State: VULNERABLE

| ssl-dh-params:
| VULNERABLE:
| Diffie-Hellman Key Exchange Insufficient Group Strength
| State: VULNERABLE

MAC Address: 08:00:27:C5:2D:32 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Host script results:
Nmap scan report for 192.168.1.104
Host is up (0.054s latency).
PORT STATE SERVICE
No vulnerable port
MAC Address: 08:00:27:FE:29:CF (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

**Passive information gathering using Wireshark.**

1. **Analysis of the Nmap scans in wireshark.**

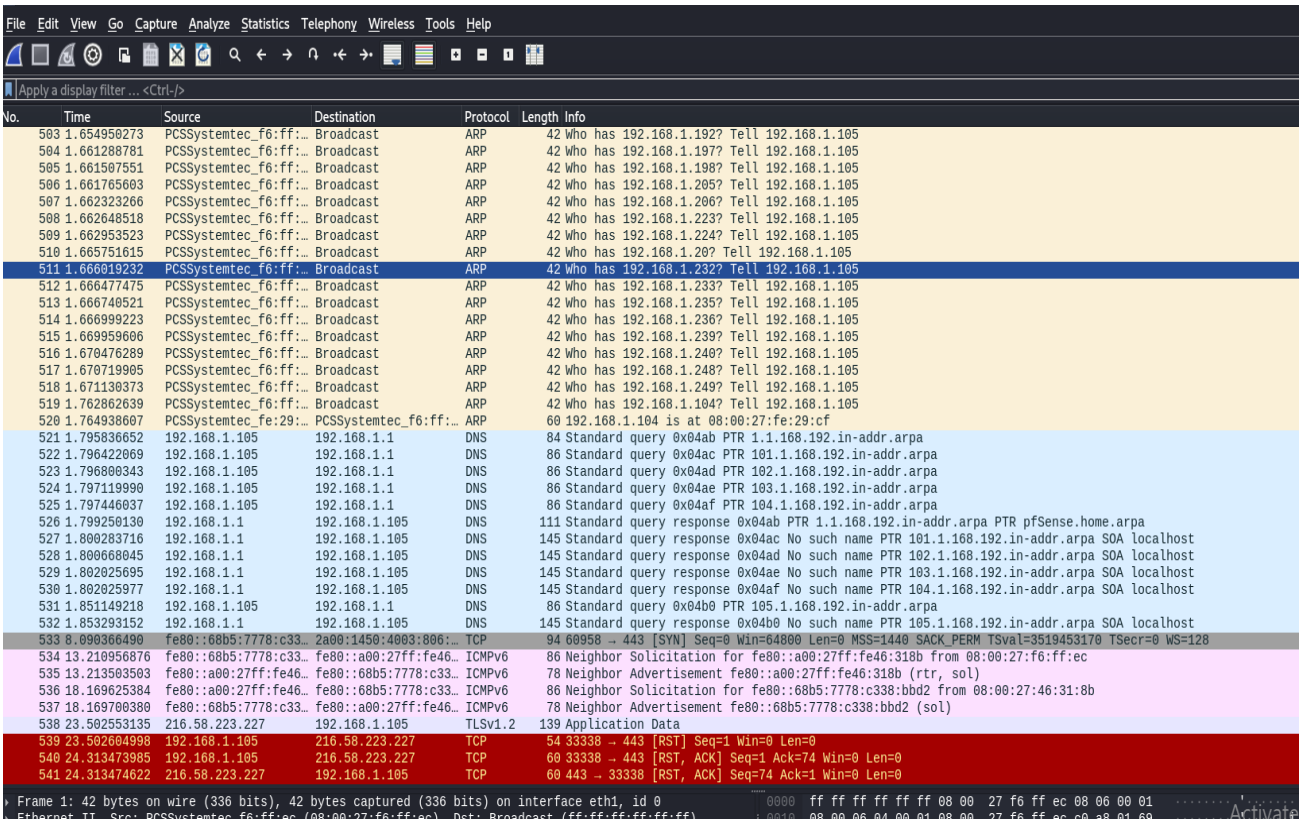
2. "No.","Time","Source","Destination","Protocol","Length","Info"
   "9.721415402","PCSSystemtec_f6:ff:ec","Broadcast","ARP","42","Who has 192.168.1.1? Tell 192.168.1.105"
3. "10.047437394","PCSSystemtec_f6:ff:ec","Broadcast","ARP","42","Who has 192.168.1.101? Tell 192.168.1.105" **(the broadcast is sent out because the sender has no idea of the subnets of this network).**
4. 
5. "11.511817445","192.168.1.105","192.168.1.1","DNS","84","Standard query 0xe058 PTR 1.1.168.192.
   in-addr.arpa""36.630657124","192.168.1.102","192.168.1.1","DNS","81","Standard query 0x50e5 A config.edge.skype.com" **(the DNS query is to know if any of the network subnets is a domain).**

,"37.812538357","192.168.1.102","52.123.243.128","TCP","66","49977 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM"

6. "550","38.200322227","52.123.243.128","192.168.1.102","TCP","66","443 > 49976 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 WS=256 SACK_PERM"
7. "551","38.202042359","192.168.1.102","52.123.243.128","TCP","60","49976 > 443 [ACK] Seq=1 Ack=1 Win=263168 Len=0"(**three-way handshake was established**)

8. ,"38.218585055","192.168.1.102","52.123.243.128","TLSv1.3","1817","Client Hello (SNI=config.edge.skype.com)"(**TLS encrypeted** )

9. "39.095244163","192.168.1.102","52.123.243.128","TCP","1454","[TCP Retransmission] 49977 > 443 [PSH, ACK] Seq=396 Ack=1 Win=263168 Len=1400"
10. "557","39.109244872","192.168.1.102","52.123.243.128","TCP","60","49976 > 443 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0"(**a reauthentication was established**),from this point a secure session has been established and some personal information's maybe sent across.
11. Wireshark will run this same connection on all the OS in the network for more info's.
12. This is what a Wireshark scan looks like.

13. 


**In conclusion of the pen testing.**

The network of 192.168.1.0/24 has life host with lots of open ports and vulnerabilities.

Host 192.168.1.103 is highly vulnerable.

Recommendation:

1. **1. Patch Management and Software Updates**
2. **Regularly update** the operating system and all installed software. Ensure that all security patches are applied as soon as they are available.
3. **Enable automatic updates** for critical software and system components.
4. **Block unnecessary ports and services** that are not required for the host to function.
5. Install and regularly **update antivirus** and **anti-malware software** to detect and prevent malicious threats.
6. **Enforce strong password policies**, requiring complex passwords for all user accounts.
7. Implement **multi-factor authentication (MFA)** for all user accounts, particularly for those with administrative privileges.
8. Disable or remove **unused user accounts** to reduce attack

9. **Disable or uninstall unnecessary services** and applications to minimize potential attack vectors.
10. Regularly **scan the host for vulnerabilities** using automated tools such as **Nmap**, **Nessus**, or **OpenVAS**.
11. ensure that the physical security of the host is adequate. For example, restrict access to servers or devices hosting critical data or applications. Develop and maintain an **incident response plan** that includes steps for detecting, responding to, and recovering from security incidents.
12. Ensure that all relevant personnel are familiar with the incident response procedures.

By following these recommendations, it will significantly reduce the vulnerability of **192.168.1.103** and improve its security posture.

Host 192.168.1.104 is a server.

Recommendation:

1. **Regularly patch and update** the server's operating system and installed software to protect against known vulnerabilities. Ensure automatic updates are enabled for critical updates.
2. **Implement the principle of least privilege (PoLP)** for all user accounts, providing only necessary access rights.
3. Use **Role-Based Access Control (RBAC)** to assign permissions based on the roles users play.
4. **Enforce strong password policies** that require complex passwords (e.g., a mix of letters, numbers, and special characters) and regular password changes.
5. Disable or remove unnecessary accounts
6. Disable **Telnet** and any other insecure remote access services.
7. Install **antivirus** and **anti-malware** software and configure it to scan the system regularly.
8. If the server stores critical data, ensure that **full disk encryption** is implemented to protect it if the server is lost or stolen.
9. **Regularly back up critical data** to offsite locations or cloud storage solutions, ensuring backups are encrypted.
10. Perform **regular vulnerability assessments** using tools like **Nessus**, **OpenVAS**, to identify potential weaknesses.

11. Ensure that your team knows how to isolate and contain compromised systems, mitigate threats, and recover data as part of the incident response process.

By implementing these cybersecurity practices, you can drastically reduce the risk of compromise for **192.168.1.104**. Regularly updating, monitoring, and hardening the server will ensure that it remains protected from various threats and remains compliant with best security practices.

Host 192.168.1.101 has the ability of hosting cloud servers and is already host one server.

Recommendation:

1. **Enforce the principle of least privilege (PoLP)** for all users and applications interacting with the cloud server. Limit access rights to only what is necessary for their tasks.
2. Use **Role-Based Access Control (RBAC)** to define and manage permissions for cloud resources.
3. Ensure that both the **operating system** and **cloud management platform** (e.g., VMware, Microsoft Azure) are **regularly patched and updated** to mitigate known vulnerabilities.
4. Automate updates where possible, but also periodically verify the patches to ensure they have been successfully applied.
5. **Update cloud applications** regularly to secure against zero-day vulnerabilities.
6. **Use TLS/SSL encryption** for data in transit to protect against eavesdropping and man-in-the-middle attacks.
7. Ensure that **encryption keys** are securely managed and rotated regularly.
8. **Restrict API access** to only authorized users and applications, ensuring that only necessary permissions are granted.
9. Use automated tools to monitor for **misconfigurations** in cloud services, as even a minor mistake can lead to major security risks.

Securing a cloud-hosting environment requires a proactive approach, including network security, data protection, continuous monitoring. The above recommendations will help reduce the risk of a breach or other cyber incidents while maintaining the availability, integrity, and confidentiality of the hosted services in host **192.168.1.101**.

Host 192.168.1.102 is open-source system.

Recommendation:

1. **Keep the operating system** and installed open-source software up to date by regularly applying **security patches** and updates. Open-source systems often release frequent security fixes for vulnerabilities.
2. **Disable unused ports** and services that are not required, for example, disable the **SSH service** if not needed or restrict access using firewalls.
3. **Secure essential services** like SSH by configuring **SSH key-based authentication** and disabling password-based authentication.
4. **Implement network segmentation** to isolate sensitive systems or critical services from less secure parts of the network.
5. **Review configuration files** regularly, especially for critical services (such as Apache, SSH), and ensure they follow secure configuration guidelines.
6. **Encrypt data in transit** using protocols like **TLS/SSL** for web traffic (if hosting websites or services) and ensure SSH connections are secured using strong cryptographic algorithms.
7. **Encrypt sensitive data** both at rest and in transit.

Securing an open-source system like **192.168.1.102** requires consistent maintenance, regular updates, strong user access control, and security best practices tailored to the open-source ecosystem. By implementing the recommendations above, you can significantly reduce the risk of a security breach or compromise on this system.

Host 192.168.1.1 is a firewall.

Recommendation:

1. Segment the network into different zones (e.g., internal, DMZ, external) to limit the spread of potential attacks. Use **Virtual Local Area Networks (VLANs)** to isolate critical systems and prevent unauthorized access. This will ensure that even if an attacker gains access to one segment, they cannot easily access others.
2. Ensure that the IDS/IPS is updated regularly with the latest signatures.
3. only the necessary traffic should be allowed through. Block all ports and protocols that are not required for the business operations.

4. Like any security device, the firewall's **firmware and software** should be kept up to date with the latest patches and updates. Vulnerabilities in firewall software can be exploited by attackers to bypass security measures.
5. Always go through firewall log.

By following these recommendations, you can strengthen the security posture of the firewall at 192.168.1.1, helping to protect your network from a variety of cyber threats.

Following this recommendation above your will be able to have a better and more secure network...

Get more insides about then Nmap scans on https://github.com/Cavdglobal/my-work.git

**End of task.....**