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).

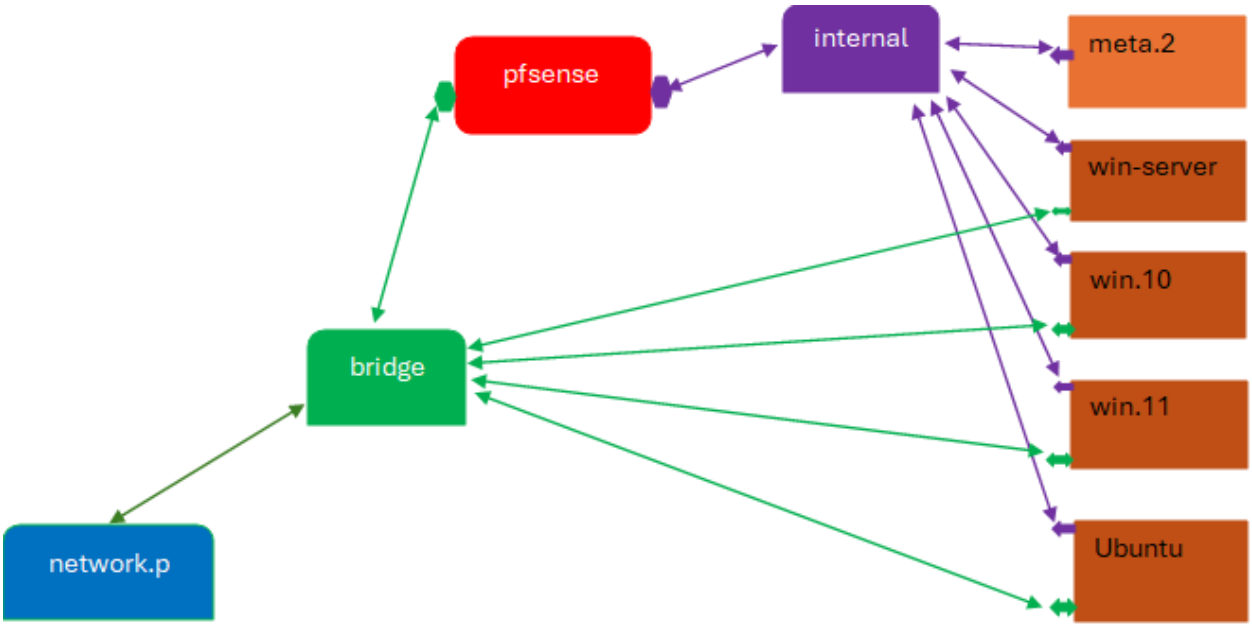
5. Pfsense: It will be configured as a firewall to shield the network.

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.

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.

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

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. Search for Af-packet and edit the ethernet e.g. enp0s8.
g. Locate community id change from false to true, then save the edited file.
h. Update Suricata to enable your changes. [Suricata-update].
i. Manually test Suricata, [Suricata –T –c etc/suricata.yaml].
j. Trace log files, [ls –al /var/log/Suricata].
k. Enable Suricata testing interface, [tail –f fast.log].
l. 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.
m. Back in your Ubuntu, a new log will be added to the Suricata testing logs with a massage that ends with bad traffic.
n. Set up a custom rule: locate the path to Suricata rules.
o. Make local.rules using touch command.
p. Back in Suricata.yaml add local.rules on the default rules path [rule-files].
q. Add a line for the local.rules path under the rule-files.
r. Save, restart Suricata and test.
s. 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].

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 with 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 -oN nmap_find.txt -sn 192.168.1.0/24**
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.

IP addresses (6 hosts up) scanned in 1.81 seconds.

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]
19. Nmap scan report for 192.168.1.2 [host down]
20. Nmap scan report for 192.168.1.9 [host down] ..... till
21. Nmap scan report for 192.168.1.255 [host down]
22. adjust_timeouts2: packet supposedly had rtt of -153801 microseconds. Ignoring time.
23. Nmap scan report for pfSense.home.arpa (192.168.1.1)
24. Host is up (0.42s latency).
25. Not shown: 997 closed tcp ports (reset)
26. PORT STATE SERVICE VERSION
27. 53/tcp open tcpwrapped
28. 80/tcp open tcpwrapped
29. |*http-server-header: nginx*
30. | *http-methods:*
31. |_ Supported Methods: GET HEAD POST OPTIONS
32. |_ *http-title: Did not follow redirect to https://pfSense.home.arpa/ 443/tcp open tcpwrapped*

33. | *tls-alpn:*
34. | *h2*
35. | *http/1.1*
36. | *http/1.0*
37. | http/0.9
38. |_ssl-date: TLS randomness does not represent time
39. |_http-title: 400 The plain HTTP request was sent to HTTPS port
40. | ssl-cert: Subject: commonName=pfSense-67341f27e86cd/organizationName=pfSense GUI default Self-Signed Certificate
41. | Subject Alternative Name: DNS:pfSense-67341f27e86cd
42. | Issuer: commonName=pfSense-67341f27e86cd/organizationName=pfSense GUI default Self-Signed Certificate | Public Key type: rsa
43. | Public Key bits: 2048 | Signature Algorithm: sha256WithRSAEncryption
44. | Not valid before: 2024-11-13T03:38:16
45. | Not valid after: 2025-12-16T03:38:16
46. | MD5: ecf5:f490:81ce:c38f:ec1f:61ef:3f00:7984
47. |_SHA-1: f64f:fdda:dc44:9b59:3b4c:3802:ca74:8cdb:35bd:57e8
48. |_http-server-header: nginx
49. MAC Address: 08:00:27:46:31:8B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
50. OS fingerprint not ideal because: Didn't receive UDP response. Please try again with -sSU
51. No OS matches for host
52. Uptime guess: 0.000 days (since Thu Mar 20 14:45:26 2025)
53. Network Distance: 1 hop TCP Sequence Prediction: Difficulty=263 (Good luck!)
54. IP ID Sequence Generation: All zeros

55. TRACEROUTE HOP RTT ADDRESS
56. 1 421.06 ms pfSense.home.arpa (192.168.1.1)
57. Nmap scan report for 192.168.1.101
58. Host is up (0.0029s latency).
59. Not shown: 998 closed tcp ports (reset)
60. PORT STATE SERVICE VERSION
61. 22/tcp open ssh OpenSSH 9.6p1 Ubuntu 3ubuntu13.8 (Ubuntu Linux; protocol 2.0)
62. | ssh-hostkey: | 256 e8:24:3f:89:b4:bc:52:4d:e3:d0:c8:4e:d5:0a:72:e9 (ECDSA)

63. |_ 256 41:09:9f:64:71:e1:1f:f5:7b:82:59:76:ac:42:5d:c7 (ED25519) 443/tcp open ssl/tcpwrapped
64. |*http-title: Site doesn't have a title (text/html; charset=utf-8)*.
65. | *http-methods:*
66. | Supported Methods: GET HEAD POST OPTIONS | tls-alpn:
67. |_ http/1.1
68. | ssl-cert: Subject: commonName=wazuh-dashboard/organizationName=Wazuh/countryName=US
69. | Subject Alternative Name: IP Address:127.0.0.1
70. | Issuer: organizationName=Wazuh
71. | Public Key type: rsa | Public Key bits: 2048
72. | Signature Algorithm: sha256WithRSAEncryption
73. | Not valid before: 2024-12-03T03:06:28
74. | Not valid after: 2034-12-01T03:06:28 | MD5: 9470:79df:4659:6438:d92d:95e7:8b92:4c0f
75. |_SHA-1: 0b43:a718:dec0:9f8d:cb7d:9a3b:1302:7297:fa9f:ad20
76. |_ssl-date: TLS randomness does not represent time
77. MAC Address: 08:00:27:3F:44:71 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
78. Device type: general purpose|router Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
79. OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
80. OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
81. Uptime guess: 9.948 days (since Mon Mar 10 16:00:54 2025) Network Distance: 1 hop TCP Sequence Prediction: Difficulty=264 (Good luck!)
82. IP ID Sequence Generation: All zeros
83. Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

84. TRACEROUTE
85. HOP RTT ADDRESS
86. 1 2.88 ms 192.168.1.101
87. Nmap scan report for 192.168.1.102
88. Host is up (0.015s latency).
89. Not shown: 996 closed tcp ports (reset)

90. PORT STATE SERVICE VERSION
91. 135/tcp open tcpwrapped
92. 139/tcp open tcpwrapped
93. 445/tcp open tcpwrapped
94. 5357/tcp open tcpwrapped
95. MAC Address: 08:00:27:37:13:6C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
96. Device type: general purpose
97. Running: Microsoft Windows 10
98. OS CPE: cpe:/o:microsoft:windows_10
99. OS details: Microsoft Windows 10 1709 - 21H2
100. Network Distance: 1 hop TCP Sequence Prediction: Difficulty=255 (Good luck!)
101. IP ID Sequence Generation: Incremental
102. Host script results:
103. |*smb2-time: Protocol negotiation failed (SMB2)*
104. | *nbstat: NetBIOS name: DESKTOP-2NO2B8J, NetBIOS user: , NetBIOS MAC: 08:00:27:37:13:6c (PCS Systemtechnik/Oracle VirtualBox virtual NIC)*
105. | *Names:*
106. | *DESKTOP-2NO2B8J<00> Flags:*
107. | *WORKGROUP<00> Flags:*
108. | *DESKTOP-2NO2B8J<20> Flags:*
109. | WORKGROUP<1e> Flags:

110. TRACEROUTE
111. HOP RTT ADDRESS
112. 1 14.54 ms 192.168.1.102
113. Nmap scan report for 192.168.1.103
114. Host is up (0.0019s latency).
115. Not shown: 977 closed tcp ports (reset)
116. PORT STATE SERVICE VERSION
117. 21/tcp open ftp vsftpd 2.3.4
118. | ftp-syst:
119. | STAT:
120. | FTP server status:
121. | Connected to 192.168.1.105

122. | Logged in as ftp | TYPE: ASCII
123. | No session bandwidth limit
124. | Session timeout in seconds is 300
125. | Control connection is plain text
126. | Data connections will be plain text
127. | vsFTPd 2.3.4 - secure, fast, stable
128. |_End of status
129. |*ftp-anon: Anonymous FTP login allowed (FTP code 230)*
130. *22/tcp open ssh OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)*
131. | *ssh-hostkey:*
132. | *1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)*
133. | 2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
134. 23/tcp open telnet Linux telnetd
135. 25/tcp open smtp Postfix smtpd
136. |_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
137. |*ssl-date: 2025-03-20T13:45:48+00:00; 0s from scanner time.*
138. | *ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX*
139. | *Issuer: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX*
140. | *Public Key type: rsa | Public Key bits: 1024*
141. | *Signature Algorithm: sha1WithRSAEncryption*
142. | *Not valid before: 2010-03-17T14:07:45*
143. | *Not valid after: 2010-04-16T14:07:45*
144. | *MD5: dcd9:ad90:6c8f:2f73:74af:383b:2540:8828*
145. |*SHA-1: ed09:3088:7066:03bf:d5dc:2373:99b4:98da:2d4d:31c6 53/tcp open tcpwrapped*
146. | *dns-nsid:*
147. | *bind.version: 9.4.2 80/tcp open tcpwrapped*
148. | *http-methods:*
149. | Supported Methods: GET HEAD POST OPTIONS
150. |*http-title: Metasploitable2 - Linux*
151. |*http-server-header: Apache/2.2.8 (Ubuntu) DAV/2 111/tcp open tcpwrapped*
152. | *rpcinfo:*
153. | *program version port/proto service*

| *100000 2 111/tcp rpcbind*
| *100000 2 111/udp rpcbind*
| *100003 2,3,4 2049/tcp nfs*
| *100003 2,3,4 2049/udp nfs*
| *100005 1,2,3 49063/tcp mountd*
| *100005 1,2,3 54936/udp mountd*
| *100021 1,3,4 47247/tcp nlockmgr*
| *100021 1,3,4 47481/udp nlockmgr*
| *100024 1 37800/udp status*
| *100024 1 38898/tcp status*
139/tcp open tcpwrapped
445/tcp open tcpwrapped Samba smbd 3.0.20-Debian
512/tcp open exec netkit-rsh rexecd
513/tcp open tcpwrapped
514/tcp open tcpwrapped
1099/tcp open tcpwrapped
1524/tcp open bindshell Metasploitable root shell
2049/tcp open tcpwrapped
2121/tcp open ftp ProFTPD 1.3.1
3306/tcp open mysql MySQL 5.0.51a-3ubuntu5
| *mysql-info:*
| *Protocol: 10*
| *Version: 5.0.51a-3ubuntu5*
| *Thread ID: 8*
| *Capabilities flags: 43564*
| *Some Capabilities: Support41Auth, LongColumnFlag, SupportsTransactions, SwitchToSSLAfterHandshake, SupportsCompression, Speaks41ProtocolNew, ConnectWithDatabase*
| *Status: Autocommit*
| Salt: 2z3-'xLO+%xOazcb?Ul? 5432/tcp open tcpwrapped
|_ssl-date: 2025-03-20T13:45:48+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=ubuntu804 base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Issuer: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Public Key type: rsa

186. | Public Key bits: 1024
187. | Signature Algorithm: sha1WithRSAEncryption
188. | Not valid before: 2010-03-17T14:07:45
189. | Not valid after: 2010-04-16T14:07:45
190. | MD5: dcd9:ad90:6c8f:2f73:74af:383b:2540:8828
191. |*SHA-1: ed09:3088:7066:03bf:d5dc:2373:99b4:98da:2d4d:31c6 5900/tcp open vnc VNC (protocol 3.3)*
192. | *vnc-info:*
193. | *Protocol version: 3.3*
194. | *Security types:*
195. | VNC Authentication (2)
196. 6000/tcp open tcpwrapped
197. 6667/tcp open irc UnrealIRCd
198. 8009/tcp open tcpwrapped
199. |_ajp-methods: Failed to get a valid response for the OPTION request 8180/tcp open tcpwrapped
200. |_http-server-header: Apache-Coyote/1.1
201. |_http-favicon: Apache Tomcat
202. |*http-title: Apache Tomcat/5.5*
203. | *http-methods:*
204. | Supported Methods: GET HEAD POST OPTIONS
205. MAC Address: 08:00:27:C5:2D:32 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
206. Device type: general purpose
207. Running: Linux 2.6.X
208. OS CPE: cpe:/o:linux:linux_kernel:2.6
209. OS details: Linux 2.6.9 - 2.6.33
210. Uptime guess: 0.018 days (since Thu Mar 20 14:20:36 2025)
211. Network Distance: 1 hop TCP Sequence Prediction: Difficulty=205 (Good luck!)
212. IP ID Sequence Generation: All zeros
213. Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
214. | message_signing: disabled (dangerous, but default)
215. | nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: , NetBIOS MAC: (unknown)
216. | Names:
217. | METASPLOITABLE<00> Flags:

| METASPLOITABLE<03> Flags:
| METASPLOITABLE<20> Flags:
| \x01\x02__MSBROWSE__\x02<01> Flags:
| WORKGROUP<00> Flags: | WORKGROUP<1d> Flags:
|_ WORKGROUP<1e> Flags:

TRACEROUTE
HOP RTT ADDRESS
1 1.91 ms 192.168.1.103
Nmap scan report for 192.168.1.104
Host is up (0.011s latency).
Not shown: 994 closed tcp ports (reset)
PORT STATE SERVICE VERSION
22/tcp open tcpwrapped
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
135/tcp open tcpwrapped
139/tcp open tcpwrapped
445/tcp open tcpwrapped
5357/tcp open tcpwrapped
5985/tcp open tcpwrapped
MAC Address: 08:00:27:FE:29:CF (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Microsoft Windows 2022
OS CPE: cpe:/o:microsoft:windows_server_2022
OS details: Microsoft Windows Server 2022
Uptime guess: 0.021 days (since Thu Mar 20 14:15:30 2025)
Network Distance: 1 hop TCP Sequence Prediction: Difficulty=261 (Good luck!)
IP ID Sequence Generation: Incremental
Host script results:
| smb2-time:
| date: 2025-03-20T13:45:42
|_ start_date: N/A
| smb2-security-mode:
| 3:1:1:

|_ Message signing enabled but not required
| nbstat: NetBIOS name: WIN-9LBCPC8O088, NetBIOS user: , NetBIOS MAC: 08:00:27:fe:29:cf (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
| Names:
| WIN-9LBCPC8O088<00> Flags:
| WORKGROUP<00> Flags:
|_ WIN-9LBCPC8O088<20> Flags:
TRACEROUTE
HOP RTT ADDRESS
1 10.77 ms 192.168.1.104
Nmap scan report for 192.168.1.105
Host is up (0.000044s latency).
All 1000 scanned ports on 192.168.1.105 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
Too many fingerprints match this host to give specific OS details Network Distance: 0 hops
Post-scan script results:
| clock-skew:
| 1h00m00s:
| 192.168.1.103
|_ 192.168.1.104
Read data files from: /usr/share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .

IP addresses (6 hosts up) scanned in 72.10 seconds

Nmap simple vulnerability script scan.

nmap -oN nmap_vuln.txt --script=vuln 192.168.1.0/24

Nmap scan report for pfSense.home.arpa (192.168.1.1)
Host is up (0.49s latency).
Not shown: 997 closed tcp ports (reset)

PORT STATE SERVICE
53/tcp open domain
80/tcp open http
| *VULNERABLE:*
| *Slowloris DOS attack*
| *State: LIKELY VULNERABLE*
| *IDs: CVE:CVE-2007-6750*
| *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.*
| *Disclosure date: 2009-09-17*
| *References:*
| *http://ha.ckers.org/slowloris/*
| *https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750*
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 (0.00056s latency).
Not shown: 998 closed tcp ports (reset)
PORT STATE SERVICE
22/tcp open ssh
| *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.*
| *References:*
|_ http://www.mkit.com.ar/labs/htexploit/
MAC Address: 08:00:27:3F:44:71 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.1.102

303. Host is up (0.00081s latency).
304. Not shown: 996 closed tcp ports (reset)
305. PORT STATE SERVICE
306. 135/tcp open msrpc
307. MAC Address: 08:00:27:37:13:6C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
308. Host script results:
309. |_samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
310. |_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: EOF
311. |_smb-vuln-ms10-054: false

312. Nmap scan report for 192.168.1.103
313. Host is up (0.00043s latency).
314. Not shown: 977 closed tcp ports (reset)
315. PORT STATE SERVICE
316. 21/tcp open ftp
317. | ftp-vsftpd-backdoor:
318. | VULNERABLE:
319. | vsFTPd version 2.3.4 backdoor
320. | State: VULNERABLE (Exploitable)
321. | IDs: BID:48539 CVE:CVE-2011-2523
322. | vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04.
323. | Disclosure date: 2011-07-03
324. | Exploit results:
325. | Shell command: id
326. | Results: uid=0(root) gid=0(root)
327. | References:
328. | http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
329. | https://www.securityfocus.com/bid/48539
330. | https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_234_backdoor.rb
331. |_ https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
332. 22/tcp open ssh

23/tcp open telnet
25/tcp open smtp
| 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.*
| *Disclosure date: 2014-10-14 | Check results:*
| *TLS_RSA_WITH_AES_128_CBC_SHA*
| *References:*
| *https://www.openssl.org/~bodo/ssl-poodle.pdf*
| *https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566*
| *https://www.securityfocus.com/bid/70574*
|*http-trace: TRACE is enabled*
| *http-sql-injection:*
| *Possible sqli for queries:*
| *http://192.168.1.103:80/dav/?C=D%3BO%3DA%27%20OR%20sqlspider*
| *http://192.168.1.103:80/mutillidae/?page=user-info.php%27%20OR%20sqlspider*
| *http://192.168.1.103:80/mutillidae/index.php?page=dns-lookup.php%27%20OR%20sqlspider*
| *http://192.168.1.103:80/mutillidae/index.php?page=set-background-color.php%27%20OR%20sqlspider*
|| *Found the following possible CSRF vulnerabilities:*
|
| *Path: http://192.168.1.103:80/dvwa/*
| *Form id:*
| *Form action: login.php*
|
| *Path: http://192.168.1.103:80/dvwa/login.php*
| *Form id:*
| Form action: login.php

| *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.*
| *Check results:*
| *WEAK DH GROUP 1*
| *Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA*
| *Modulus Type: Safe prime*
| *References:*
| *https://weakdh.org*
| *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.*
| *Disclosure date: 2014-10-14*
| *Check results:*
| *TLS_RSA_WITH_AES_128_CBC_SHA*
| *References:*
| *https://www.openssl.org/~bodo/ssl-poodle.pdf*
| *https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566*
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).
Not shown:
994 closed tcp ports (reset)
PORT STATE SERVICE

22/tcp open ssh
135/tcp open msrpc
MAC Address: 08:00:27:FE:29:CF (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Host script results:
|_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
|_smb-vuln-ms10-054: false
Nmap scan report for 192.168.1.105
Host is up (0.0000020s latency).
All 1000 scanned ports on 192.168.1.105 are in ignored states. Not shown: 1000 closed tcp ports (reset)

IP addresses (6 hosts up) scanned in 66.94 seconds

Nmap version detection scan.

nmap -oN nmap_vulnVs.txt -sV --script=vuln 192.168.1.0/24

Nmap scan report for pfSense.home.arpa (192.168.1.1)
Host is up (0.17s latency).
Not shown: 997 closed tcp ports (reset)
| *VULNERABLE:*
| *Slowloris DOS attack*
| *State: LIKELY VULNERABLE*
| *IDs: CVE:CVE-2007-6750*
| *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.

413. |
414. | *Disclosure date: 2009-09-17*
415. | *References:*
416. | *http://ha.ckers.org/slowloris/*
417. | https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
418. MAC Address: 08:00:27:46:31:8B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

419. Nmap scan report for 192.168.1.101
420. Host is up (0.00059s latency).
421. Not shown: 998 closed tcp ports (reset)
422. | *VULNERABLE:*
423. | *Authentication bypass by HTTP verb tampering*
424. | *State: VULNERABLE (Exploitable)*
425. | *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.*
426. |
427. | *Extra information:*
428. |
429. | *URIs suspected to be vulnerable to HTTP verb tampering:*
430. | */%5C1quot [POST]* |
431. | *References:*
https://www.owasp.org/index.php/Testing_for_HTTP_Methods_and_XST%28OWASP-CM-008%29
432. MAC Address: 08:00:27:3F:44:71 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
433. Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

434. Nmap scan report for 192.168.1.102
435. Host is up (0.00044s latency).
436. Not shown: 996 closed tcp ports (reset)

437. MAC Address: 08:00:27:37:13:6C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
438. Host script results:
439. |_smb-vuln-ms10-054: false

440. Nmap scan report for 192.168.1.103
441. Host is up (0.00042s latency).
442. Not shown: 977 closed tcp ports (reset)
443. | ftp-vsftpd-backdoor:
444. | VULNERABLE:
445. | vsFTPd version 2.3.4 backdoor
446. | State: VULNERABLE (Exploitable)
447. | IDs: CVE:CVE-2011-2523 BID:48539
448. | vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04.
449. | Disclosure date: 2011-07-03
450. | Exploit results:
451. | Shell command: id
452. | Results: uid=0(root) gid=0(root)
453. | References:
454. | https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_234_backdoor.rb
455. | https://www.securityfocus.com/bid/48539
456. 22/tcp open ssh OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0) 23/tcp open tcpwrapped
457. 25/tcp open smtp Postfix smtpd | ssl-dh-params:
458. | VULNERABLE:
459. | Anonymous Diffie-Hellman Key Exchange MitM Vulnerability
460. | State: VULNERABLE
461. | Transport Layer Security (TLS) services that use anonymous
462. | Diffie-Hellman key exchange only provide protection against passive eavesdropping, and are vulnerable to active man-in-the-middle attacks which could completely compromise the confidentiality and integrity of any data exchanged over the resulting session.
463. | Check results:
464. | ANONYMOUS DH GROUP 1
465. | Cipher Suite: TLS_DH_anon_WITH_DES_CBC_SHA

466. | Modulus Type: Safe prime
467. | Modulus Source: postfix builtin
468. | Modulus Length: 1024
469. | Generator Length: 8
470. | Public Key Length: 1024
471. | References:
472. | https://www.ietf.org/rfc/rfc2246.txt
473. |
474. | Transport Layer Security (TLS) Protocol DHE_EXPORT Ciphers Downgrade MitM (Logjam)
475. | State: VULNERABLE
476. | IDs: CVE:CVE-2015-4000 BID:74733
477. | The Transport Layer Security (TLS) protocol contains a flaw that is triggered when handling Diffie-Hellman key exchanges defined with the DHE_EXPORT cipher. This may allow a man-in-the-middle attacker to downgrade the security of a TLS session to 512-bit export-grade cryptography, which is significantly weaker, allowing the attacker to more easily break the encryption and monitor or tamper with the encrypted stream.
478. | Disclosure date: 2015-5-19
479. | Check results:
480. | EXPORT-GRADE DH GROUP 1
481. | Cipher Suite: TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
482. | Modulus Type: Safe prime
483. | Modulus Source: Unknown/Custom-generated
484. | Modulus Length: 512
485. | Generator Length: 8
486. | Public Key Length: 512
487. | References:
488. | https://weakdh.org
489. | https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-4000
490. |
491. | Diffie-Hellman Key Exchange Insufficient Group Strength
492. | State: VULNERABLE
493. | 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.
494. | Check results:
495. | WEAK DH GROUP 1

| Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA
| Modulus Type: Safe prime
| Modulus Source: postfix builtin
| Modulus Length: 1024
| Generator Length: 8
| Public Key Length: 1024
| References:
|_ https://weakdh.org
| ssl-poodle:
| VULNERABLE:
| SSL POODLE information leak
| State: VULNERABLE
| IDs: CVE:CVE-2014-3566 BID:70574
| 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.
| Disclosure date: 2014-10-14
| Check results:
| TLS_RSA_WITH_AES_128_CBC_SHA
| References:
| https://www.openssl.org/~bodo/ssl-poodle.pdf
| smtp-vuln-cve2010-4344:
|_ The SMTP server is not Exim: NOT VULNERABLE
|*sslv2-drown: ERROR: Script execution failed (use -d to debug)*
|*http-vuln-cve2017-1001000: ERROR: Script execution failed (use -d to debug)*
| *program version port/proto service*
| *100000 2 111/tcp rpcbind*
445/tcp open tcpwrapped
| *VULNERABLE:*
| *RMI registry default configuration remote code execution vulnerability*
| *State: VULNERABLE | Default configuration of RMI registry allows loading classes from remote URLs which can lead to remote code execution.*
|
| *References:*
| *https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/misc/java_rmi_server.rb*

1524/tcp open bindshell Metasploitable root shell

| ssl-dh-params:

| 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.

| Check results:

| WEAK DH GROUP 1

| Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA

| Modulus Type: Safe prime

| References:

| https://weakdh.org

| ssl-poodle:

| VULNERABLE:

| SSL POODLE information leak

| State: VULNERABLE

| IDs: CVE:CVE-2014-3566 BID:70574

| 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.

| Disclosure date: 2014-10-14

| Check results:

| TLS_RSA_WITH_AES_128_CBC_SHA

| References:

| https://www.openssl.org/~bodo/ssl-poodle.pdf

|http-server-header: Apache-Coyote/1.1

| http-method-tamper:

| 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.

|

| *Extra information:*
|
| *URIs suspected to be vulnerable to HTTP verb tampering:*
| */manager/html [HEAD]*
|
| *References:*
| *https://www.owasp.org/index.php/Testing_for_HTTP_Methods_and_XST*%28OWASP-CM-008%29
| http://capec.mitre.org/data/definitions/274.html
MAC Address: 08:00:27:C5:2D:32 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
Nmap scan report for 192.168.1.104
Host is up (0.00051s latency).
Not shown: 994 closed tcp ports (reset)
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH for_Windows_8.1 (protocol 2.0)
MAC Address: 08:00:27:FE:29:CF (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Host script results:
|_smb-vuln-ms10-054: false
Nmap scan report for 192.168.1.105
Host is up (0.0000010s latency).
All 1000 scanned ports on 192.168.1.105 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

IP addresses (6 hosts up) scanned in 99.22 seconds

583. Full vulnerability scan with Nmap...

nmap -oN nmap_vulnFull.txt -p 1-65535 --script=vuln 192.168.1.0/24

584. Nmap scan report for pfSense.home.arpa (192.168.1.1)
585. Host is up (0.0020s latency).
586. Not shown: 65532 filtered tcp ports (no-response)
587. PORT STATE SERVICE
588. 53/tcp open domain
589. 80/tcp open http
590. |_http-dombased-xss: Couldn't find any DOM based XSS.
591. MAC Address: 08:00:27:46:31:8B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

592. Nmap scan report for 192.168.1.101
593. Host is up (0.0022s latency).
594. PORT STATE SERVICE
595. 22/tcp open ssh
596. 443/tcp open https
597. 1514/tcp open fujitsu-dtcns
598. MAC Address: 08:00:27:3F:44:71 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

599. Nmap scan report for 192.168.1.102
600. Host is up (0.0022s latency).
601. PORT STATE SERVICE
602. 135/tcp open msrpc

139/tcp open netbios-ssn
5040/tcp open unknown
5357/tcp open wsdapi
49664/tcp open unknown
MAC Address: 08:00:27:37:13:6C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Host script results:
|_smb-vuln-ms10-054: false
Nmap scan report for 192.168.1.103
Host is up (0.0011s latency).
Not shown: 65505 closed tcp ports (reset)
PORT STATE SERVICE
21/tcp open ftp
| ftp-vsftpd-backdoor:
| 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
| Results: uid=0(root) gid=0(root)
| References:
| https://www.securityfocus.com/bid/48539
| https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_234_backdoor.rb
22/tcp open ssh
23/tcp open telnet
25/tcp open smtp
| smtp-vuln-cve2010-4344:
|_ The SMTP server is not Exim: NOT VULNERABLE
| *ssl-poodle:*
| *VULNERABLE:*
| *SSL POODLE information leak*
| *State: VULNERABLE*
| *IDs: BID:70574 CVE:CVE-2014-3566*

636. | *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.*
637. | *Disclosure date: 2014-10-14*
638. | *Check results:*
639. | *TLS_RSA_WITH_AES_128_CBC_SHA*
640. | *References:*
641. | *https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566*
642. | ssl-dh-params:
643. | VULNERABLE:
644. | Anonymous Diffie-Hellman Key Exchange MitM Vulnerability
645. | State: VULNERABLE
646. | Transport Layer Security (TLS) services that use anonymous Diffie-Hellman key exchange only provide protection against passive eavesdropping, and are vulnerable to active man-in-the-middle attacks which could completely compromise the confidentiality and integrity | of any data exchanged over the resulting session.
647. | Check results:
648. | ANONYMOUS DH GROUP 1
649. | Cipher Suite: TLS_DH_anon_EXPORT_WITH_DES40_CBC_SHA
650. | Modulus Type: Safe prime Modulus Source: Unknown/Custom-generated Modulus Length: 512
651. | Generator Length: 8
652. | Public Key Length: 512
653. | References:
654. | https://www.ietf.org/rfc/rfc2246.txt
655. |
656. | Transport Layer Security (TLS) Protocol DHE_EXPORT Ciphers Downgrade MitM (Logjam)
657. | State: VULNERABLE
658. | IDs: BID:74733 CVE:CVE-2015-4000
659. | The Transport Layer Security (TLS) protocol contains a flaw that is triggered when handling Diffie-Hellman key exchanges defined with the DHE_EXPORT cipher. This may allow a man-in-the-middle attacker to downgrade the security of a TLS session to 512-bit export-grade cryptography, which is significantly weaker, allowing the attacker to more easily break the encryption and monitor or tamper with | the encrypted stream. Disclosure date: 2015-5-19

660. | Check results:
661. | EXPORT-GRADE DH GROUP 1
662. | Cipher Suite: TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
663. | Modulus Type: Safe prime
664. | Modulus Source: Unknown/Custom-generated
665. | Modulus Length: 512
666. | Generator Length: 8
667. | Public Key Length: 512
668. | References:
669. | https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-4000
670. | https://weakdh.org
671. |
672. | Diffie-Hellman Key Exchange Insufficient Group Strength
673. | State: VULNERABLE
674. | 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.
675. | Check results:
676. | WEAK DH GROUP 1
677. | Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA
678. | Modulus Type: Safe prime
679. | Modulus Source: postfix builtin
680. | Modulus Length: 1024
681. | Generator Length: 8
682. | Public Key Length: 1024
683. | References:
684. |_ https://weakdh.org
685. 53/tcp open domain
686. |*http-trace: TRACE is enabled | http-sql-injection: | Possible sqli for queries:*
687. | *http://192.168.1.103:80/mutillidae/index.php?page=user-info.php%27%20OR%20sqlspider*
688. | *http-enum:*
689. | */tikiwiki/: Tikiwiki*
690. | */test/: Test page*
691. | */phpinfo.php: Possible information file*
692. | */phpMyAdmin/: phpMyAdmin*
693. | */doc/: Potentially interesting directory w/ listing on 'apache/2.2.8 (ubuntu) dav/2'*

694. | *http-csrf:*
695. | *Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=192.168.1.103*
696. | *Found the following possible CSRF vulnerabilities:*
697. |
698. | *Path: http://192.168.1.103:80/dvwa/*
699. | *Form id:* | *Form action: login.php*
700. |
701. | *Path: http://192.168.1.103:80/dvwa/login.php*
702. | *Form id:* | *Form action: login.php*
703. |
704. | *Path: http://192.168.1.103:80/twiki/TWikiDocumentation.html*
705. | *Form id:* | *Form action: http://TWiki.org/cgi-bin/passwd/TWiki/WebHome*
706. | *Form id: id-bad-blog-entry-tr*
707. | Form action: index.php?page=view-someones-blog.php
708. | rmi-vuln-classloader:
709. | VULNERABLE:
710. | RMI registry default configuration remote code execution vulnerability
711. | State: VULNERABLE
712. | Default configuration of RMI registry allows loading classes from remote URLs which can lead to remote code execution.
713. |
714. | References:
715. |_ https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/misc/java_rmi_server.rb
716. | *distcc-cve2004-2687:*
717. | *VULNERABLE:*
718. | *distcc Daemon Command Execution*
719. | *State: VULNERABLE (Exploitable)*
720. | *IDs: CVE:CVE-2004-2687*
721. | *Risk factor: High CVSSv2: 9.3 (HIGH) (AV:N/AC:M/Au:N/C:C/I:C/A:C)*
722. | *Allows executing of arbitrary commands on systems running distccd 3.1 and*
723. | *earlier. The vulnerability is the consequence of weak service configuration.*
724. |
725. | *Disclosure date: 2002-02-01* | *Extra information:*
726. |
727. | *uid=1(daemon) gid=1(daemon) groups=1(daemon)*

728. |
729. | *References:*
730. | *https://distcc.github.io/security.html*
731. | ssl-ccs-injection:
732. | VULNERABLE:
733. | SSL/TLS MITM vulnerability (CCS Injection)
734. | State: VULNERABLE
735. | Risk factor: High
736. | OpenSSL before 0.9.8za, 1.0.0 before 1.0.0m, and 1.0.1 before 1.0.1h does not properly restrict processing of ChangeCipherSpec messages, which allows man-in-the-middle attackers to trigger use of a zero length master key in certain OpenSSL-to-OpenSSL communications, and consequently hijack sessions or obtain sensitive information, via a crafted TLS handshake, aka the "CCS Injection" vulnerability.
737. |
738. | References:
739. | http://www.openssl.org/news/secadv_20140605.txt
740. | https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0224
741. |_ http://www.cvedetails.com/cve/2014-0224
742. | ssl-dh-params:
743. | VULNERABLE:
744. | Diffie-Hellman Key Exchange Insufficient Group Strength
745. | State: VULNERABLE
746. | 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.
747. | Check results:
748. | WEAK DH GROUP 1
749. | Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA
750. | Modulus Type: Safe prime | Modulus Source: Unknown/Custom-generated
751. | Modulus Length: 1024
752. | Generator Length: 8
753. | Public Key Length: 1024
754. | References:
755. |_ https://weakdh.org
756. | ssl-poodle:
757. | VULNERABLE:
758. | 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.
| Disclosure date: 2014-10-14
| Check results:
| TLS_RSA_WITH_AES_128_CBC_SHA
| References:
| https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566
|_ ERROR: Closing Link: [192.168.1.105] (Throttled: Reconnecting too fast) - Email admin@Metasploitable.LAN for more information.
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).
Not shown: 65521 closed tcp ports (reset)
PORT STATE SERVICE
22/tcp open ssh
49671/tcp open unknown
MAC Address: 08:00:27:FE:29:CF (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Host script results:
Nmap scan report for 192.168.1.105 Host is up (0.0000010s latency).
All 65535 scanned ports on 192.168.1.105 are in ignored states.
Not shown: 65535 closed tcp ports (reset)

IP addresses (6 hosts up) scanned in 885.31 seconds

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"
   "10.047876751","PCSSystemtec_3f:44:71","PCSSystemtec_f6:ff:ec","ARP","60","192.168.1.101 is at 08:00:27:3f:44:71" **(the broadcast is sent out because 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.

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.....