

Final Project

Digmine

Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:



Network Topology & Critical Vulnerabilities



Exploits Used



Avoiding Detect

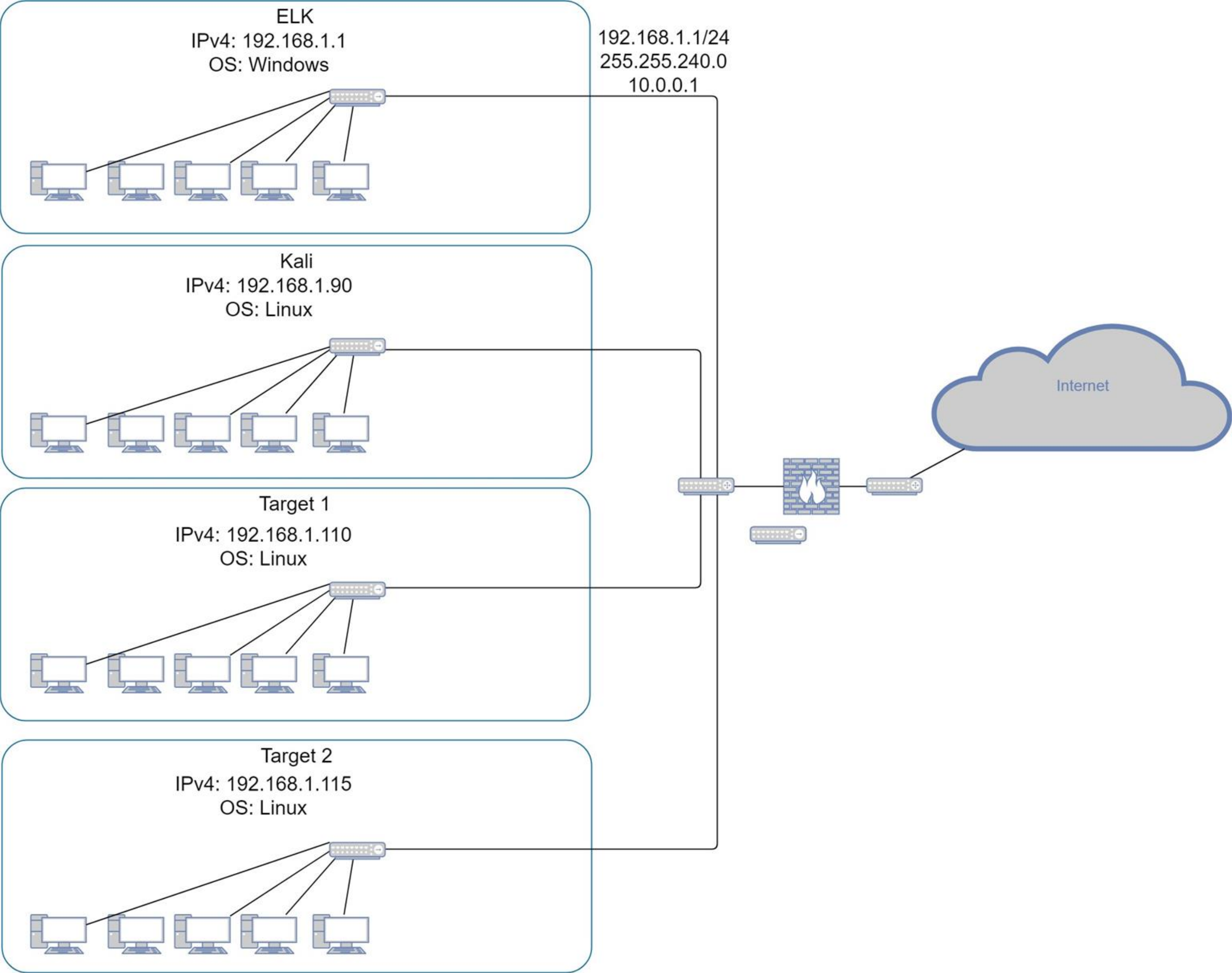


Maintaining Access



Network Topology & Critical Vulnerabilities

Network Topology



Network

Address Range:

192.168.1.1/225

Netmask:

255.255.240.0

Gateway:

10.0.0.1

Machines

IPv4:192.168.1.1

OS: Windows

Hostname: ELK

IPv4:192.168.90

OS: Linux

Hostname: Kali

IPv4:192.168.110

OS: Linux

Hostname: Target 1

IPv4:192.168.115

OS: Linux

Hostname: Target 2

Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in Target 1.

Vulnerability	Description	Impact
SSH	22/tcp	OpenSSH
HTTP	80/tcp	Apache httpd 2.4.10
rpcbind	111/tcp	2-4
netbios-ssn	139/tcp	Samba smbd 3.X - 4.X

Critical Vulnerabilities: Target 2

Our assessment uncovered the following critical vulnerabilities in Target 2.

Vulnerability	Description	Impact
SSH	22/tcp	OpenSSH
HTTP	80/tcp	Apache httpd 2.4.10
rpcbind	111/tcp	2-4
netbios-ssn	139/tcp	Samba smbd 3.X - 4.X

Exploits Used

Exploitation: SSH

Summarize the following:

- How did you exploit the vulnerability?
SSH method to log in with user1 account we found
- What did the exploit achieve?
Gaining a user shell
- Include a screenshot or command output illustrating the exploit.
`ssh michael@192.168.1.110`

Exploitation: HTTP

Summarize the following:

- How did you exploit the vulnerability?

Nmap and wpscan

- What did the exploit achieve?

Enumerating users and vulnerable plugins from wordpress website

- Include a screenshot or command output illustrating the exploit.

`wpscan --url http://192.168.1.110/wordpress --wp-content-dir -eu`

Exploitation: MySQL 5.5

Summarize the following:

- How did you exploit the vulnerability?

Hosting the file with Python's SimpleHTTPServer module

- What did the exploit achieve?

Log in to the MySQL database mysql

- Include a screenshot or command output illustrating the exploit.

`python -m SimpleHTTPServer 80`

Avoiding Detection

Stealth Exploitation of HTTP Errors

Monitoring Overview

- Which alerts detect this exploit? Excessive HTTP Errors
- Which metrics do they measure? `http.response.status_code`
- Which thresholds do they fire at? 400

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
- Are there alternative exploits that may perform better?
- If possible, include a screenshot of your stealth technique.

Stealth Exploitation of HTTP Request Size

Monitoring Overview

- Which alerts detect this exploit?
- Which metrics do they measure? `http.request.bytes`
- Which thresholds do they fire at? 3500

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
- Are there alternative exploits that may perform better?
- If possible, include a screenshot of your stealth technique.

Stealth Exploitation of CPU Usage Monitor

Monitoring Overview

- Which alerts detect this exploit? `system.process.cpu.total.pct`
- Which metrics do they measure? Cpu Total %
- Which thresholds do they fire at? 0.5

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
- Are there alternative exploits that may perform better?
- If possible, include a screenshot of your stealth technique.

Maintaining Access

Backdooring the Target

Backdoor Overview

- What kind of backdoor did you install
 - reverse shell/backdoor.php with netcat listener
- How did you drop it (via Metasploit, phishing, etc.)?
 - *command injection attacks*
 - *using curl as the main driver*
 - *http://192.168.1.115/contact.php*
- How do you connect to it?
 - *http://192.168.1.115/contact.php?cmd=id*

Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:



Network Topology & Critical Vulnerabilities



Alerts Implemented



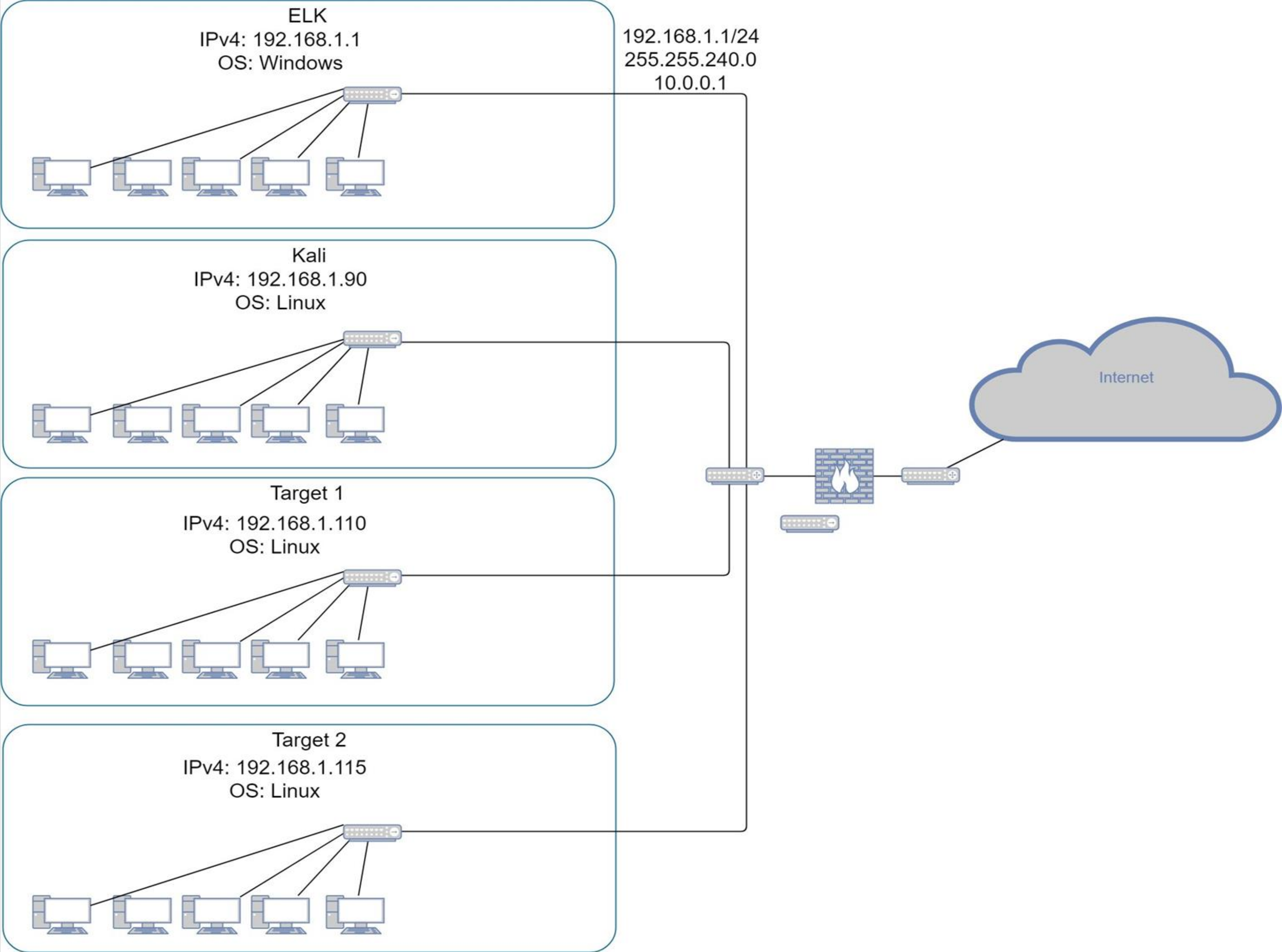
Hardening



Implementing Patches

Network Topology & Critical Vulnerabilities

Network Topology



Network
Address Range:
192.168.1.1/255
Netmask: 255.255.240.0
Gateway: 10.0.0.1

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

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

Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in Target 1.

Vulnerability	Description	Impact
SSH	22/TCP OpenSSH	Medium
HTTP	80/TCP Apache httpd 2.4.10	High
rpcbind	111/TCP 2-4	Medium
netbios-ssn	139/TCP Samba smbd 3.X - 4.X	Medium

Critical Vulnerabilities: Target 2

Our assessment uncovered the following critical vulnerabilities in Target 2.

Vulnerability	Description	Impact
SSH	22/TCP OpenSSH	Medium
HTTP	80/TCP Apache httpd 2.4.10	High
rpcbind	111/TCP 2-4	Medium
netbios-ssn	139/TCP Samba smbd 3.X - 4.X	Medium
Contact.php	Backdoor Access	High

Alerts Implemented

Excessive HTTP Errors

- Metric: packetbeat-*, HTTP Errors
- Threshold: Above 400 for the last 5 minutes.



☐

be0a11d4-2f13-4968-bcaf-a9bed764ba8c

Excessive HTTP Errors

✓ OK

a few seconds ago



Current status for 'Excessive HTTP Errors'

Execution history

Action statuses




Last one hour

▼

Trigger time	State
2020-08-25T04:23:00+00:00	✓ OK
2020-08-25T04:22:00+00:00	✓ OK
2020-08-25T04:21:00+00:00	✓ OK
2020-08-25T04:20:00+00:00	✓ OK
2020-08-25T04:19:00+00:00	✓ OK
2020-08-25T04:18:00+00:00	✓ OK
2020-08-25T04:17:00+00:00	✓ OK
2020-08-25T04:16:00+00:00	✓ OK
2020-08-25T04:15:00+00:00	✓ OK

HTTP Request Size Monitor


- Metric: packetbeat-*http.request.bytes
- Threshold: Above 3500 for the last minute.

<input type="checkbox"/> ID	Name	State	Last fired	Last triggered	Comment	Actions
<input type="checkbox"/> da5e6c1d-c770-40c1-856d-100e62e2757a	HTTP Request Size Monitor	 Firing	a few seconds ago	a few seconds ago		 

Current status for 'HTTP Request Size Monitor'

Execution history

Action statuses

Last one hour 

Trigger time

State

2020-08-25T04:22:00+00:00

 Firing

2020-08-25T04:21:00+00:00

 Firing

2020-08-25T04:20:00+00:00

 Firing

2020-08-25T04:19:00+00:00

 Firing


2020-08-25T04:18:00+00:00

 Firing


2020-08-25T04:17:00+00:00

 Firing

2020-08-25T04:16:00+00:00

 OK

2020-08-25T04:15:00+00:00

 OK

2020-08-25T04:14:00+00:00

 OK

CPU Usage Monitor

- Metric: metricbeat-*, system.process.cpu.total.pct
- Threshold: Above 0.5 for the last 5 minutes

28b4dca7-f5e6-40da-b6d2-f5e5f8043d31

CPU Usage Monitor

✓ OK

a few seconds ago

Current status for 'CPU Usage Monitor'

Execution history

Action statuses

Last one hour

Trigger time	State	Comment
2020-08-25T04:20:00+00:00	✓ OK	
2020-08-25T04:19:00+00:00	✓ OK	
2020-08-25T04:18:00+00:00	✓ OK	
2020-08-25T04:17:00+00:00	✓ OK	
2020-08-25T04:16:00+00:00	✓ OK	
2020-08-25T04:15:00+00:00	✓ OK	
2020-08-25T04:14:00+00:00	✓ OK	
2020-08-25T04:13:00+00:00	✓ OK	
2020-08-25T04:12:00+00:00	✓ OK	
2020-08-25T04:11:00+00:00	✓ OK	

Hardening

Hardening Against Brute Force Attacks on Target 1,2

Patch: Invalid credentials lock out.

Why the patch works: Prevents excessive login attempts

How to install it: Implementing an account lockout/timeout system.

Hardening Against DOS Attacks on Target 1,2

Patch: Whitelisting IP addresses, Load Balancer

Why the patch works: Only accepts connections from trusted IP address ranges. Installing a load balancer will help lighten the traffic burden placed on each server and optimize network traffic and processing.

How to install:

Through Network/Firewall settings, set Whitelisting list.

Load Balancer can be hardware or software.

Hardware: Install the device alongside the network.

Software: Through the application.

Hardening Against Excessive CPU Usage on Target 1,2

Patch: Creating several different alerts at different threshold of CPU Usage. Limit max CPU usage for each core.

Why the patch works: Alerts us to how much activity is going on in the machine. Sets a limit to how much CPU can be actually used.

How to install:

Create an alert at 75%, 100% CPU Usage.

Install software/program that limits CPU usage.

Can also use Task Manager to limit what cores a process is allowed to use.

Hardening Against Remote Access on Target 2

1. Patch: Update Kernel

Why the patch works: This is the standard way to make sure everything is up to date.

How to install: `sudo apt-get upgrade kernel`
`sudo reboot`

1. Patch: Canonical Livepatch

Why the patch works: Livepatch that updates software as soon as it comes out.

How to install: `sudo snap install canonical-livepatch`
`sudo canonical-livepatch enable`

1. Patch: KernelCare

Why the patch works: KernelCare is an 'install and forget' solution. Once installed, KernelCare automatically downloads and applies new kernel security patches, without rebooting the server.

How to install: `wget -qq -O - -- https://kernelcare.com/installer | bashsudo /usr/bin/kcarectl --register <your key>`

Implementing Patches

Implementing Patches with Ansible

Playbook Overview


Ansible is a popular open-source tool that provides automation, configuration management, and orchestration all in one. The patching is customizable via role's variables definition.

Run:

ansible-playbook orapatch.yml -k

The "-k" option will prompt you to enter the SSH password.

If you're using SSH keys then "-k" option can be omitted.



Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:



Network Topology & Critical Vulnerabilities



Traffic Profile



Normal Activity

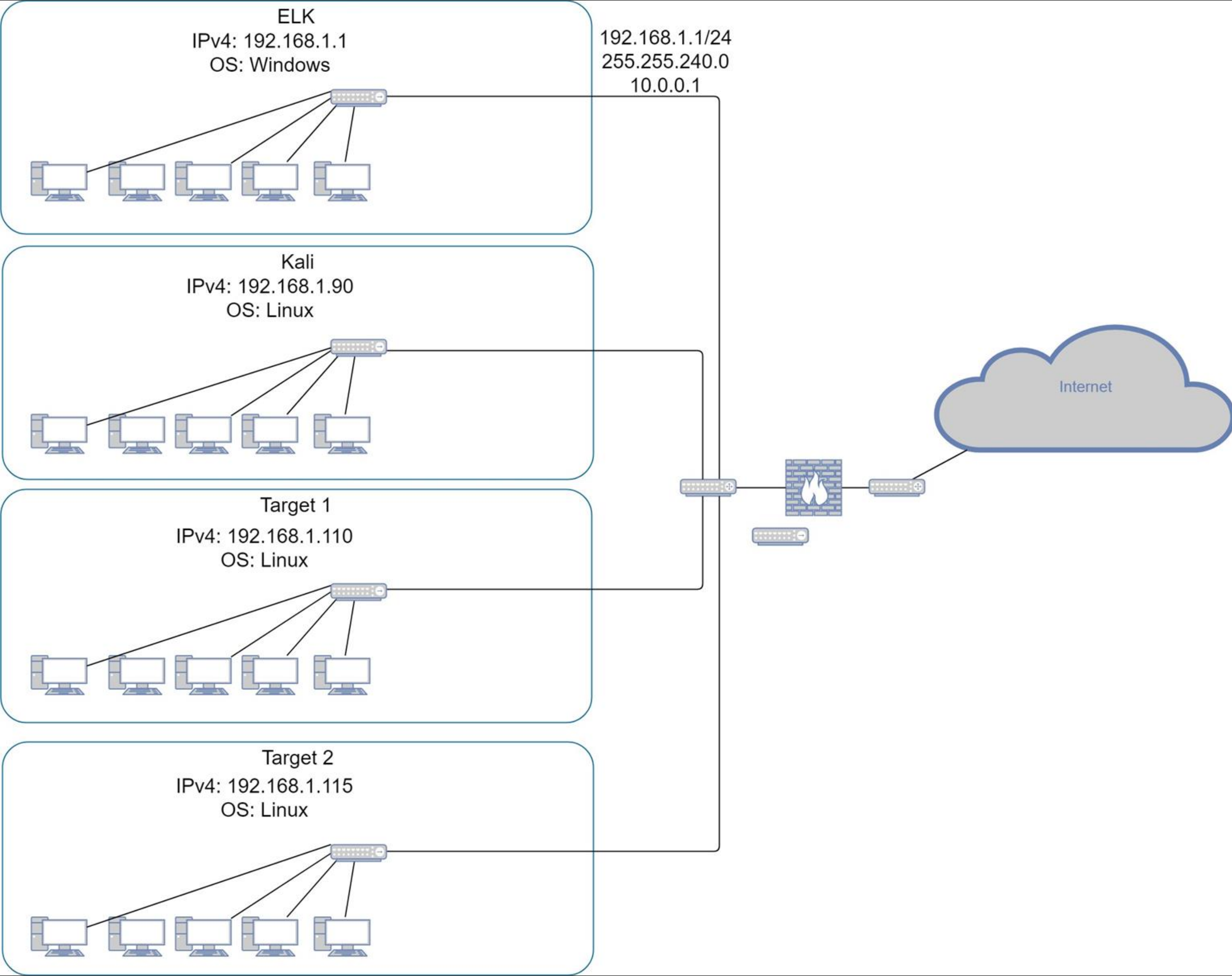


Malicious Activity



Network Topology & Critical Vulnerabilities

Network Topology



Network

Address Range:
192.168.1.1/225
Netmask:
255.255.240.0
Gateway:
10.0.0.1

Machines

IPv4:192.168.1.1
OS: Windows
Hostname: ELK

IPv4:192.168.90
OS: Linux
Hostname: Kali

IPv4:192.168.110
OS: Linux
Hostname: Target 1

IPv4:192.168.115
OS: Linux
Hostname: Target 2

Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in Target 1.

Vulnerability	Description	Impact
SSH	22/tcp	OpenSSH
HTTP	80/tcp	Apache httpd 2.4.10
rpcbind	111/tcp	2-4
netbios-ssn	139/tcp	Samba smbd 3.X - 4.X

Critical Vulnerabilities: Target 2

Our assessment uncovered the following critical vulnerabilities in Target 2.

Vulnerability	Description	Impact
SSH	22/tcp	OpenSSH
HTTP	80/tcp	Apache httpd 2.4.10
rpcbind	111/tcp	2-4
netbios-ssn	139/tcp	Samba smbd 3.X - 4.X

Traffic Profile

Traffic Profile

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

Feature	Value	Description
Top Talkers (IP Addresses)	172.16.4.205 (26m Bytes) 166.62.111.64 (16M) Bytes	Machines that sent the most traffic.
Most Common Protocols	UDP TCP TLSv1.2 and 1.3	Three most common protocols on the network.
# of Unique IP Addresses	808	Count of observed IP addresses.
Subnets	255.255.255.0 is the only range observed in the private ip's	Observed subnet ranges.
# of Malware Species	68	Number of malware binaries identified in traffic.

Behavioral Analysis

Purpose of Traffic on the Network

Users were observed engaging in the following kinds of activity.

“Normal” Activity

- Watching YouTube and surfing the internet
- Downloading and installing desktop backgrounds

Suspicious Activity

- Set up AD network and domain controller
- Downloading malware

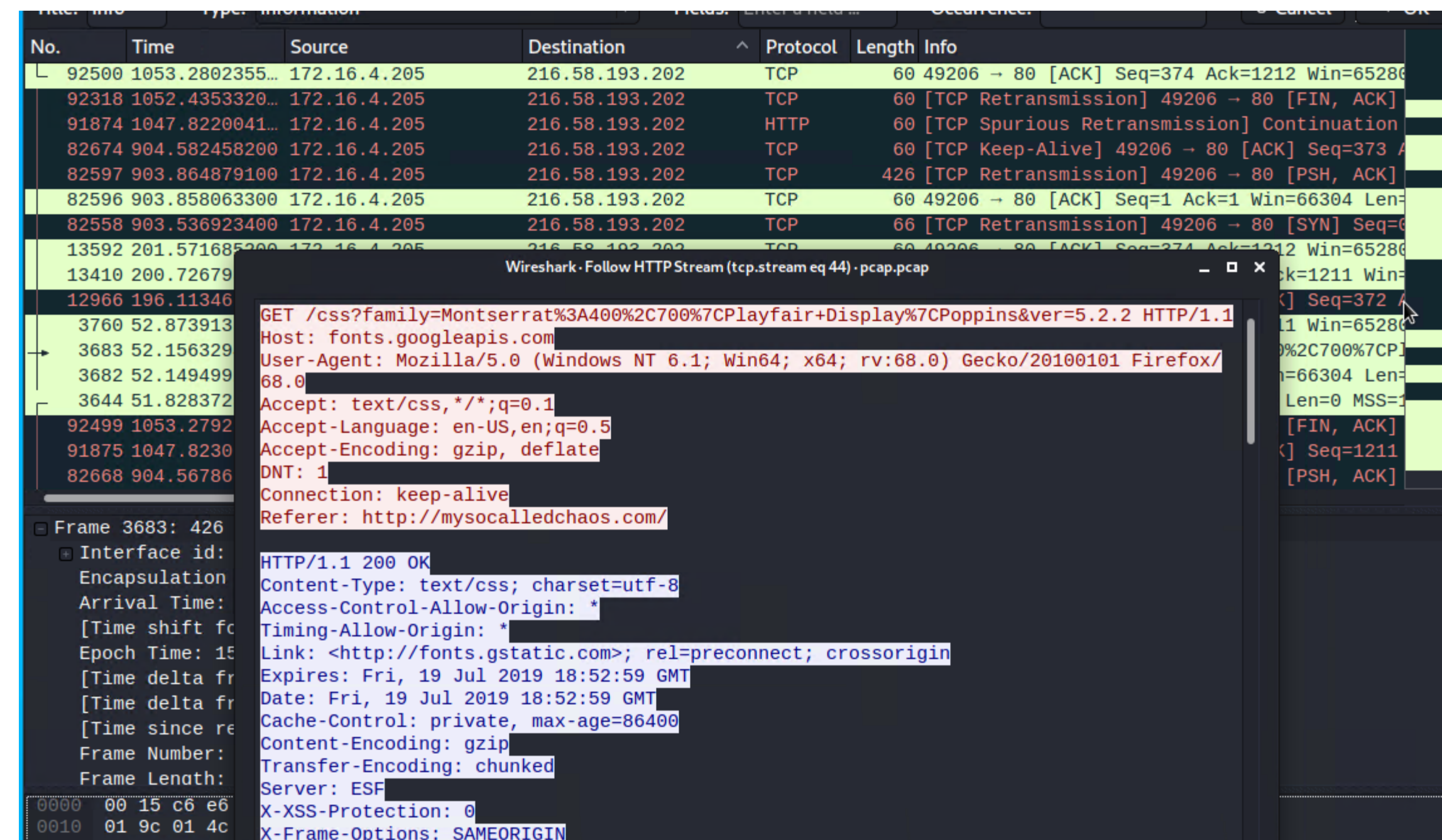


Normal Activity

Watching YouTube

Summarize the following:

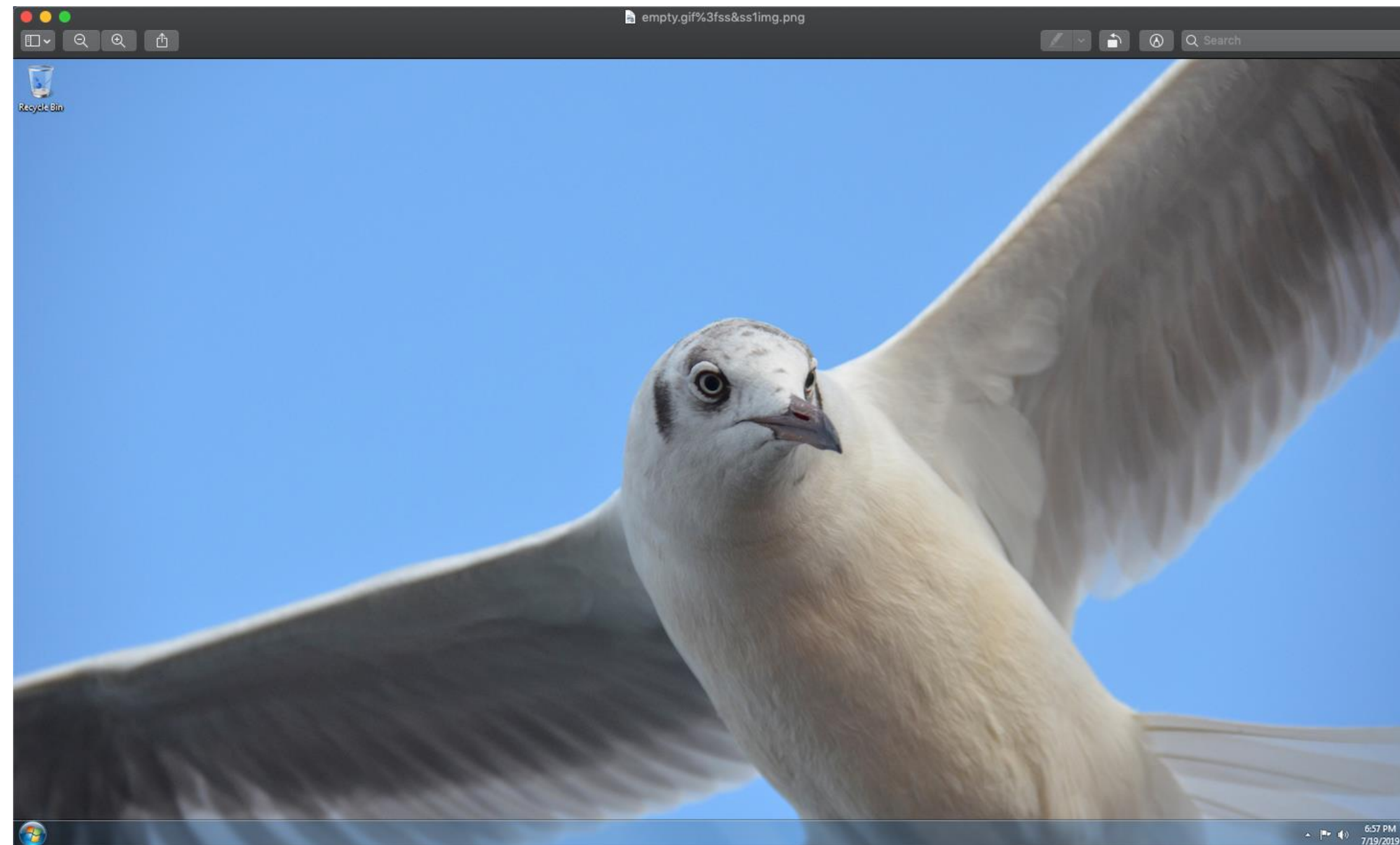
- They had a lot of traffic to YouTube IP addresses using protocols like TCP and HTTP
- The users were steaming packets from specific youtube IP addresses like 216.52.193.200 & 202
- References to <http://mysocalledchaos.com/>



Installing desktop backgrounds

Summarize the following:

- The user was observed downloading and installing an img file (empty.gif?ss&ss1img).
- The file came from a site named green.mattingssolutions.co.
- The image was used has a desktop background.



Malicious Activity

Set up AD network and domain controller

Summarize the following:

- We observe the client and server passing DNS, DHCP and LDAP protocols.
- The client machine DESKTOP-86J4BX authenticated to the Frank-n-ted.com domain.
- This was a domain setup inside the corporate domain.

No.	Time	Source	Destination	Protocol	Length	Info
82108	901.503341900	172.16.4.205	172.16.4.4	TCP	56	[TCP Retransmission] 49163 → 88 [FIN, ACK] Seq=24
82106	901.495079100	172.16.4.205	172.16.4.4	TCP	297	[TCP Retransmission] 49163 → 88 [PSH, ACK] Seq=1
82105	901.490329100	172.16.4.205	172.16.4.4	TCP	56	49163 → 88 [ACK] Seq=1 Ack=1 Win=65536 Len=0
82103	901.488376400	172.16.4.205	172.16.4.4	TCP	68	[TCP Retransmission] 49163 → 88 [SYN] Seq=0 Win=0
3189	49.792925100	172.16.4.205	172.16.4.4	TCP	56	49163 → 88 [FIN, ACK] Seq=244 Ack=290 Win=65280
3187	49.786544600	172.16.4.205	172.16.4.4	KRB5	297	AS-REQ
3186	49.781929800	172.16.4.205	172.16.4.4	TCP	56	49163 → 88 [ACK] Seq=1 Ack=1 Win=65536 Len=0
3184	49.779832900	172.16.4.205	172.16.4.4	TCP	68	49163 → 88 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS
82112	901.505332800	172.16.4.4	172.16.4.205	TCP	54	88 → 49163 [RST, ACK] Seq=290 Ack=245 Win=0 Len=0
82111	901.504469200	172.16.4.4	172.16.4.205	TCP	54	88 → 49163 [ACK] Seq=290 Ack=245 Win=131328 Len=0
82107	901.503336200	172.16.4.4	172.16.4.205	TCP	343	[TCP Retransmission] 88 → 49163 [PSH, ACK] Seq=1
82104	901.489436600	172.16.4.4	172.16.4.205	TCP	66	[TCP Retransmission] 88 → 49163 [SYN, ACK] Seq=0
3193	49.796795800	172.16.4.4	172.16.4.205	TCP	54	88 → 49163 [RST, ACK] Seq=290 Ack=245 Win=0 Len=0
3192	49.795970300	172.16.4.4	172.16.4.205	TCP	54	88 → 49163 [ACK] Seq=290 Ack=245 Win=131328 Len=0
3188	49.792033500	172.16.4.4	172.16.4.205	KRB5	343	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
3185	49.781033200	172.16.4.4	172.16.4.205	TCP	54	88 → 49163 [ACK] Seq=290 Ack=245 Win=131328 Len=0

Wireshark · Follow TCPStream (tcp.stream eq 10) · pcap.pcap

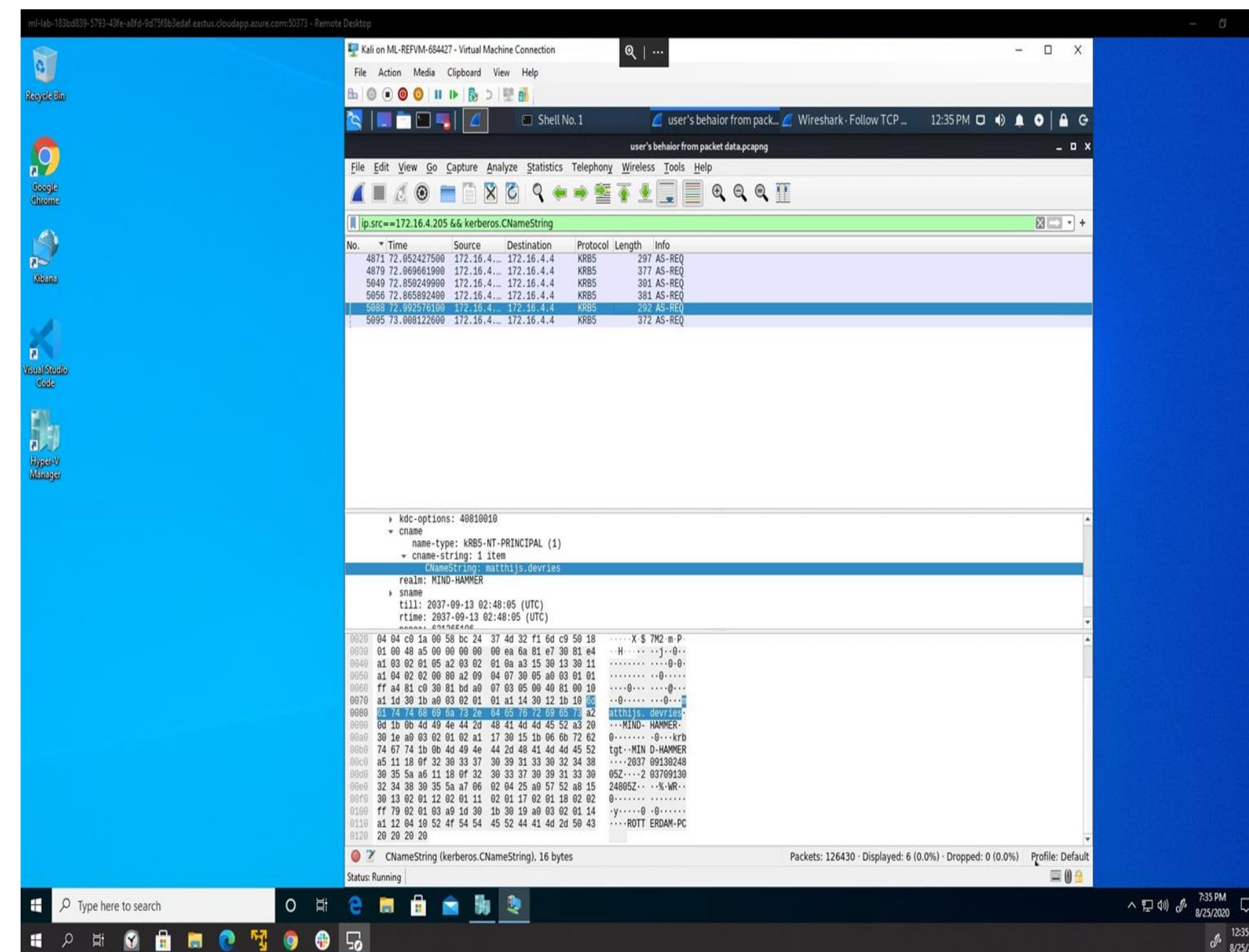
Frame 3193: 54 bytes on wire (432 bits) captured on interface 0 (eth0) 159.796795800 seconds
Interface id: 0
Encapsulation type: Ethernet II (Ethernet)
Arrival Time: 159.796795800 seconds
[Time shift for this capture: 0.000000000 seconds]
Epoch Time: 159.796795800 seconds
[Time delta from previous capture: 0.000000000 seconds]
[Time delta from previous capture: 0.000000000 seconds]
[Time since reference capture: 0.000000000 seconds]

rotterdam-pc\$...MIND-HAMMER.NET.\$0".....0...krbtgt..MIND-HAMMER.NET....
20370913024805Z....20370913024805Z.....J'2..0.....y.....
0.0.....ROTTERDAM-PC.....0.....20190719185232Z....
7t.....MIND-HAMMER.NET.\$0".....0...krbtgt..MIND-HAMMER.NET.....0..0.....}
0{08.....1./MIND-HAMMER.NEThostrotterdam-pc.mind-hammer.net0.....08.....1./MIND-
HAMMER.NEThostrotterdam-pc.mind-hammer.net0.....0.....0

Downloading malware

Summarize the following:

- We observed some HTTP traffic download sum suspect traffic.
- The user Matthijs.devries downloaded some malware form the address 182.243.115.84 containing a file june11.dll
- The file contain 68 malware binaries, including multiple trojans.





The End