Lab 4: Network Monitoring and Security Awareness

By: Alex Diaz, Jonathan Ramos, Hubert Meneses, Andres Ulloa

Introduction

Goals

Goal 1 (Youtube Traffic/IP/MAC ADDRESS/PORTS)

Goal 2 (Simulate Ping Flood Attack)

Goal 3 (Port Scanning)

Goal 4 (Identify traffic anomalies)

Challenges Faced

References /QA

Goals

1) Capture network traffic using Wireshark and analyze IP addresses, MAC addresses, and protocols.

2) Simulate a ping flood attack and observe network behavior.

3) Detect port scanning attempts using nmap.

4) Identify traffic anomalies such as high volume, unknown IPs, and unusual packet sizes.

Goal 1

We first were tasked with was to capture and analyze packets through wireshark.

For this we decided to do it using the platform Youtube. We ran a 40 minute long video for 3 minutes to capture and observe the packets, addresses and protocols.

Youtube Traffic Filters

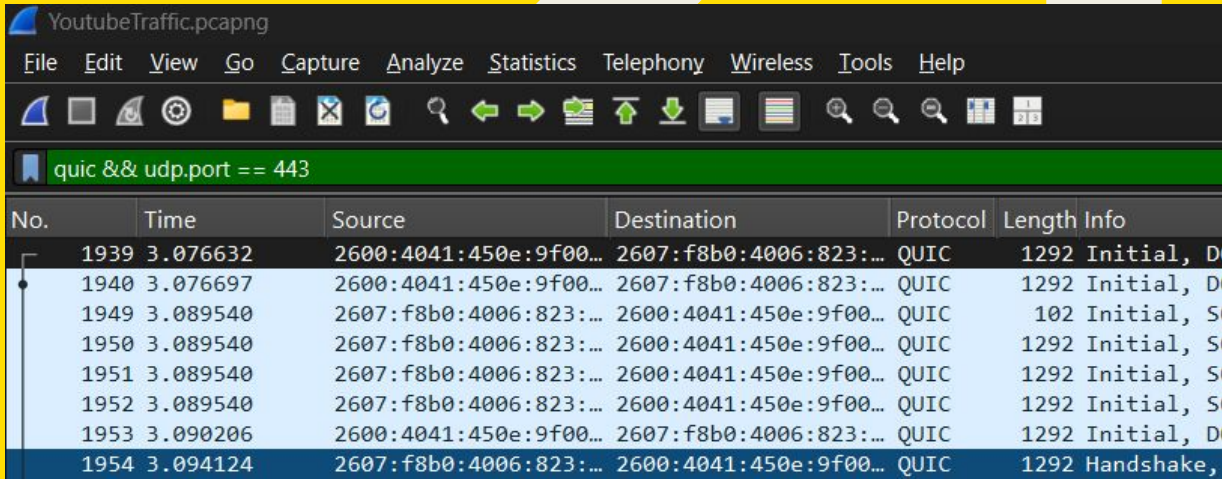
Once we had watched our selected Youtube video for 3 minutes we had to find out exactly which packets were coming from Youtube specifically. This was done by filtering the packets by specific protocols and ports.

Filter 1: Only QUIC streaming traffic

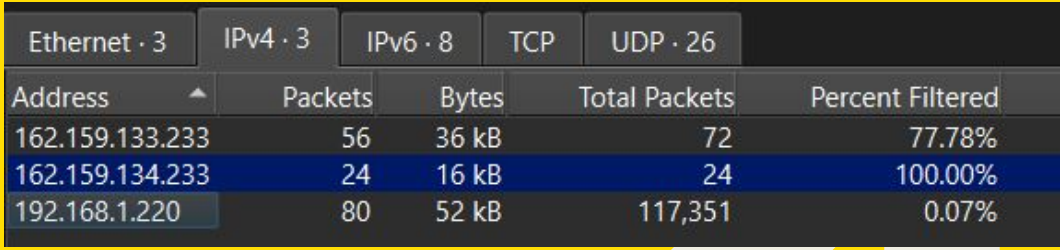
quic && udp.port == 443

- QUIC is the type of protocol Youtube's packets use
- The Destination Port to Youtube is 443

By filtering the information we were able to see solely Youtubes traffic which would let us observe the upcoming information.



Youtube Traffic IPs



When going into the Statistics then endpoint settings within wireshark we were able to see all of the available IP addresses that were being used while the filter in the past slide was still on. This was done by viewing the **IPv4 tab**.

To find out where these IP addresses were from we went to **https://whois.domaintools.com** Which let us input these IPs to find their source.

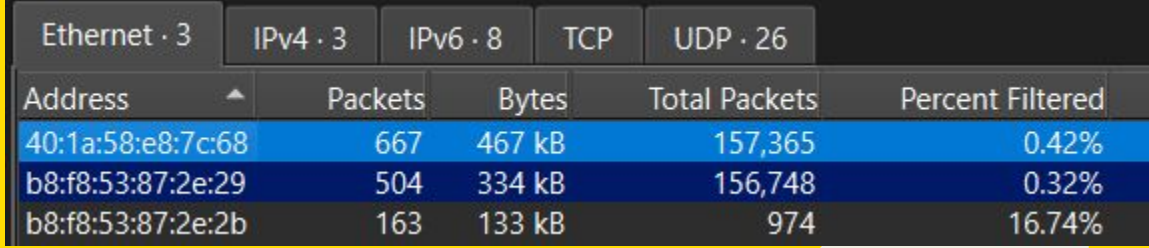
We found out that:

162.159.133.233 & **162.159.134.233** were both IP addresses coming from **CloudFlare Inc.**

The **192.168.1.220** number was **our own device's IP** with packets from all other devices that were on the Wifi
Network.

CloudFlare is known to be used with with Youtube a lot for video delivery, this told us that the packets for Youtube specifically were being used within our Test

Youtube Traffic MAC Addresses



While still in the statistics and endpoint settings within wireshark we were able to see all of the available MAC addresses that were being used while the filter in the past slide was still on. This was done by viewing the Ethernet Tab.

MAC address **40:1a:58:e8:7c:68** belonged to our own computer's Wifi network adapter, which created the streaming traffic.

MAC address **b8:f8:53:87:2e:29 && b8:f8:53:87:2e:2b** shared the same manufacturer and represented the different interfaces on our home router.

Since MAC addresses only exist on the local network (Layer 2), this confirms to us that all YouTube traffic captured was moving between our device and the router before being forwarded to the external IP addresses that were shown in the slides prior.

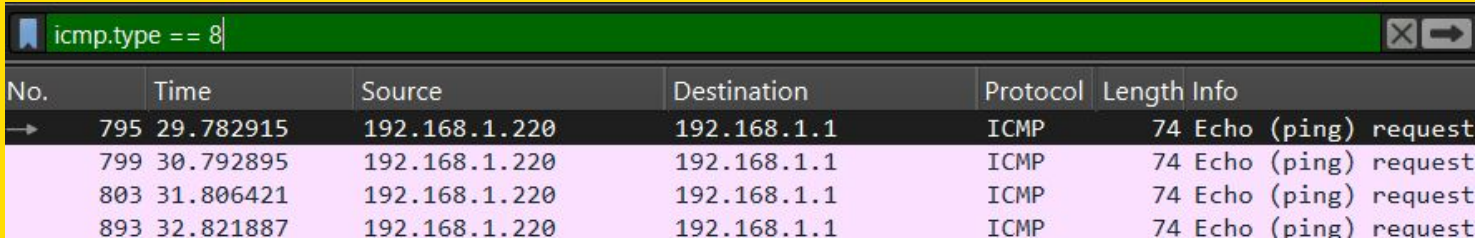
Goal 2
Flood Ping Attack

Since we found it difficult to try a ping flood attack using the Youtube video being ran at the same time we decide to just run our network normally without a video and simulated it that way

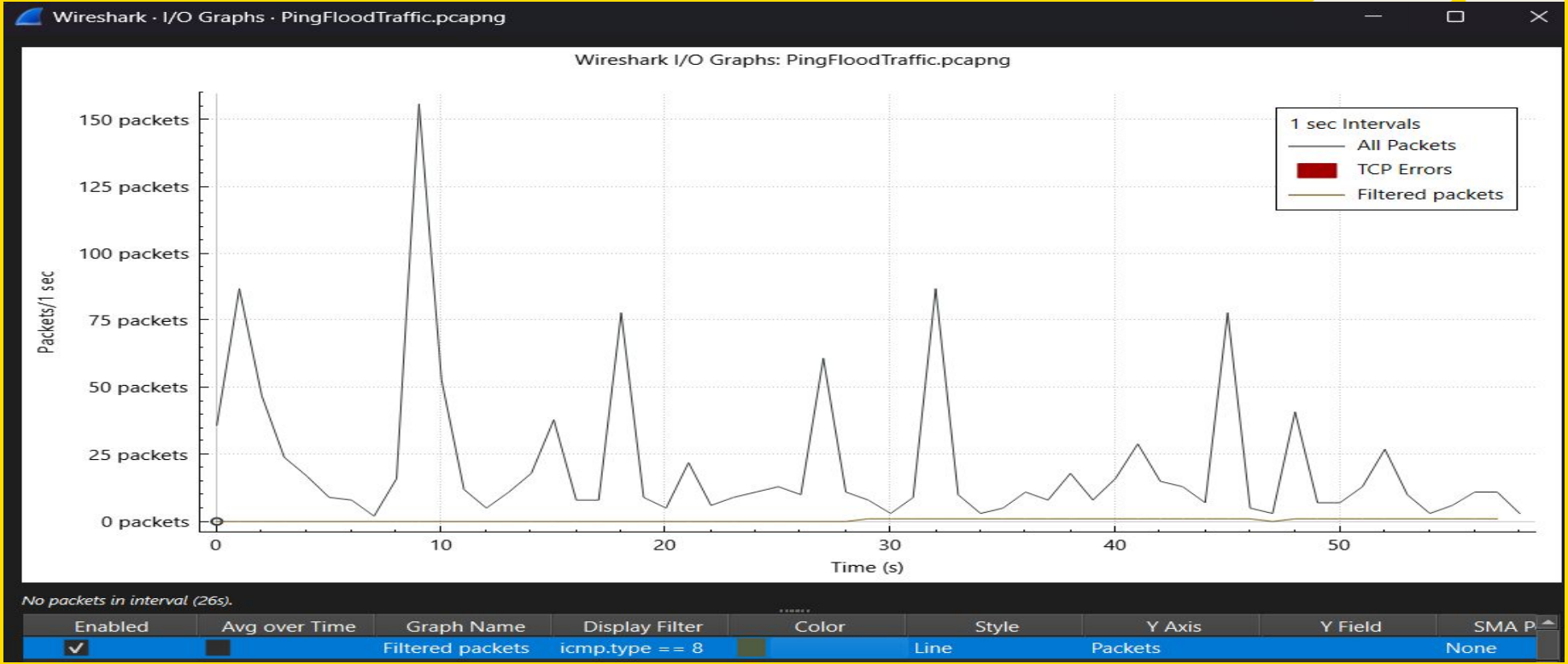
- We opened up command prompt and used the command **ping -t 192.168.1.1.** At the same time the wireshark packet tracer was on and running,

- in real time we saw a bunch of ICMP echo requests being sent across the network. The packets appeared at a very high rate and it intentionally generated traffic where continuous ICMP traffic would overwhelm a network if ran long enough.

We then filtered it once more using the filter **icmp.type == 8** so it would only show the ICMP echo request packets that were done through the ping flood attack.



Goal 2
Flood Ping Attack

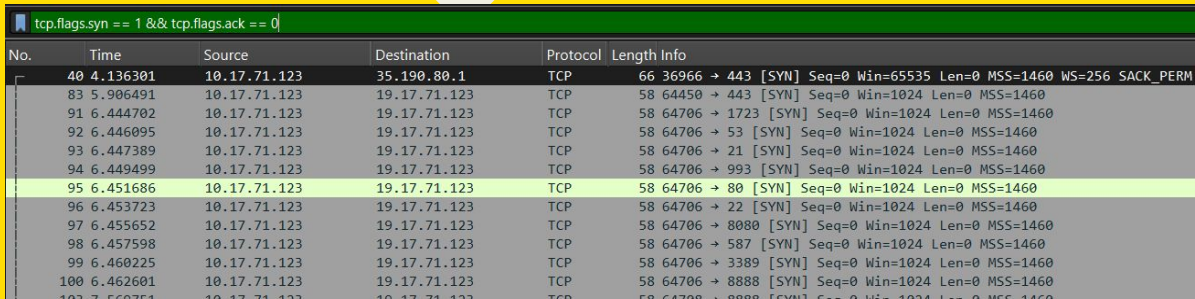


Goal 3 (Detect port scanning attempts using nmap.)

- We performed a SYN port scan using Nmap while Wireshark was capturing packets. The command **nmap -sS <our IP>** was used to see the open ports within the network.

- Wireshark displayed a large number of TCP SYN packets sent to different port numbers, which is know to be common in a port scanning attempt.

- Using the filter **tcp.flags.syn == 1 && tcp.flags.ack == 0**, we were able to isolate these NMAP SYN packets and observe its behavior. Open ports responded with SYN/ACK packets, while closed ports returned RST packets. This pattern confirms the presence of a port scan, overall it represents the type of packets an attacker might use before attempting this scan attempt.



Goal 4 (Identify traffic anomalies such as high volume, unknown IPs, and unusual packet sizes.)

During the YouTube traffic capture, the packets appeared normal in terms of regular web streaming. The protocols observed included QUIC, TLS, TCP, and UDP, all flowing normally between our device and the Google/Cloudflare servers. Packet sizes and timing were stable, indicating to us normal browsing and video playback behavior.

The IPs we observed were the IPs **162.159.133.233** and **162.159.134.233**, they were initially unknown, but through using a WHOIS lookup tool, we identified both addresses as belonging to Cloudflare which gave us the conclusion that there were no malicious anomalies here..

In contrast, the ping flood traffic showed clear anomalies. Wireshark displayed hundreds of ICMP Echo Request packets, and the I/O graph showed high volume with large spikes in packets per second, which is not normal in usual traffic.

With Nmap, we observed bursts of TCP SYN packets being sent to many different port numbers rapidly. Some ports responded with SYN/ACK (open), while most returned RST (closed).

By comparing these captures, normal traffic has steady, packet flow with consistent protocols, while non normal traffic shows either high volume (ping flood) or a wide range of targeted ports (port scan). These anomalies highlight how Wireshark can be used to see how routine network usage can potentially be malicious activity.

Challenges faced

- We were unable to identify which IP and MAC address was either Youtube or our own device we were running the tests on. To fix this, we used an IP Domain lookup site as well as a MAC address lookup site which gave us the information needed to understand where these IPs were located.

- We were unsure on how to correctly showcase our flood ping attack that occurred in our test. The solution to our issue was finding the I/O Graph within wireshark that displayed the overall ICMP echo request packets being sent.

- We did not know how to go about using nmap in this part of the project since no one had it, our solution was installing it as well as researching commands nmap uses which we then implemented into our work when scanning for traffic

References

https://whois.domaintools.com

https://maclookup.app

https://www.youtube.com/watch?v=a3JIPHlifzE

Q/A ?