NETWORK SECURITY

EXPERIMENT REPORT

Course Title and Number:

Computer Networks

Project Title:

Packet Analysis of Facebook and YouTube Traffic

Virtual private network (VPN) For Remote Access

Group Members:

M FUZAIL RAZA (39091)

M WAQAS ZAFAR (38605) | Abdul Quds (36154)

Riphah International University, Lahore

Submission Deadline:

17 January 2023.

WIRESHARK REPORT

WIRESHARK:

Wireshark is a network protocol analyzer that allows you to capture and inspect the data traveling back and forth on a computer network in real-time. It is an open-source tool commonly used for network troubleshooting, analysis, software development, and education. Wireshark supports a wide range of network protocols and can display the captured data in a detailed and readable format.

Usage:

1. **Network Troubleshooting:** Wireshark is often used to identify and troubleshoot network issues by analyzing the packets exchanged between devices.
2. **Security Analysis:** Security professionals use Wireshark to detect and analyze malicious activities on a network, such as unauthorized access or suspicious traffic patterns.
3. **Protocol Development:** Developers use Wireshark to understand and debug the communication between different devices or software components by examining the protocol messages.
4. **Educational Purposes:** Wireshark is commonly used in networking courses and training programs to teach students about network protocols, packet analysis, and network security.

5. **Performance Optimization:** It helps in optimizing network performance by identifying bottlenecks, latency issues, or inefficient data transfers.

Experiment Location:

Conducted at: Home

Type of Computer:

Hardware:	AMD A8-4500M APU with Radeon(tm) HD Graphics (with SSE4.2)
OS:	64-bit Windows 10 (22H2), build 19045
Application:	Dump cap (Wireshark) 4.2.0 (v4.2.0-0-g54eedfc63953)

CAPTURE FILE PROPERTIES:

Name:	C:\Users\MFC~1\AppData\Local\Temp\wireshark_Wi-FiT67TG2.pcapng
Length:	18 MB
Hash (SHA256):	cace611c5799d14cbb473a84647362a02a8dfe4a4ee6027510acc3d254daa2b1
Hash (SHA1):	34e16cabb90969af3a98f59b12efacb05e926c04
Format:	Wireshark/... - pcapng
Encapsulation:	Ethernet

Time

First packet: 2024-01-07 15:49:33

Last packet: 2024-01-07 16:23:33

Elapsed: 00:34:00

Capture

Hardware:	AMD A8-4500M APU with Radeon(tm) HD Graphics (with SSE4.2)
OS:	64-bit Windows 10 (22H2), build 19045
Application:	Dumpcap (Wireshark) 4.2.0 (v4.2.0-0-g54eedfc63953)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
Wi-Fi	0 (0.0%)	none	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	28321	28321 (100.0%)	—
Time span, s	2040.020	2040.020	—
Average pps	13.9	13.9	—
Average packet size, B	634	634	—
Bytes	17946836	17946836 (100.0%)	0

Average bytes/s	8797	8797	—
Average bits/s	70 k	70 k	—

Conversation:

The overall packet send and receive across the networks:

Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s
192.168.181.113	51724	157.240.227.21	443	608	270 kB	1	298	249 kB	310	21 kB	0.088081	97.3997	20
192.168.181.113	51732	157.240.227.56	443	111	100 kB	2	36	2 kB	75	98 kB	0.252091	65.1703	251
192.168.181.113	51735	157.240.227.1	443	408	204 kB	3	159	15 kB	249	188 kB	1.020435	99.5675	1230
192.168.181.113	51697	51.89.98.179	443	40	2 kB	4	22	1 kB	18	1 kB	1.219286	98.0634	97
192.168.181.113	51722	157.240.227.1	443	31	2 kB	5	18	1 kB	13	898 bytes	5.283998	95.6346	111
192.168.181.113	51723	157.240.227.1	443	42	4 kB	6	23	2 kB	19	2 kB	5.790833	94.1339	134
192.168.181.113	51728	157.240.227.21	443	47	7 kB	7	25	5 kB	22	1 kB	8.255262	91.9214	445
20.192.44.78	443	192.168.181.113	50608	1	290 bytes	8	1	290 bytes	0	0 bytes	13.034955	0.0000	
192.168.181.113	51731	20.192.44.78	443	32	2 kB	9	19	2 kB	13	872 bytes	13.498653	84.8142	146
192.168.181.113	51707	173.194.76.188	5228	7	404 bytes	10	4	218 bytes	3	186 bytes	18.909408	60.3714	28
192.168.181.113	51713	157.240.227.1	443	5	271 bytes	11	3	163 bytes	2	108 bytes	21.111483	15.2103	85
192.168.181.113	51716	34.159.167.110	443	10	640 bytes	12	5	295 bytes	5	345 bytes	27.784035	30.1728	78
20.198.118.190	443	192.168.181.113	51700	3	673 bytes	13	2	486 bytes	1	187 bytes	28.075202	0.1437	27
192.168.181.113	51718	111.119.184.209	443	7	380 bytes	14	4	218 bytes	3	162 bytes	28.304989	48.7395	35
192.168.181.113	51736	157.240.227.61	443	61	12 kB	15	27	2 kB	34	10 kB	28.423005	4.0036	4485
192.168.181.113	51719	121.91.41.83	443	7	380 bytes	16	4	218 bytes	3	162 bytes	28.888005	48.1572	36
192.168.181.113	51720	121.91.40.83	443	7	380 bytes	17	4	218 bytes	3	162 bytes	29.157989	47.8872	36
192.168.181.113	51740	157.240.227.60	443	131	71 kB	18	69	5 kB	62	66 kB	33.524863	1028.9215	36
192.168.181.113	51739	163.70.137.60	443	32	6 kB	19	20	2 kB	12	4 kB	33.524863	1042.8886	13
192.168.181.113	51741	157.240.227.60	443	30	6 kB	20	18	2 kB	12	4 kB	33.525438	1037.2727	12
192.168.181.113	51742	163.70.132.60	443	32	6 kB	21	20	2 kB	12	4 kB	33.525548	1046.4446	13
192.168.181.113	51743	121.91.41.99	443	28	6 kB	22	18	1 kB	10	4 kB	33.526014	1029.7010	11
192.168.181.113	51744	111.119.184.224	443	47	8 kB	23	26	2 kB	21	5 kB	33.526182	1029.9067	17
192.168.181.113	51745	157.240.227.60	443	28	5 kB	24	19	2 kB	9	4 kB	33.526353	1034.9097	11
192.168.181.113	51746	121.91.41.99	443	47	8 kB	25	26	2 kB	21	5 kB	33.526607	1029.8896	17
192.168.181.113	51747	121.91.40.99	443	29	6 kB	26	19	2 kB	10	4 kB	33.526853	1029.4525	11
192.168.181.113	51748	111.119.184.224	443	28	6 kB	27	18	1 kB	10	4 kB	33.527173	1029.7172	11
192.168.181.113	51749	121.91.41.99	443	29	6 kB	28	19	2 kB	10	4 kB	33.527535	1029.7169	11
192.168.181.113	51750	121.91.40.99	443	42	7 kB	29	23	2 kB	19	5 kB	33.527989	1024.6408	16
192.168.181.113	51751	111.119.184.224	443	29	6 kB	30	19	2 kB	10	4 kB	33.528381	1029.8877	11
192.168.181.113	51752	121.91.41.99	443	30	6 kB	31	19	2 kB	11	4 kB	33.528715	1024.6402	11
192.168.181.113	51753	121.91.40.99	443	30	6 kB	32	19	2 kB	11	4 kB	33.529749	1031.7567	11
192.168.181.113	51754	111.119.184.224	443	30	6 kB	33	20	2 kB	10	4 kB	33.529934	1029.4495	12
192.168.181.113	51755	121.91.40.99	443	28	6 kB	34	18	1 kB	10	4 kB	33.530101	1028.3380	11
192.168.181.113	51756	157.240.227.61	5222	30	3 kB	35	13	1 kB	17	2 kB	33.597778	15.0434	608
192.168.181.113	51717	157.240.227.13	443	9	526 bytes	36	5	271 bytes	4	255 bytes	37.536979	21.3089	101
192.168.181.113	51726	20.212.88.117	443	12	855 bytes	37	10	723 bytes	2	132 bytes	40.740910	92.1950	62

WIRESHARK FILTERS:

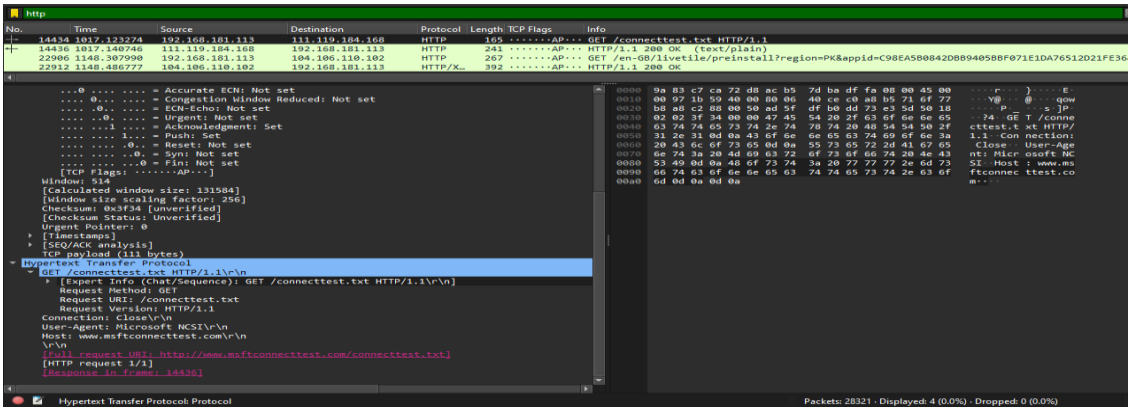
FACEBOOK:

HTTP PACKETS:

Total displayed = 4 Packets

Filter: http

The filter **http** is applied to display HTTP packets, revealing a total of 4 packets in the captured network traffic



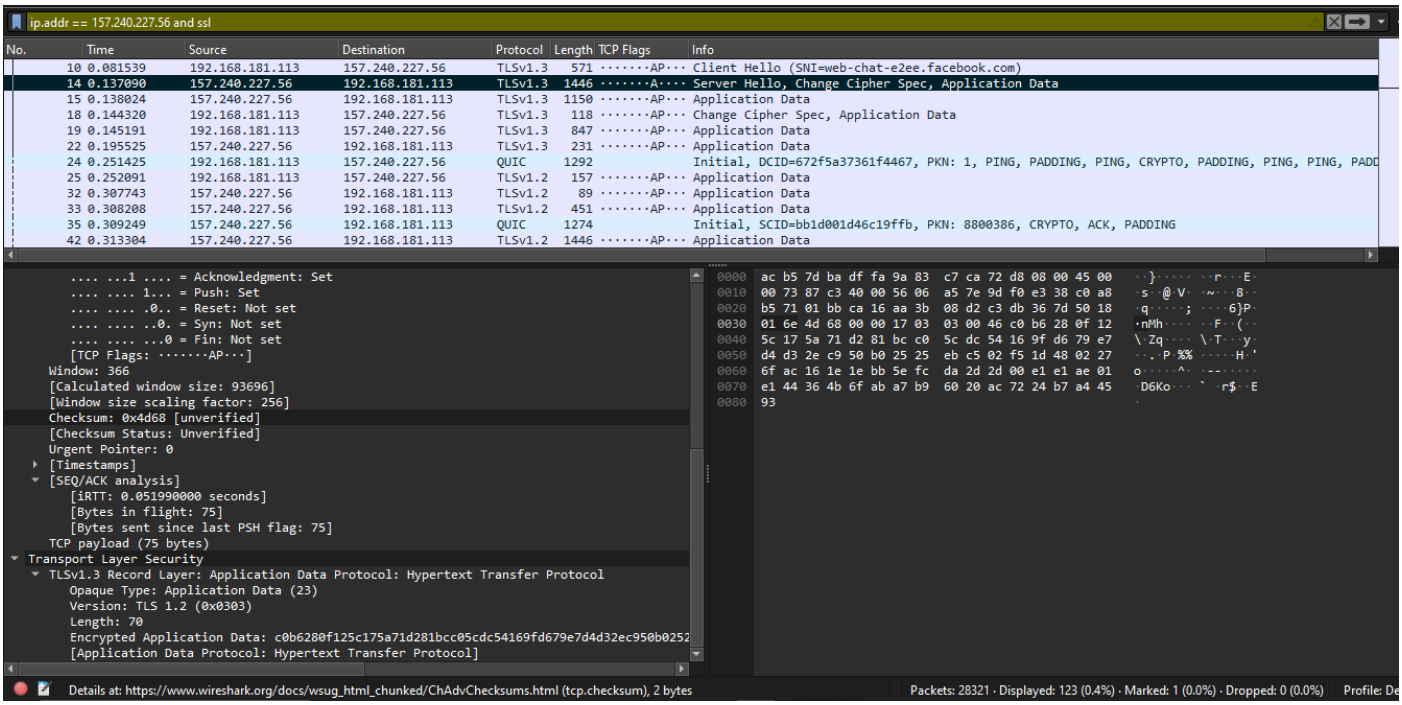
SSL PACKETS:

The filter `**ip.addr == 157.240.227.56 and ssl**` is used to capture SSL packets associated with the IP address **157.240.227.56**. In the captured data, 123 bytes are displayed, constituting **0.4%** of the total traffic.

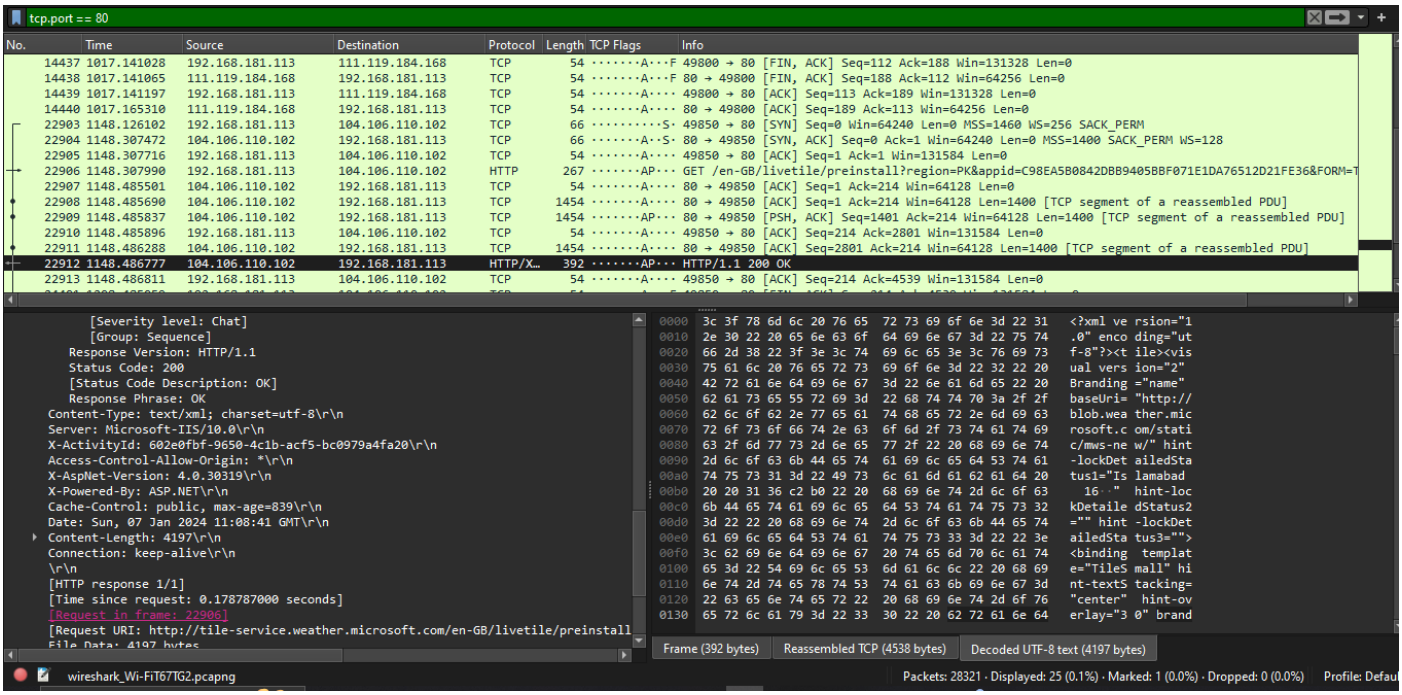
ip.addr == 157.240.227.56 and ssl

Displayed: 123 bytes

Fraction: 0.4%



Then use display filters to separate the subset of TCP packets that are also HTTP packets.

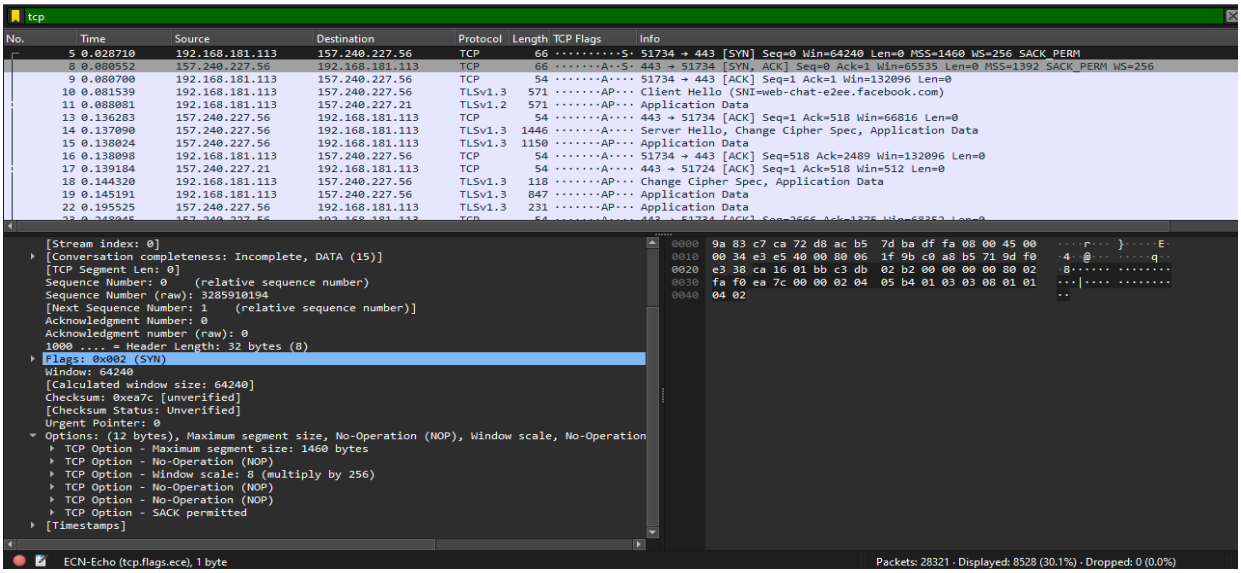


Displayed: 25 bytes

Fraction: 0.1%

TCP to/from Facebook:

The filter **`tcp`** is applied to capture TCP packets to and from the IP address 157.240.227.56. In total, 8528 packets matching this filter are displayed in the captured network traffic.



Filter from src to destination:

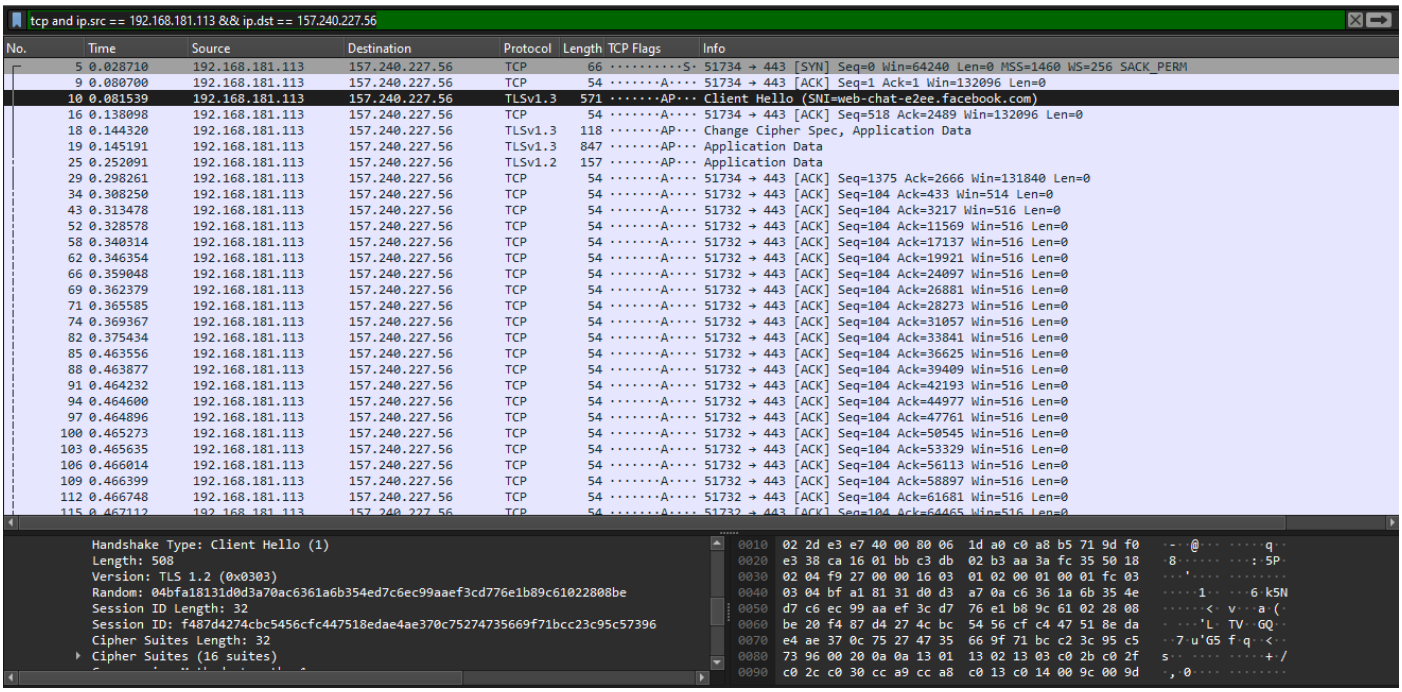
FROM MY PC TO FACEBOOK:

The filter `**tcp and ip.src == 192.168.181.113 && ip.dst == 157.240.227.56**` is applied to capture TCP packets sent from your computer (source IP: 192.168.181.113) to Facebook (destination IP: 157.240.227.56). A total of 148 packets matching this filter are displayed, constituting 0.5% of the total traffic.

tcp and ip.src == 192.168.181.113 && ip.dst == 157.240.227.56

Total displayed: 148

Fraction: 0.5%

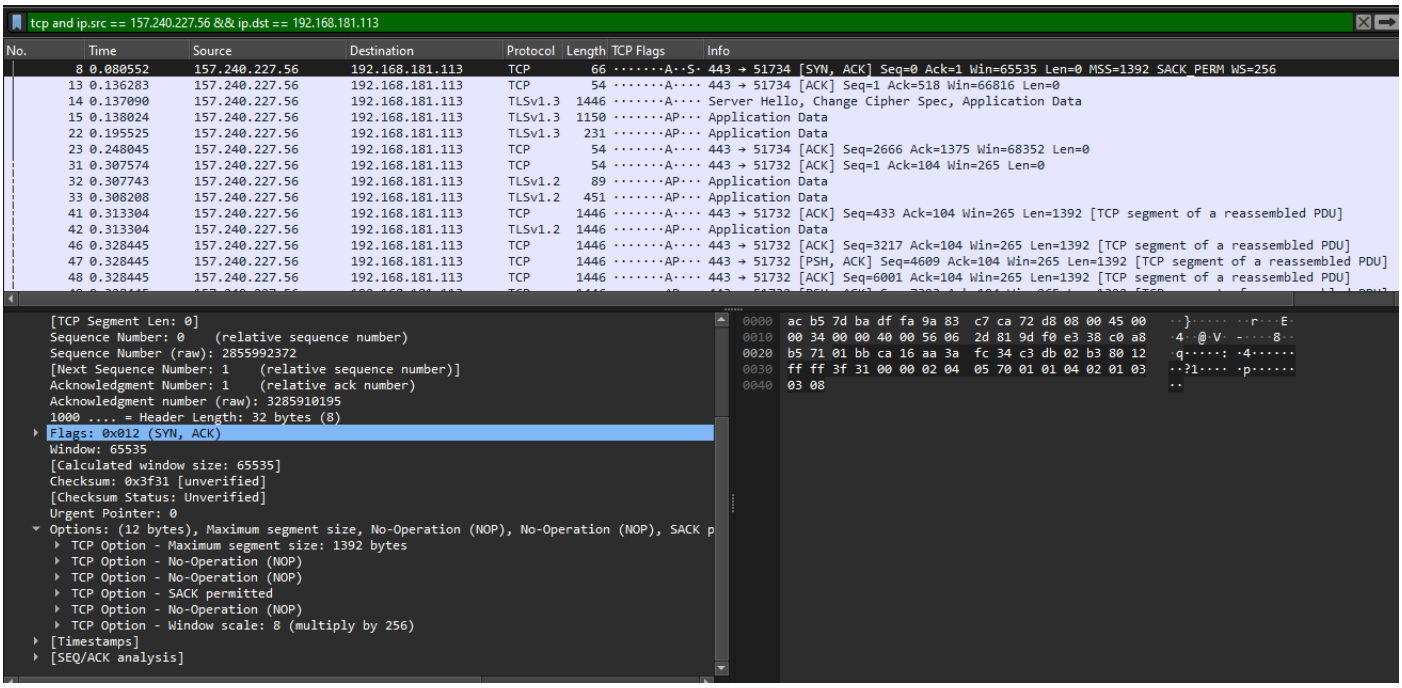


From Facebook to Computer:

The filter `**tcp and ip.src == 157.240.227.56 && ip.dst == 192.168.181.113**` is used to capture TCP packets sent from Facebook (source IP: 157.240.227.56) to your computer (destination IP: 192.168.181.113). A total of 186 packets matching this filter are displayed, constituting 0.7% of the total traffic.

tcp and ip.src == 157.240.227.56 && ip.dst == 192.168.181.113

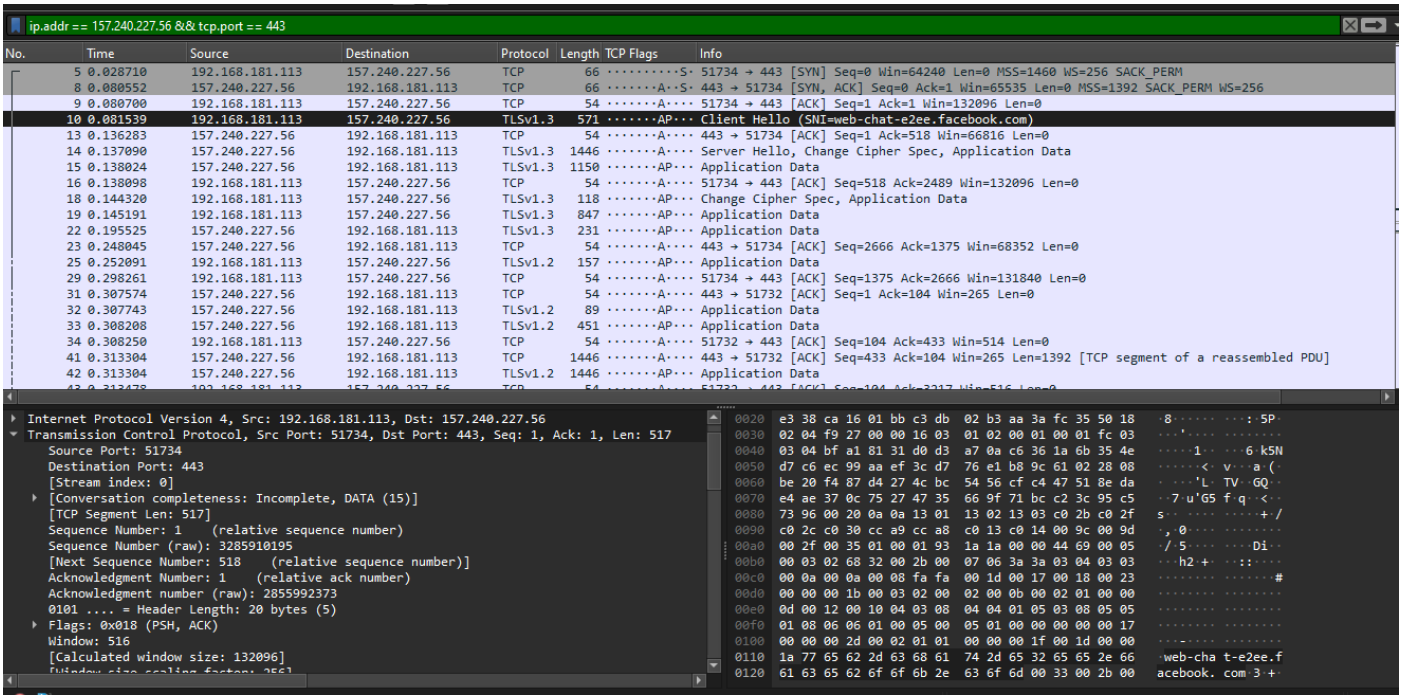
Total Displayed: 186 (0.7%)



HTTP to/from facebook:

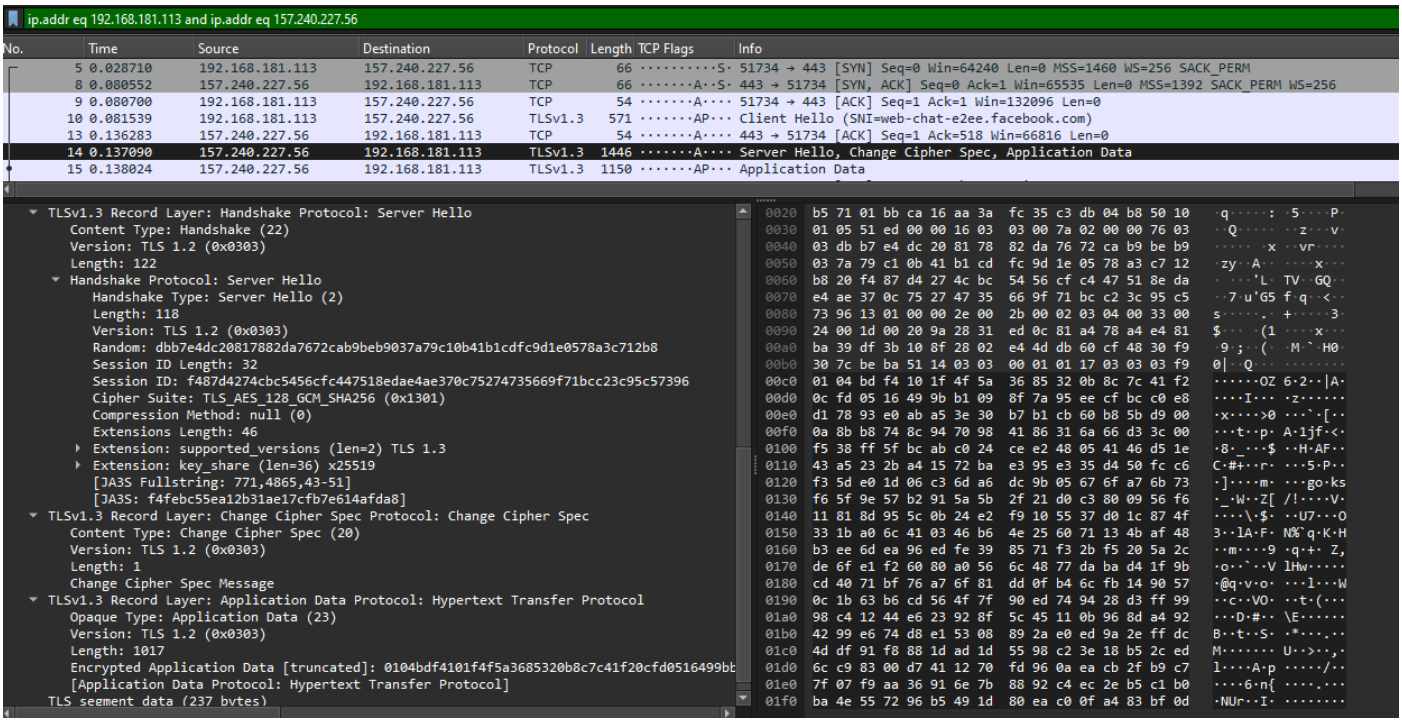
The filter `**ip.addr == 157.240.227.56 && tcp.port == 443**` is applied to capture HTTP packets to and from Facebook on port 443 (HTTPS). A total of 334 packets matching this filter are displayed, constituting **1.2%** of the total traffic. The source port is 51734, and the destination port is 443.

Total displayed: 334 packets (1.2%)



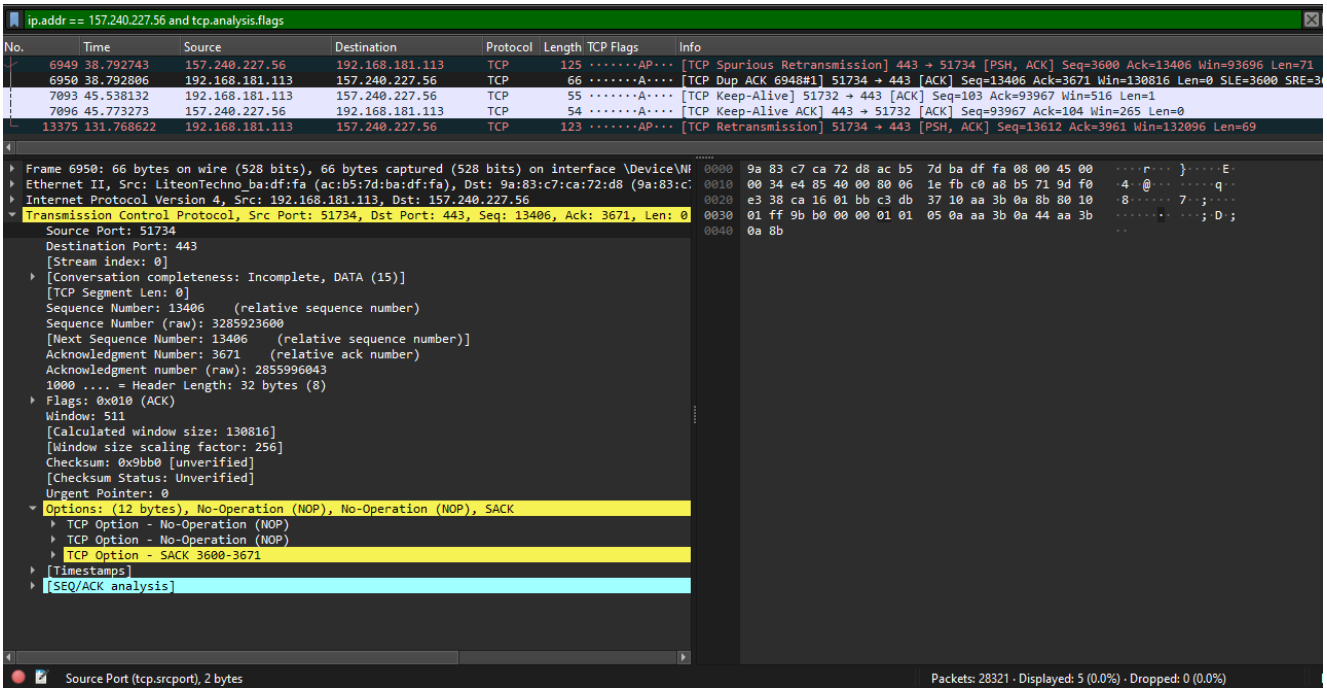
Conversation filter between pc to Facebook:

```
ip.addr eq 192.168.181.113 and ip.addr eq 157.240.227.56
```



FLAGS:

The filter **ip.addr == 157.240.227.56 and tcp.analysis.flags** in Wireshark is designed to display TCP packets to and from the IP address 157.240.227.56 while also showing additional information related to TCP flags analysis.



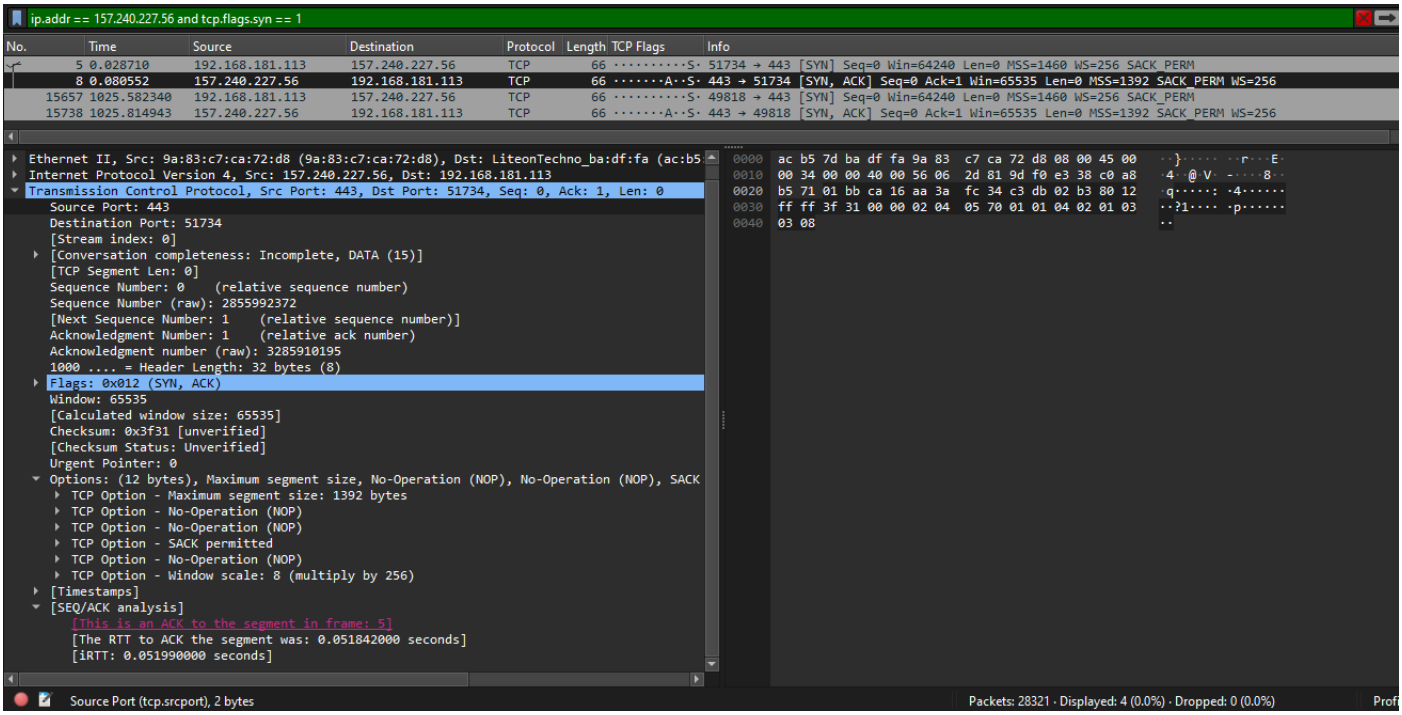
SYN FLAG:

The filter `**ip.addr == 157.240.227.56 and tcp.flags.syn == 1**` is used to capture TCP packets with the SYN flag set sent to or from the IP address 157.240.227.56. A total of 4 packets matching this filter are displayed, constituting 0.0% of the total 28321 SYN-flagged packets.

ip.addr == 157.240.227.56 and tcp.flags.syn == 1

Displayed: 4 packets

Fraction: 0.0% because total are 28321

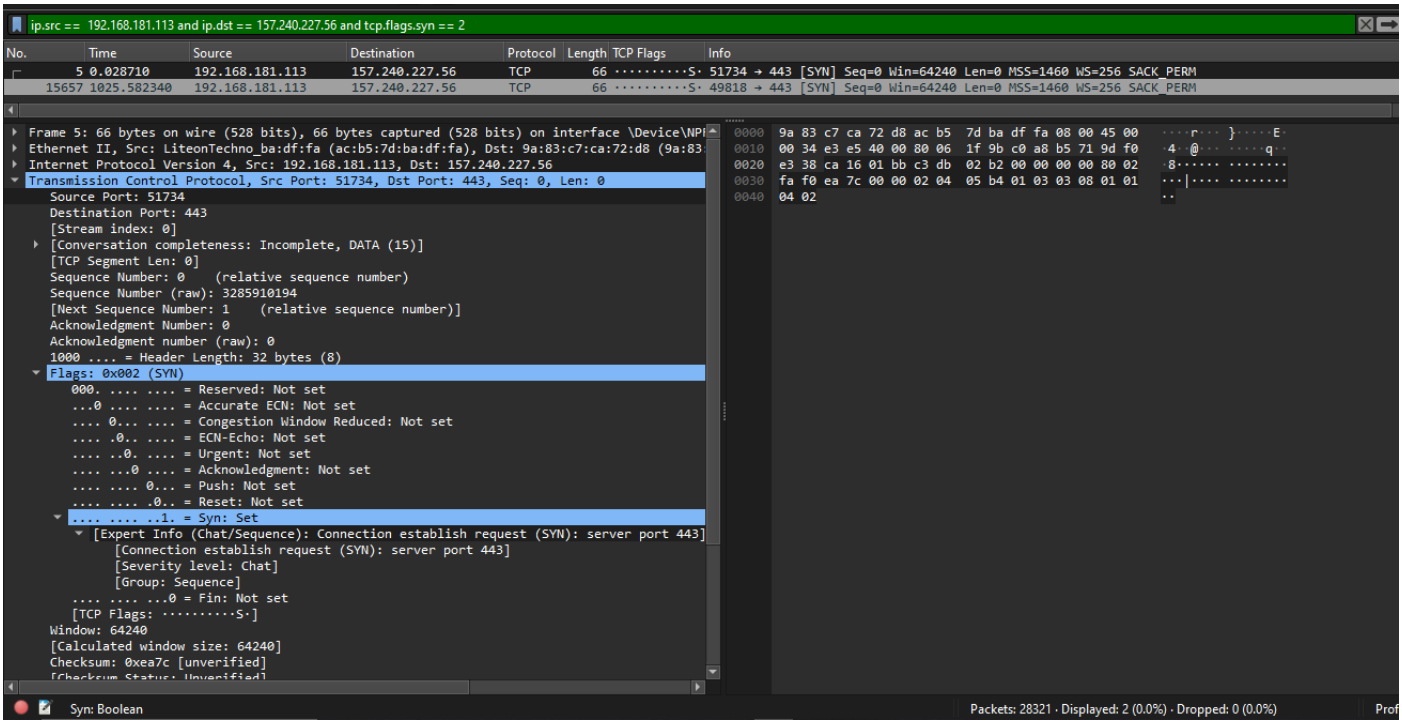


From Pc to Facebook:

```
ip.src ==  192.168.181.113 and ip.dst ==
157.240.227.56 and tcp.flags.syn == 2
ip.src ==  192.168.181.113 and ip.dst ==
157.240.227.56 and tcp.flags.syn == 3
```

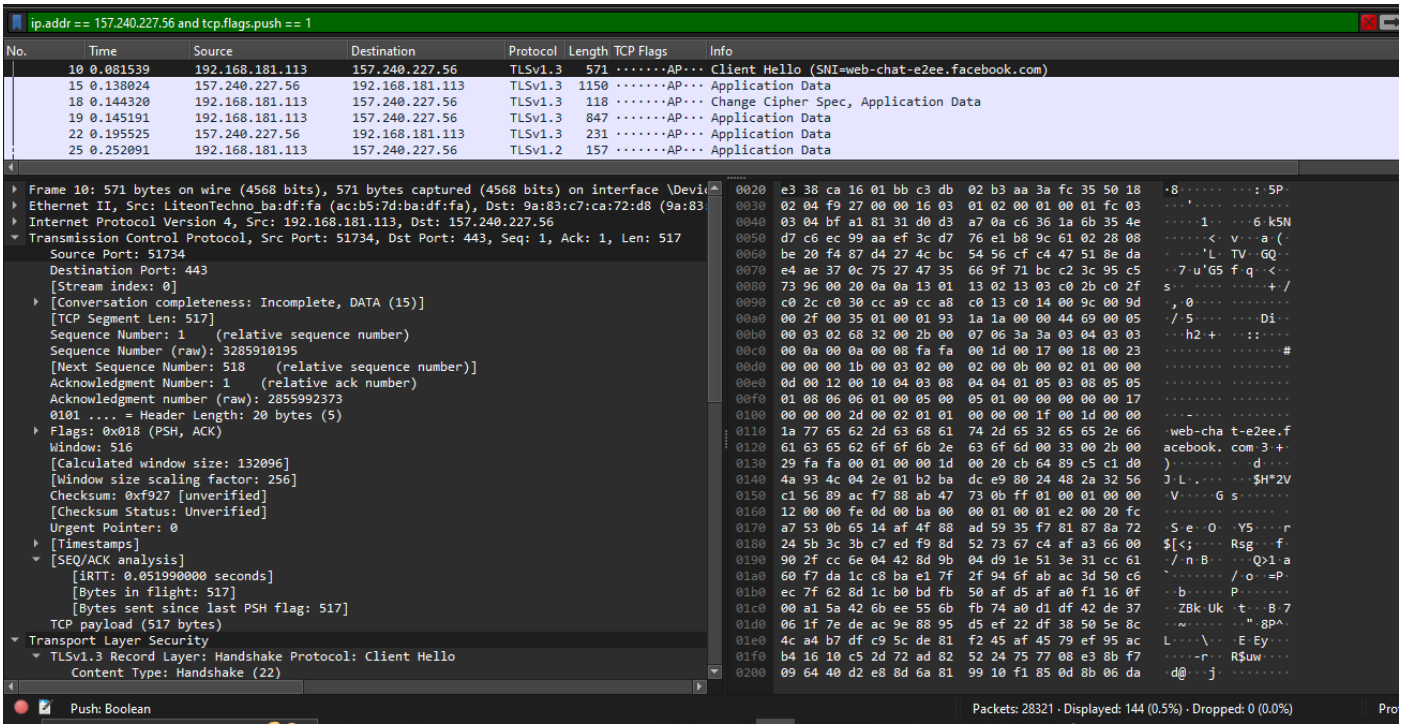
Displayed: 2

Fraction: 0.2 %



PUSH FLAG:

The filter `**ip.addr == 157.240.227.56 and tcp.flags.push == 1**` is applied to capture TCP packets with the PUSH flag set sent to or from the IP address 157.240.227.56. A total of 144 packets matching this filter are displayed, constituting 0.5% of the total traffic.



ip.addr == 157.240.227.56 and tcp.flags.push == 1

Displayed packets: 144

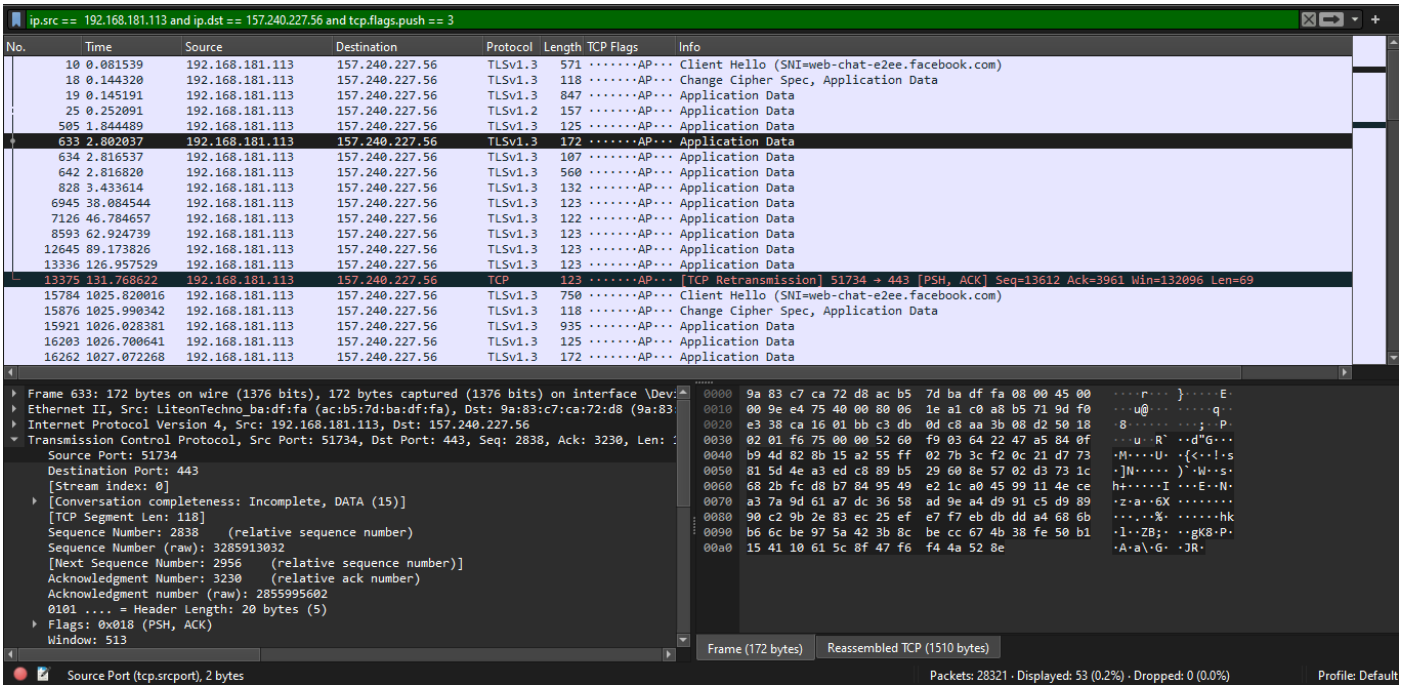
Fraction: 0.5%

From PC to Facebook:

```
ip.src ==  192.168.181.113 and ip.dst == 157.240.227.56 and tcp.flags.push == 3
ip.src ==  192.168.181.113 and ip.dst == 157.240.227.56 and tcp.flags.push == 2
```

Displayed: 53

Fraction: 0.2%



Reset:

The filter `**ip.addr == 157.240.227.56 and tcp.flags.reset == 1**` is used to capture TCP packets with the Reset (RST) flag set sent to or from the IP address 157.240.227.56. However, in the provided analysis, no packets matching this filter are displayed, indicating that there are no TCP packets with the RST flag set in the specified communication with Facebook.

Reset flag does not display any packets.

```
ip.addr == 157.240.227.56 and
tcp.flags.reset == 1
```

From Computer to the Facebook:

```
ip.src ==  192.168.181.113 and ip.dst ==
157.240.227.56 and tcp.flags.reset == 1
ip.src ==  192.168.181.113 and ip.dst ==
157.240.227.56 and tcp.flags.reset == 2
ip.src ==  192.168.181.113 and ip.dst ==
157.240.227.56 and tcp.flags.reset == 3
```

0 displayed

Facebook Statistics for Set Flags:

FLAGS	COUNT	FRACTION
SYN	4	0.0%
PSH	144	0.5%
RST	0	0%

From YouTube:

Capture Properties:

Name: E:\University\Smester 5\computer network\youtube.pcapng

Length: 219 MB

Hash (SHA256):	c3970028d87fc75da568d7c704729c83462ef91bdf96daf5975a1afd298c70a2
Hash (SHA1):	2aa2ce529f2f39f6be8ef4ee05cd17f083cdd56c
Format:	Wireshark/... - pcapng
Encapsulation:	Ethernet

Time

First packet:	2024-01-08 20:31:07
Last packet:	2024-01-08 21:02:26
Elapsed:	00:31:18

Capture

Hardware:	AMD A8-4500M APU with Radeon(tm) HD Graphics (with SSE4.2)
OS:	64-bit Windows 10 (22H2), build 19045
Application:	Dumpcap (Wireshark) 4.2.0 (v4.2.0-0-g54eedfc63953)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
Wi-Fi	0 (0.0%)	none	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	208212	208212 (100.0%)	—
Time span, s	1878.976	1878.976	—
Average pps	110.8	110.8	—
Average packet size, B	1020	1020	—
Bytes	212411107	212411107 (100.0%)	0
Average bytes/s	113 k	113 k	—
Average bits/s	904 k	904 k	—

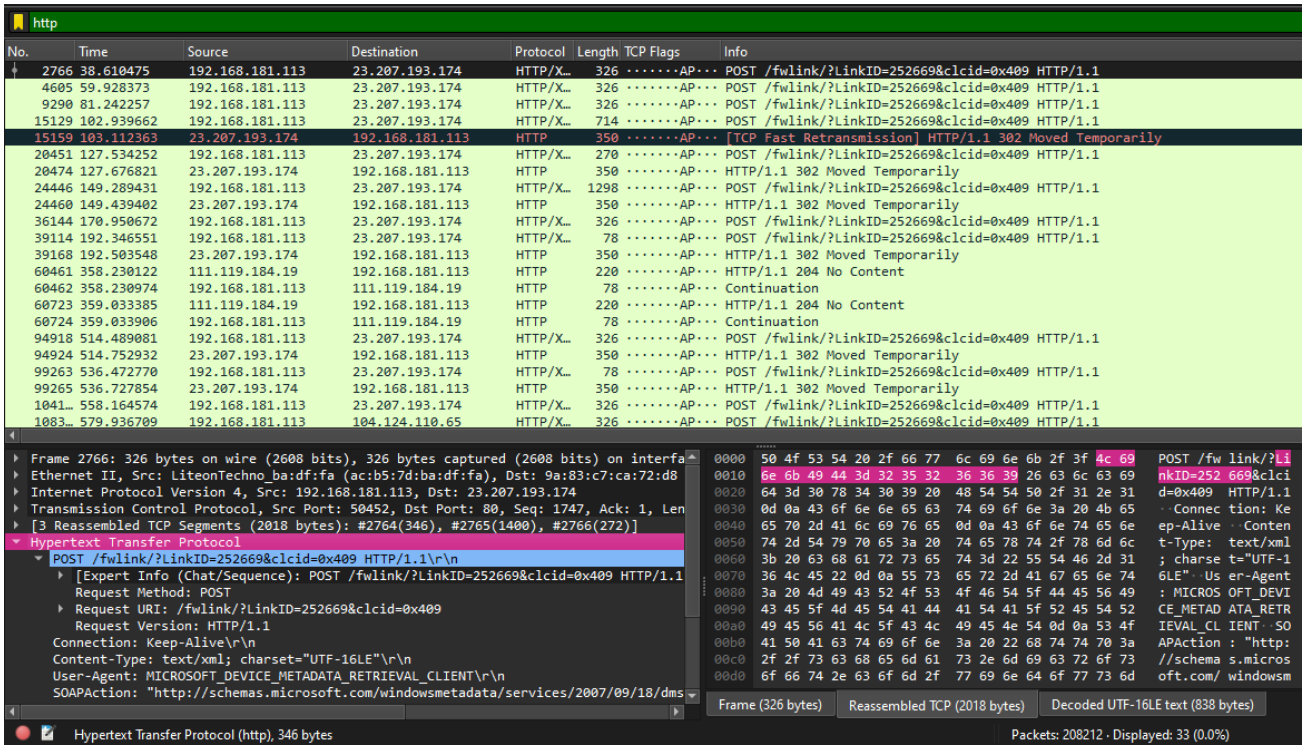
HTTP Protocol Filter for YouTube:

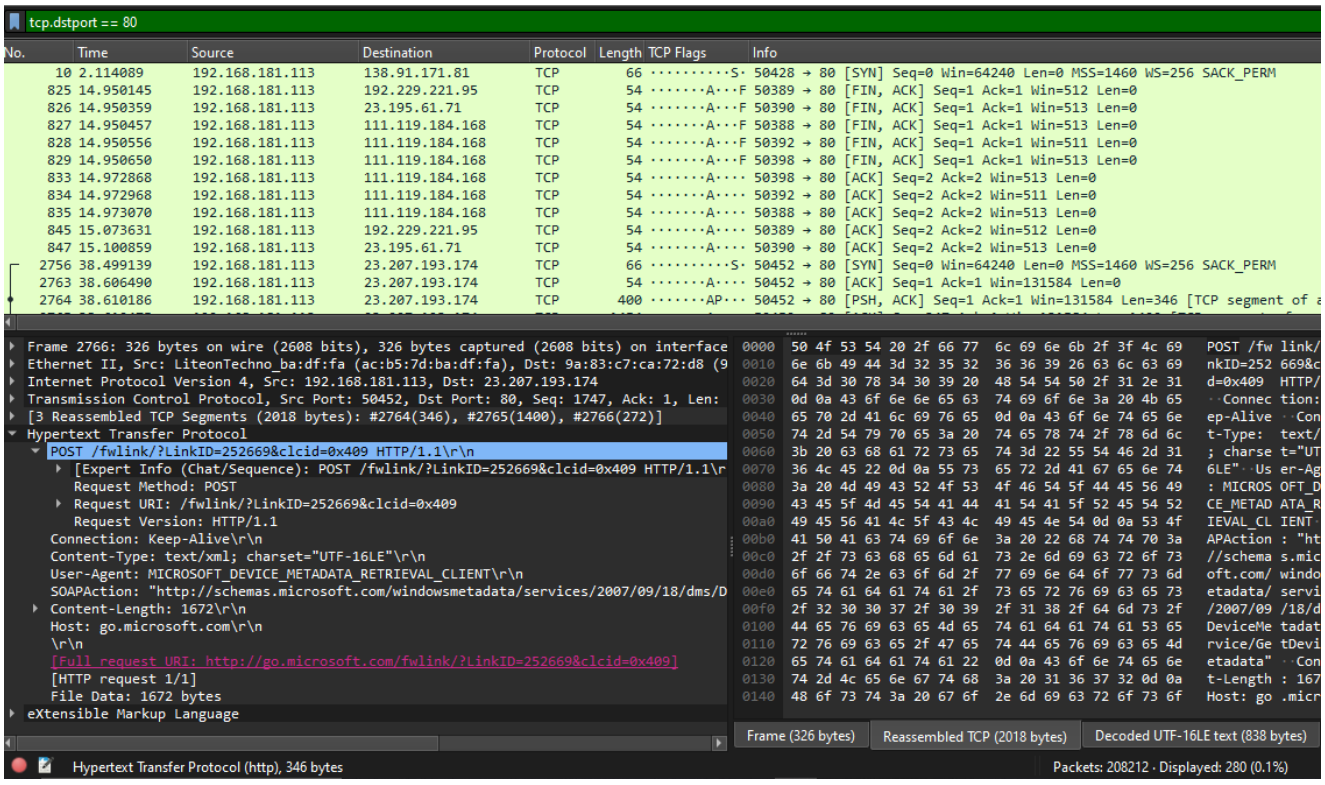
1. http

   You'll notice that all the packets in the list show HTTP for the protocol.

   To display all the HTTP traffic you need to use the following protocol and port display filter:

```
tcp.dstport == 80
```

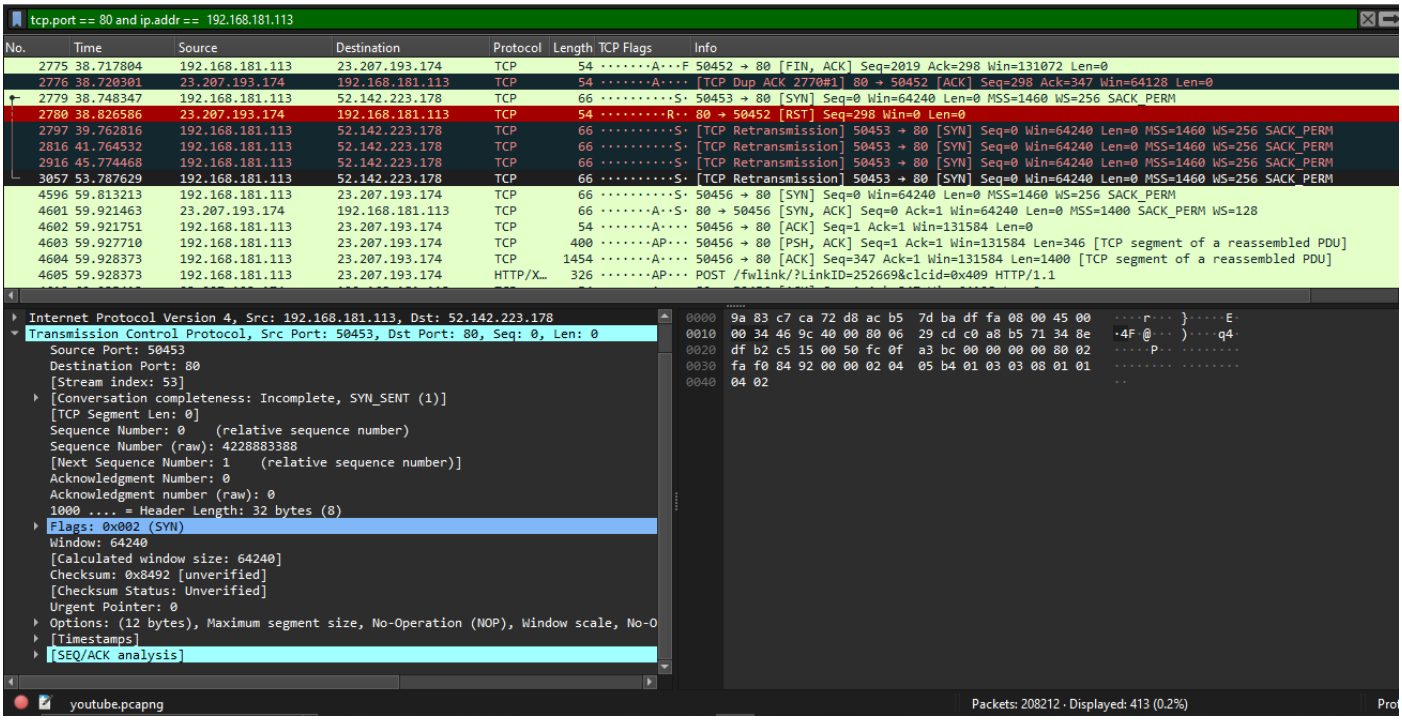




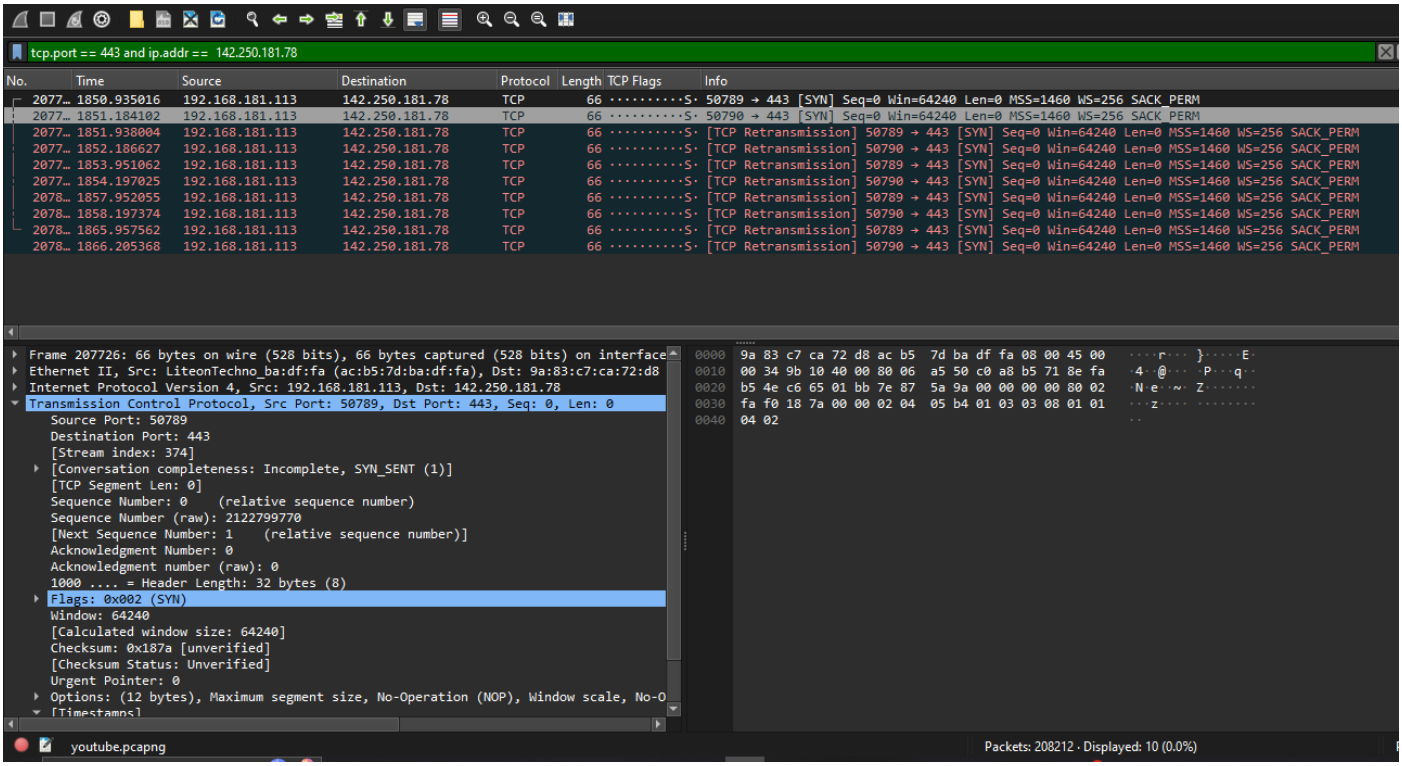
Filtering HTTP Traffic to and from Specific IP Address in Wireshark

If you want to filter for all HTTP traffic exchanged with a specific you can use the “and” operator. If, for example, you wanted to see all HTTP traffic related to a site at you could use the following filter:

```
tcp.port == 80 and ip.addr ==
192.168.181.113
```



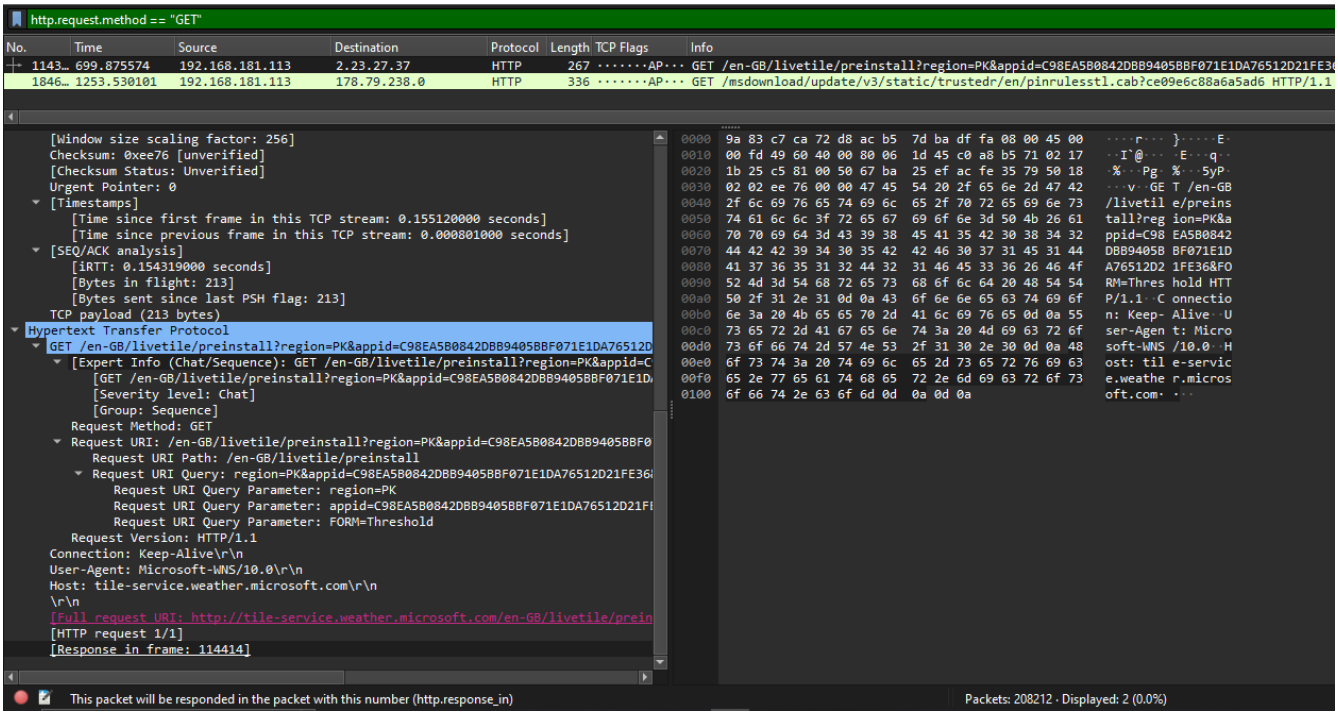
From YouTube `tcp.port == 443 and ip.addr == 142.250.181.78`



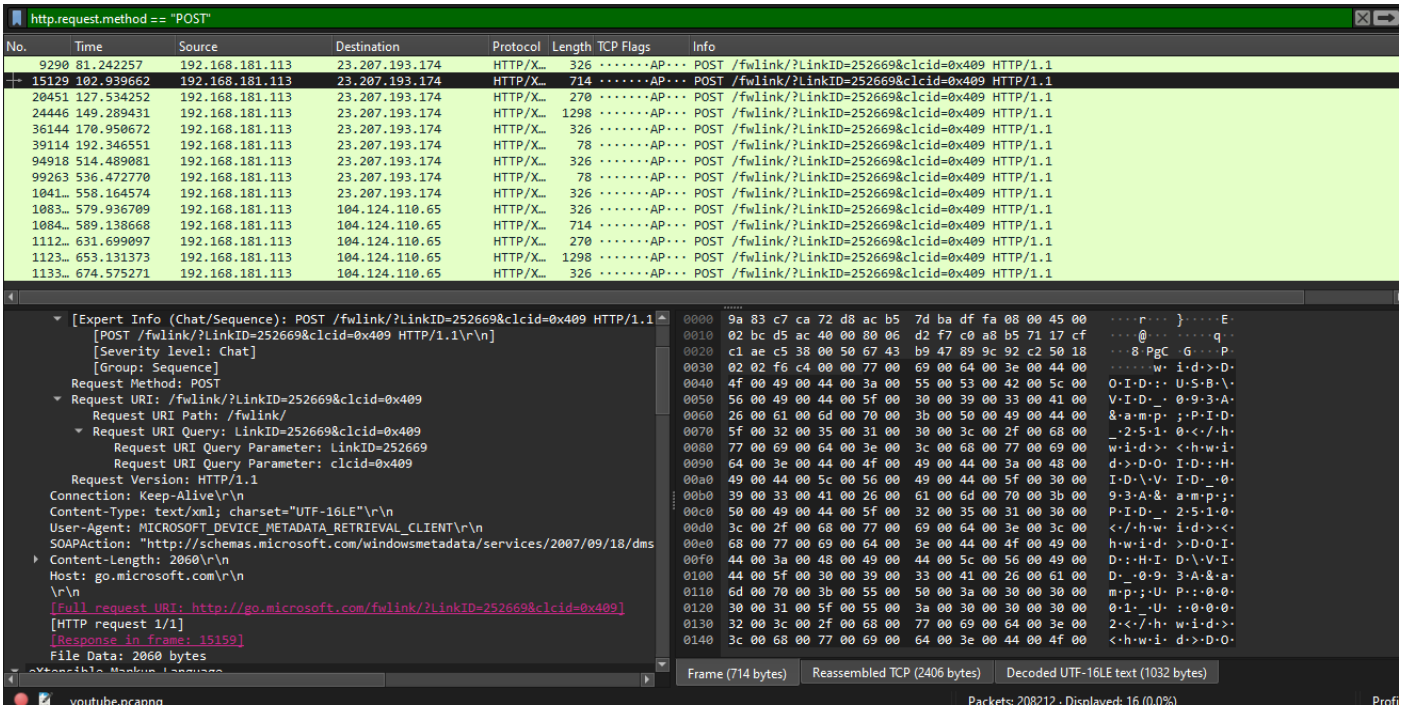
HTTP Method Filter

if you want to dig into your HTTP traffic you can filter for things like GET, PUT, POST, DELETE, HEAD, OPTIONS, CONNECT, and TRACE. To filter for these methods use the following filter syntax:

`http.request.method == "GET"`



POST:

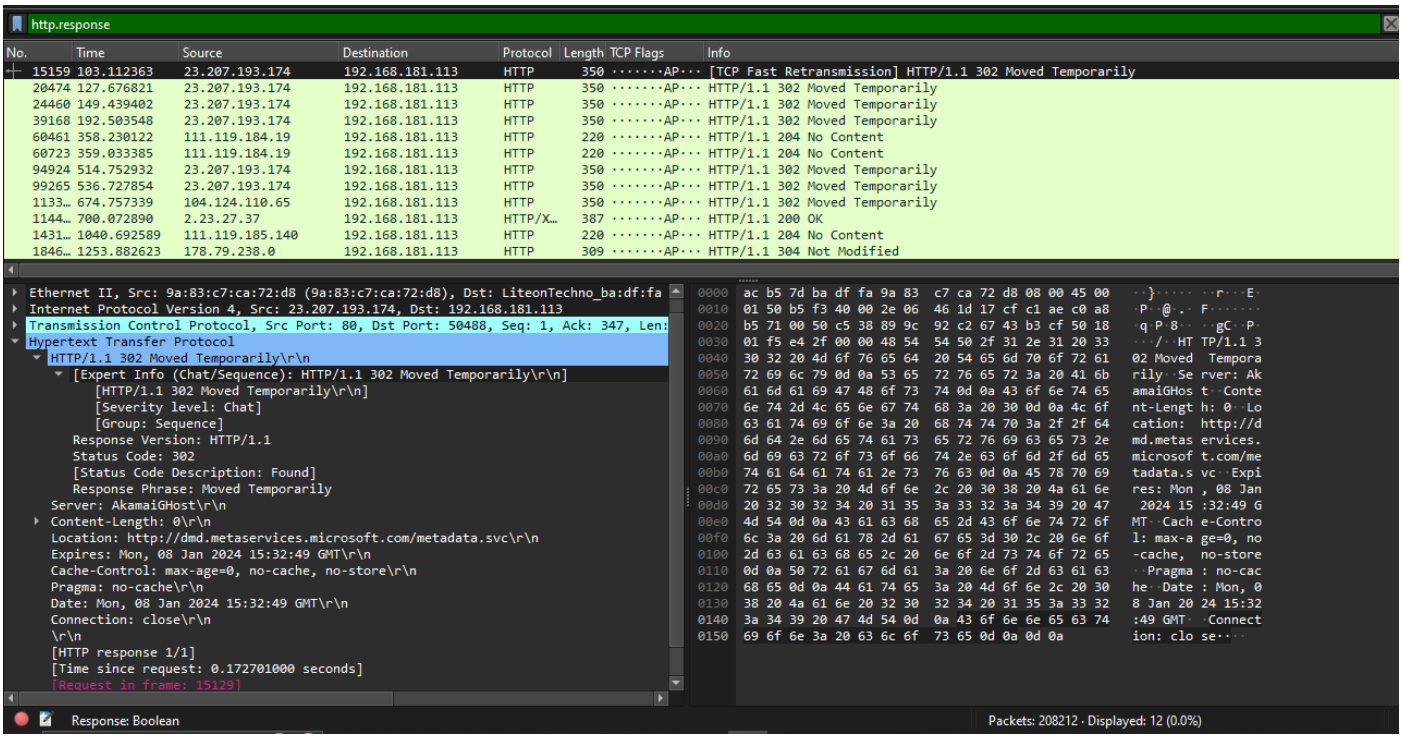


HTTP RESPONSE FILTER:

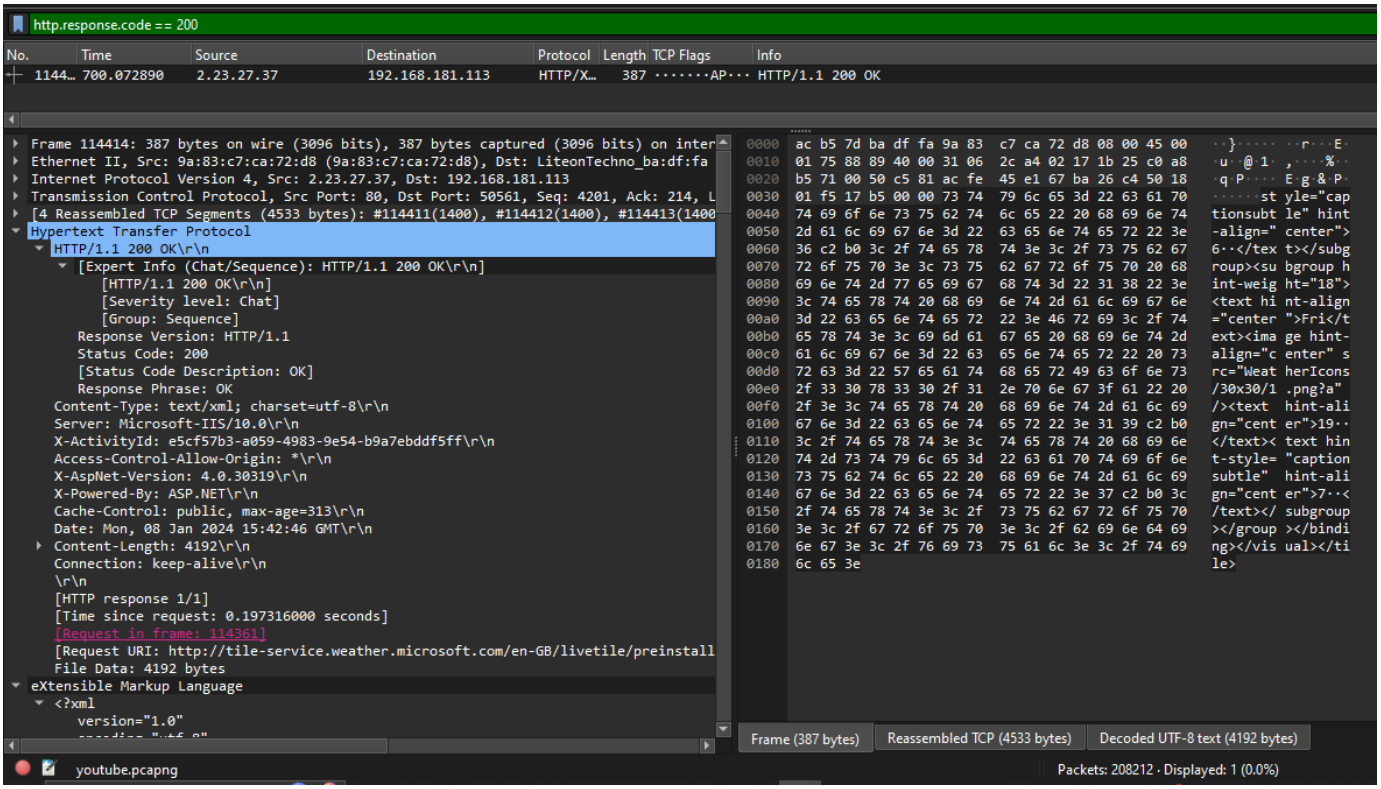
One of the many valuable bits of information in a HTTP conversation is the response. This is the code a website returns that tells the status of the asset that was requested. You've probably seen things like Error 404 (Not Found) and 403 (Forbidden). These are HTTP responses and only a couple of the many that exist.

To filter for all responses enter the following display filter:

```
http.response
```



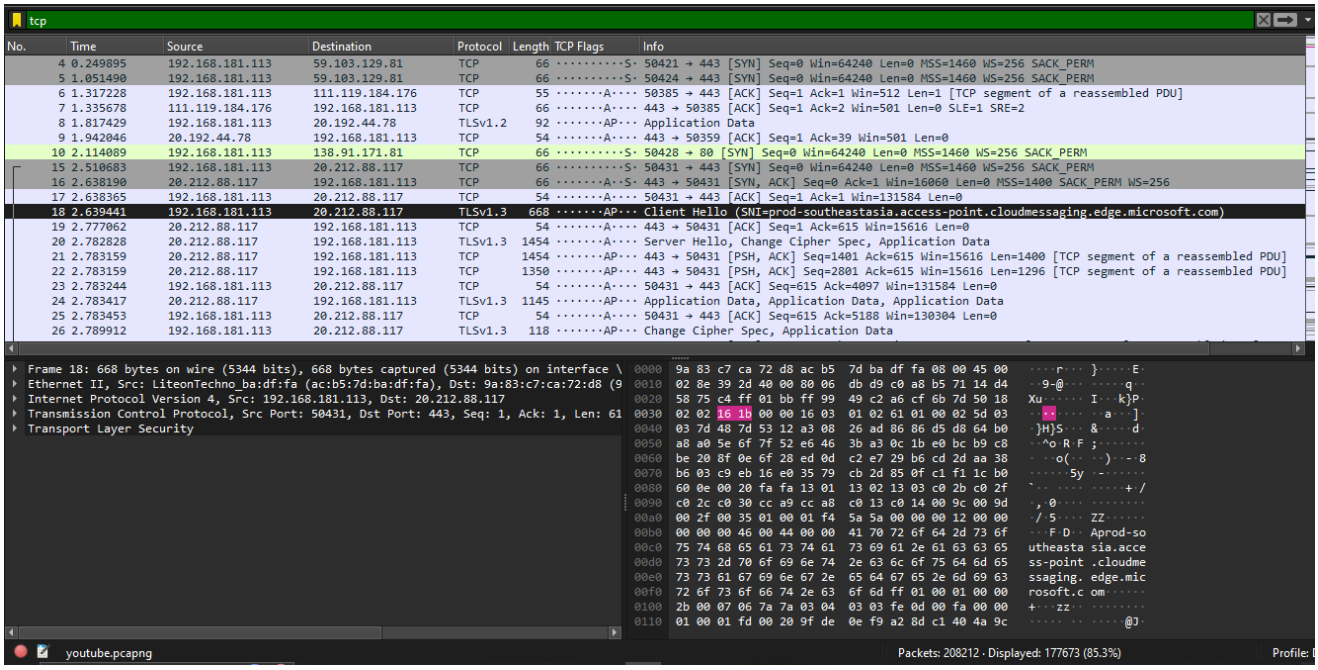
The Wireshark display filter **http.response.code == 200** is used to filter and display only those packets where the HTTP response code is 200. This filter is particularly useful when you want to focus on successful HTTP responses, as HTTP response code 200 indicates a successful request.



TCP Filter for YouTube:

1. tcp

Captures all TCP traffic related to YouTube, providing a comprehensive overview of both incoming and outgoing connections.

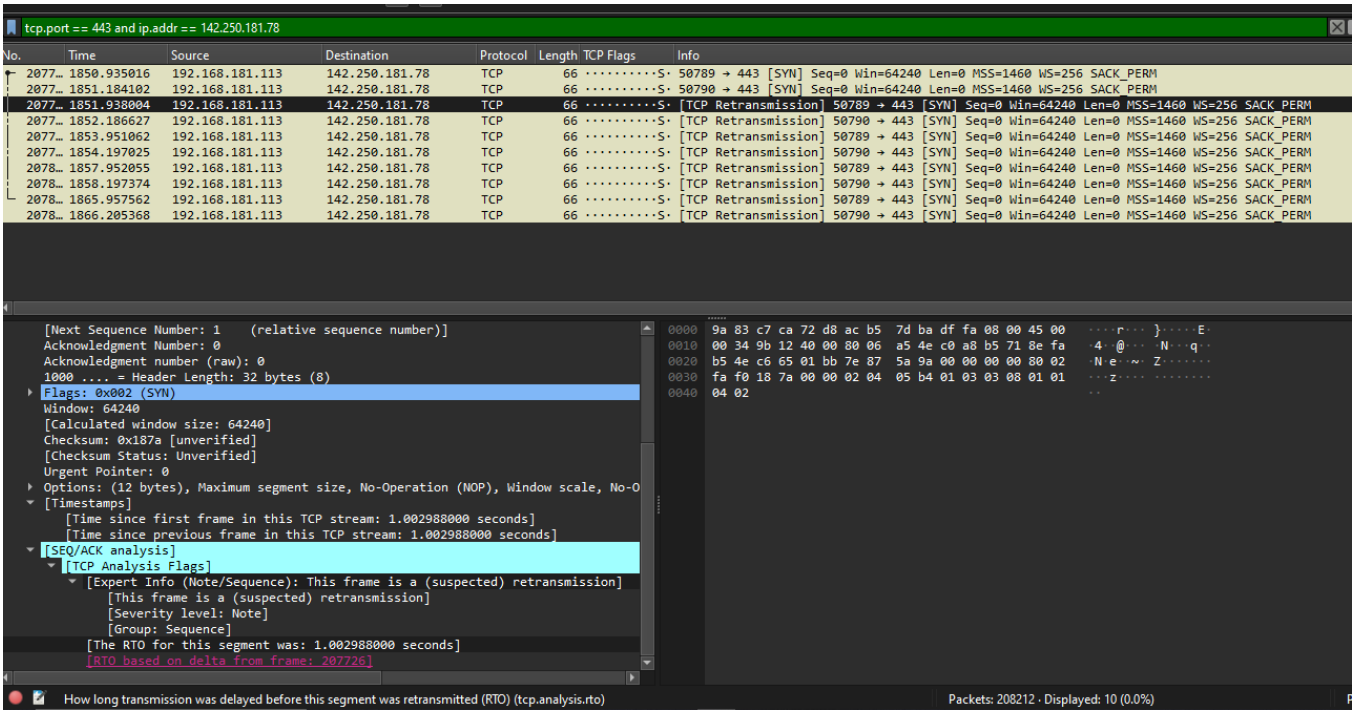


Capture TCP traffic on port 443 (HTTP) to/from YouTube:

tcp.port == 443 and ip.addr == 142.250.181.78

DISPLAYED: 10 Packets

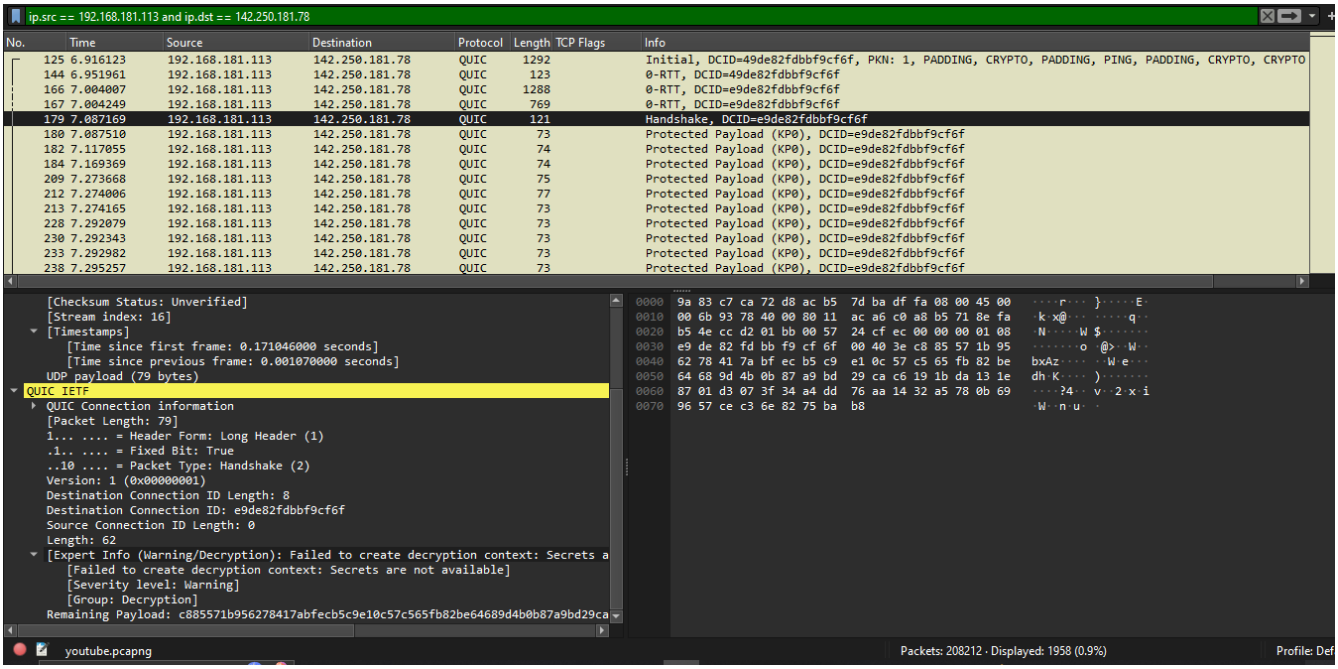
Filters TCP traffic on port 443 with a source or destination IP address of 142.250.181.78, displaying a total of 10 packets meeting these criteria.



Capture TCP traffic from computer to YouTube servers:

```
ip.src == 192.168.181.113 and ip.dst ==
142.250.181.78
Displayed: 1958 (0.9%)
```

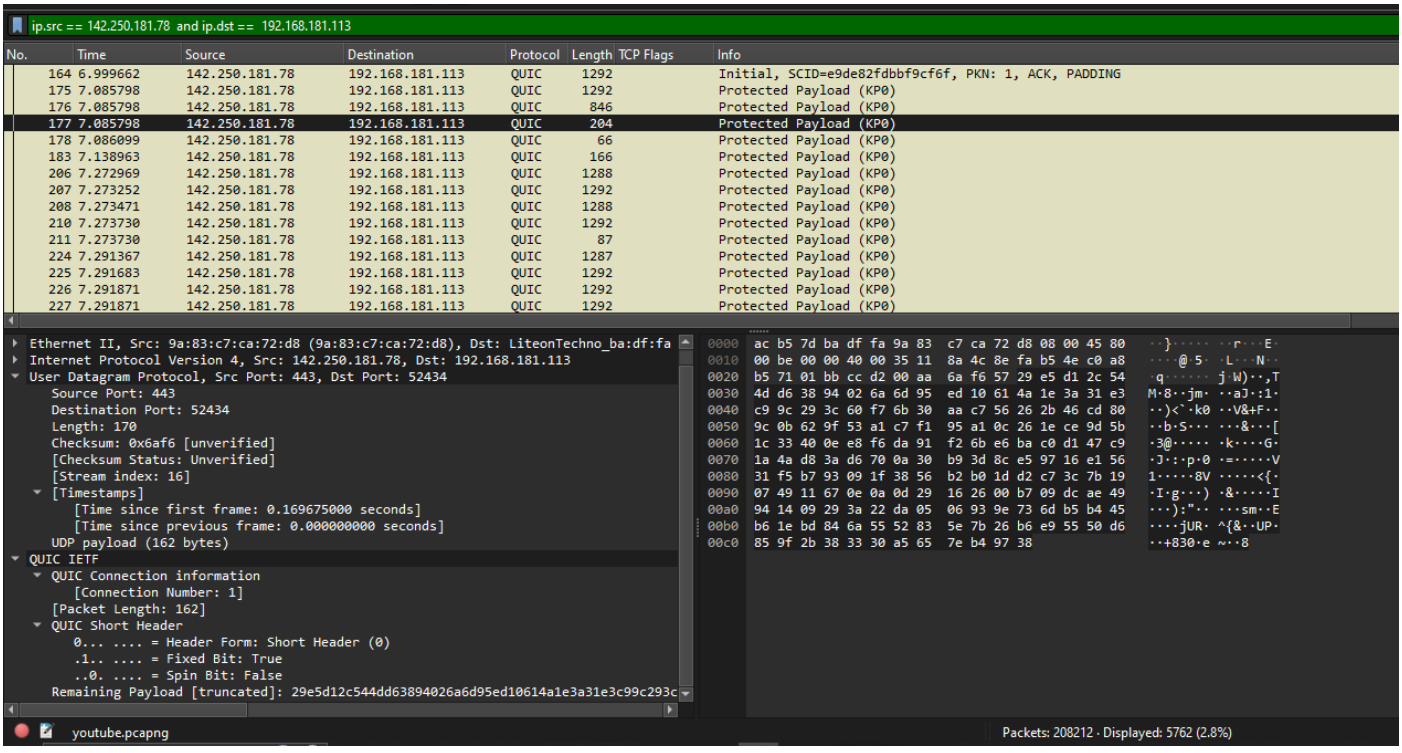
Captures TCP traffic from the local computer (source IP 192.168.181.113) to YouTube servers (destination IP 142.250.181.78), displaying 1958 packets, and representing 0.9% of the total captured traffic.



Capture TCP traffic from YouTube servers to your computer:

```
ip.src == 142.250.181.78  and ip.dst == 192.168.181.113
DISPLAYED: 5762 (2.8%)
```

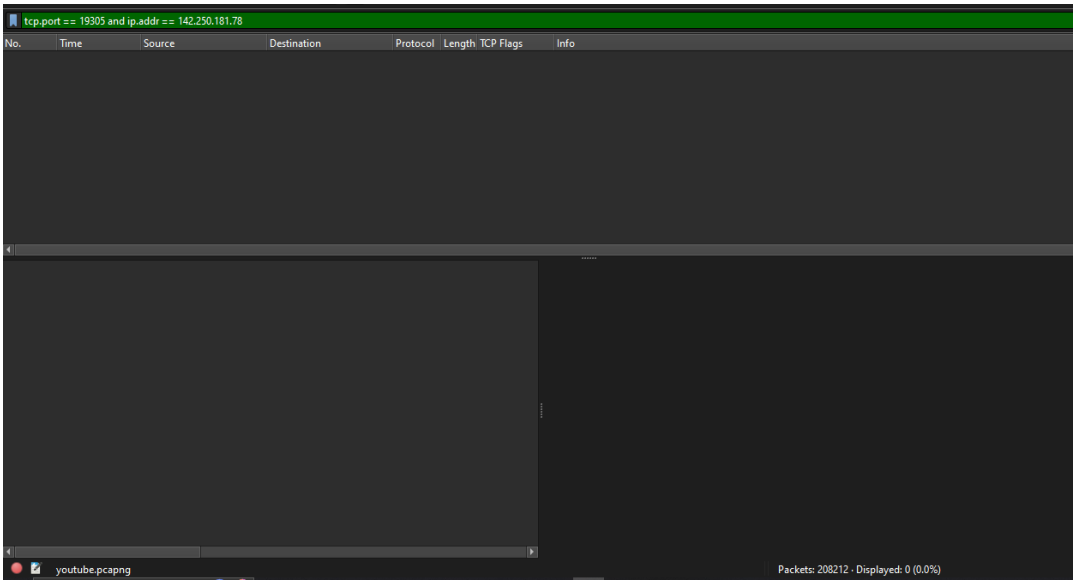
Captures TCP traffic from YouTube servers (source IP 142.250.181.78) to your computer (destination IP 192.168.181.113), displaying 5762 packets, and constituting 2.8% of the total captured traffic.



Capture TCP traffic on port 19305 (commonly used for YouTube RTMFP - Real-Time Messaging Protocol):

You are seeing zero displayed packets, there could be several reasons for this:

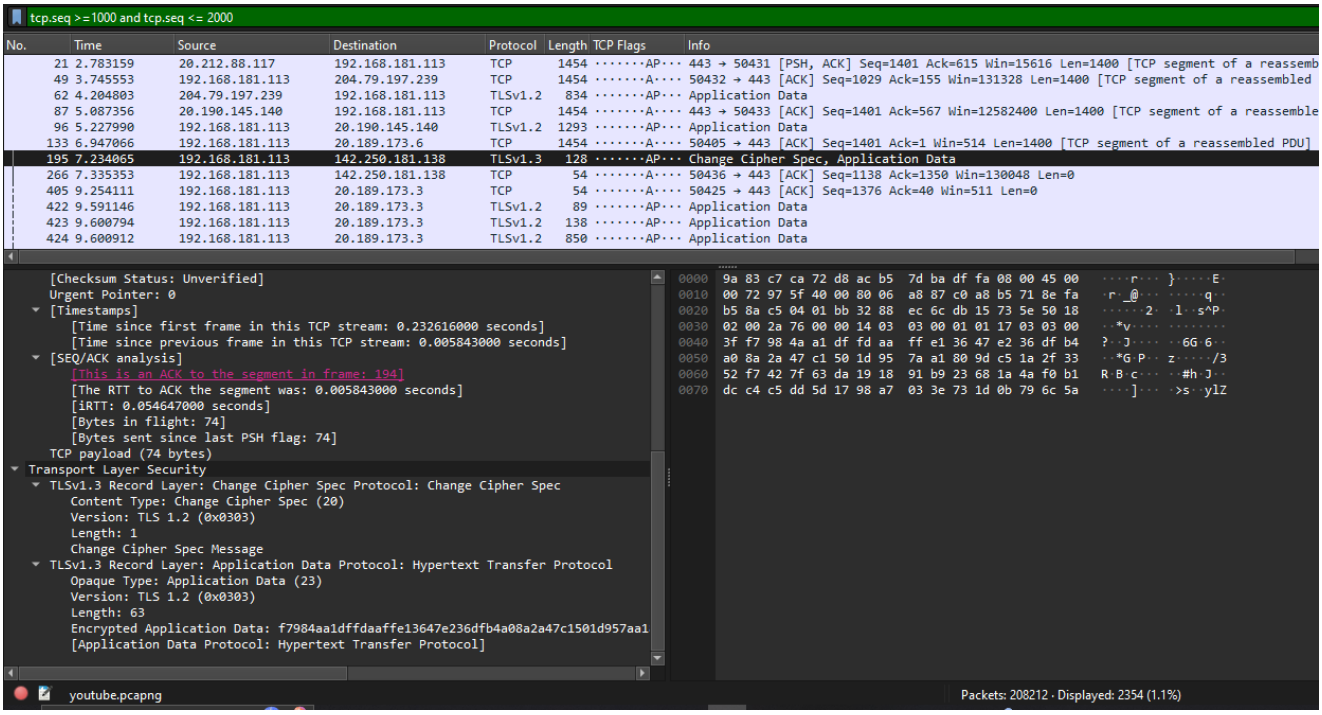
1. **No RTMFP Traffic:**
    - It's possible that there is no RTMFP traffic on the network during the time of your capture. RTMFP is specific to applications that use Adobe technologies, and not all applications or websites utilize this protocol.
2. **Encryption or Other Protocols:**
    - RTMFP traffic may be encrypted or encapsulated within another protocol, making it harder to capture or recognize using a simple port-based filter.



Capture TCP traffic with a specific sequence number range:

```
tcp.seq >=1000 and tcp.seq <= 2000
```

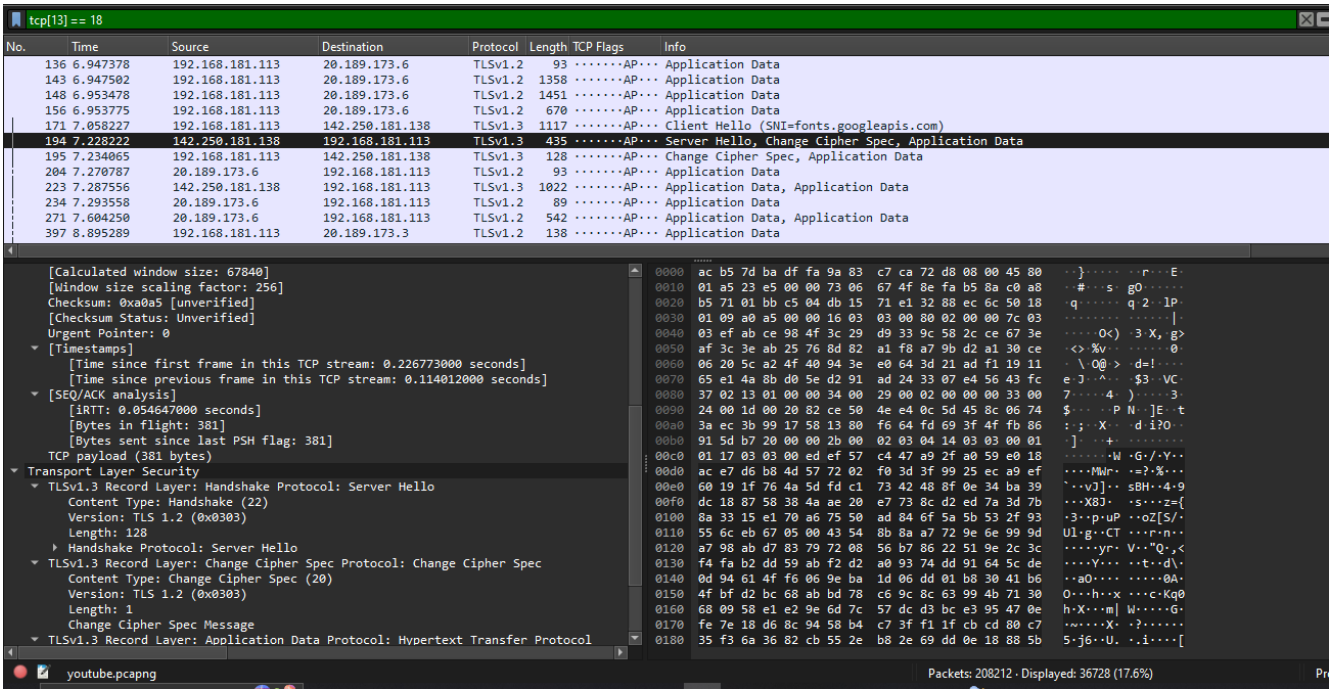
Captures TCP traffic with sequence numbers between 1000 and 2000, focusing on a specific range of sequence numbers for analysis.



Capture TCP traffic with specific TCP flags set (e.g., SYN and ACK):

```
tcp[13] == 18
DISPLAYED: 36728 (17.6%)
```

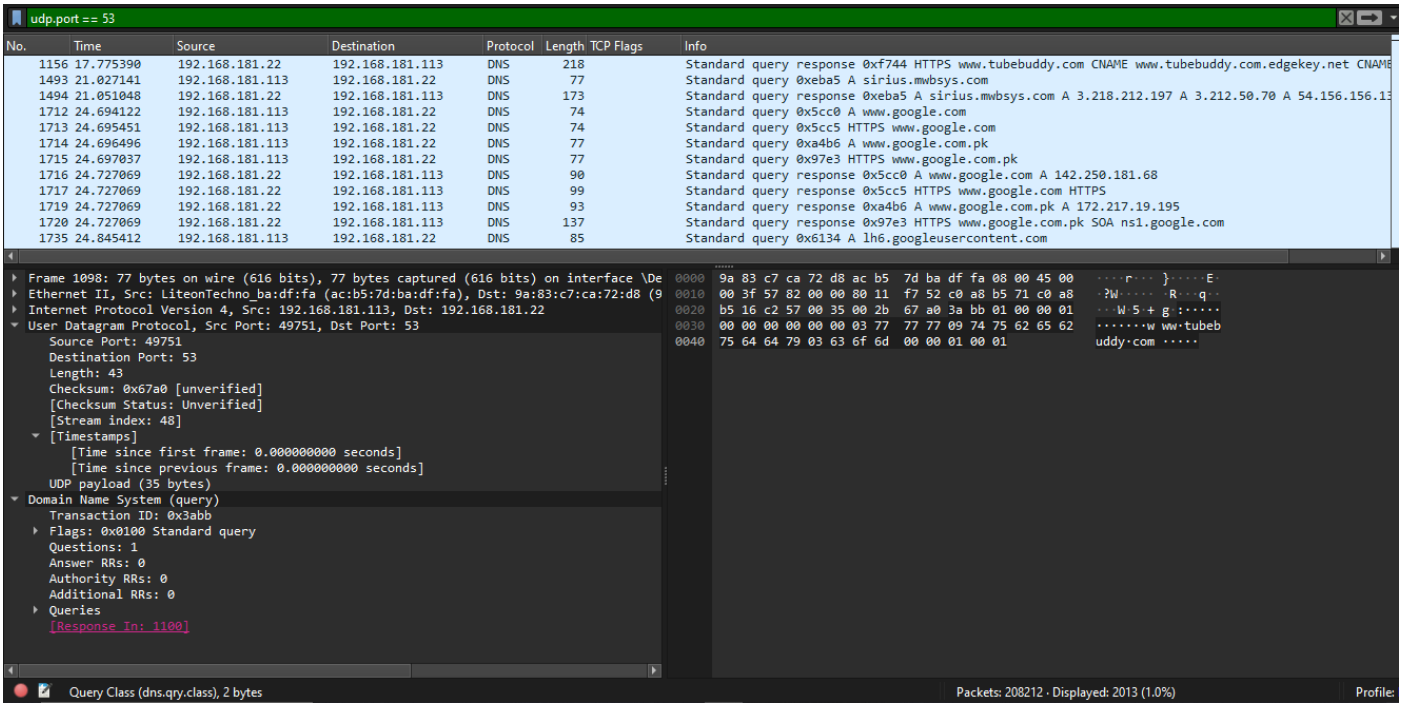
Captures TCP traffic with specific flags set (e.g., SYN and ACK) by filtering on the 13th byte of the TCP header, displaying 36,728 packets, representing 17.6% of the total captured traffic. Filters TCP traffic with specific flags set, focusing on the 13th byte of the TCP header to capture packets with intricate flag combinations (e.g., SYN and ACK). The displayed count of 36,728 packets accounts for 17.6% of the total captured traffic, providing insights into nuanced TCP communications.



Filter for DNS queries:

```
udp.port == 53
Displayed: 2013 (1%)
```

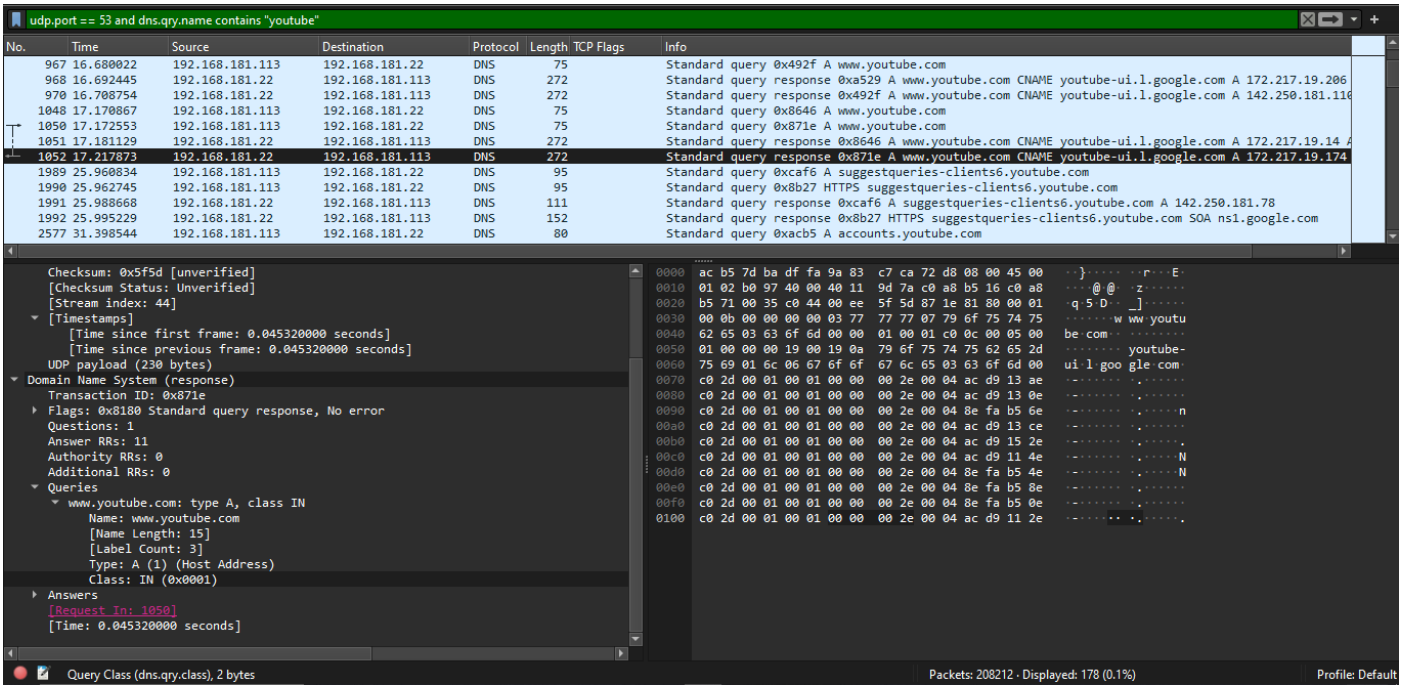
Filters DNS queries by capturing UDP traffic on port 53, displaying 2013 packets, constituting 1% of the total captured traffic.



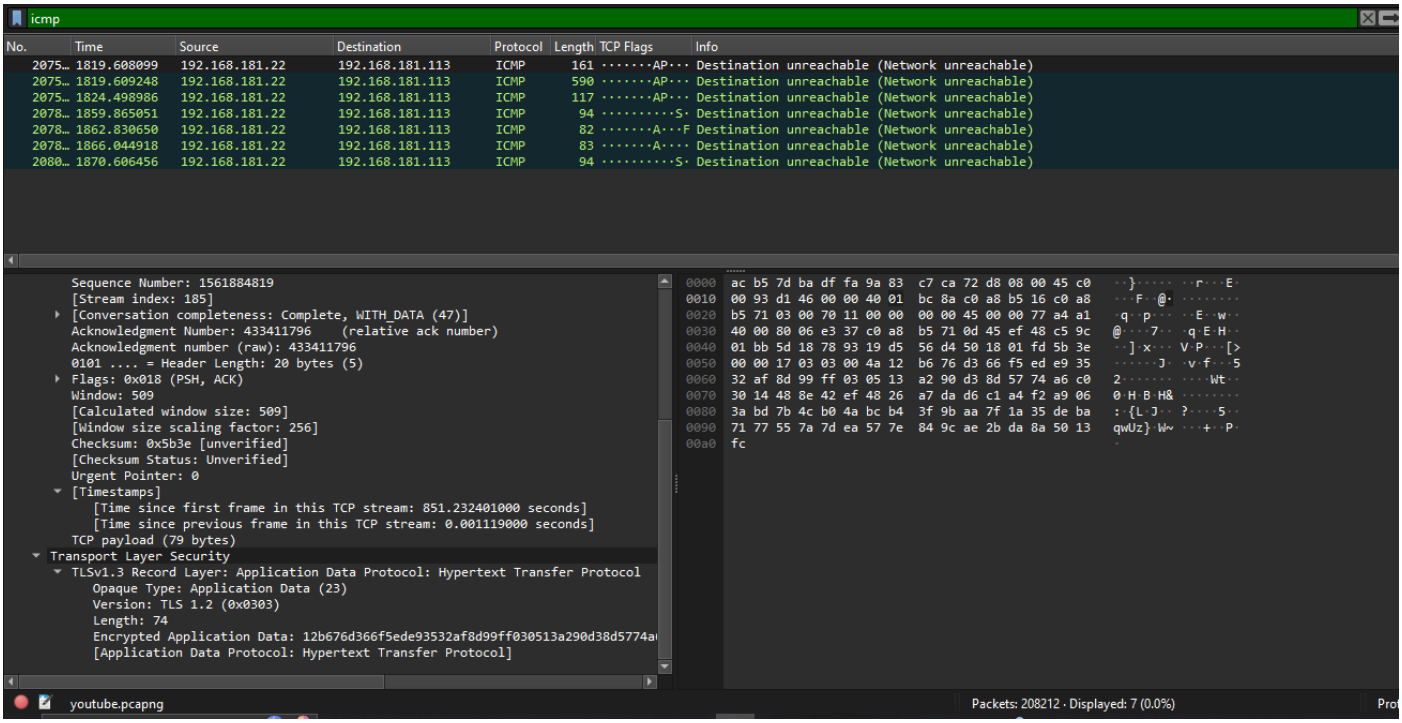

```
udp.port == 53 and dns.qry.name contains
"youtube"
```

Displayed: 178

Filters DNS queries related to YouTube by capturing UDP traffic on port 53 and checking if the DNS query name contains "YouTube", displaying 178 relevant packets.



Filter for ICMP (ping) packets:

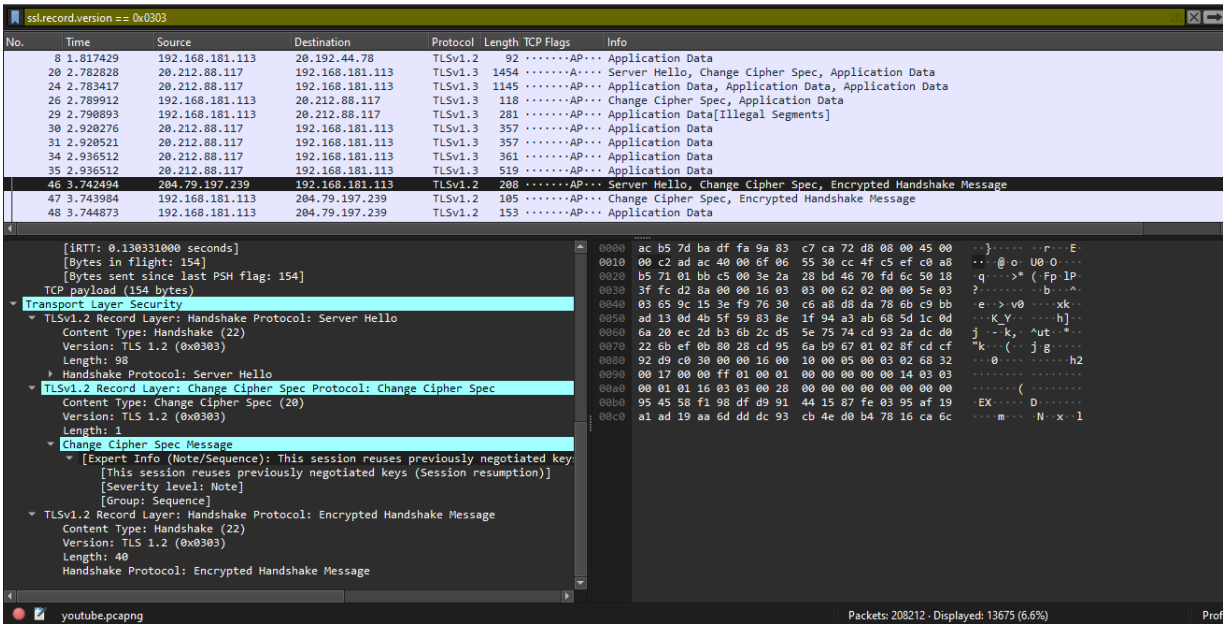


SSL:

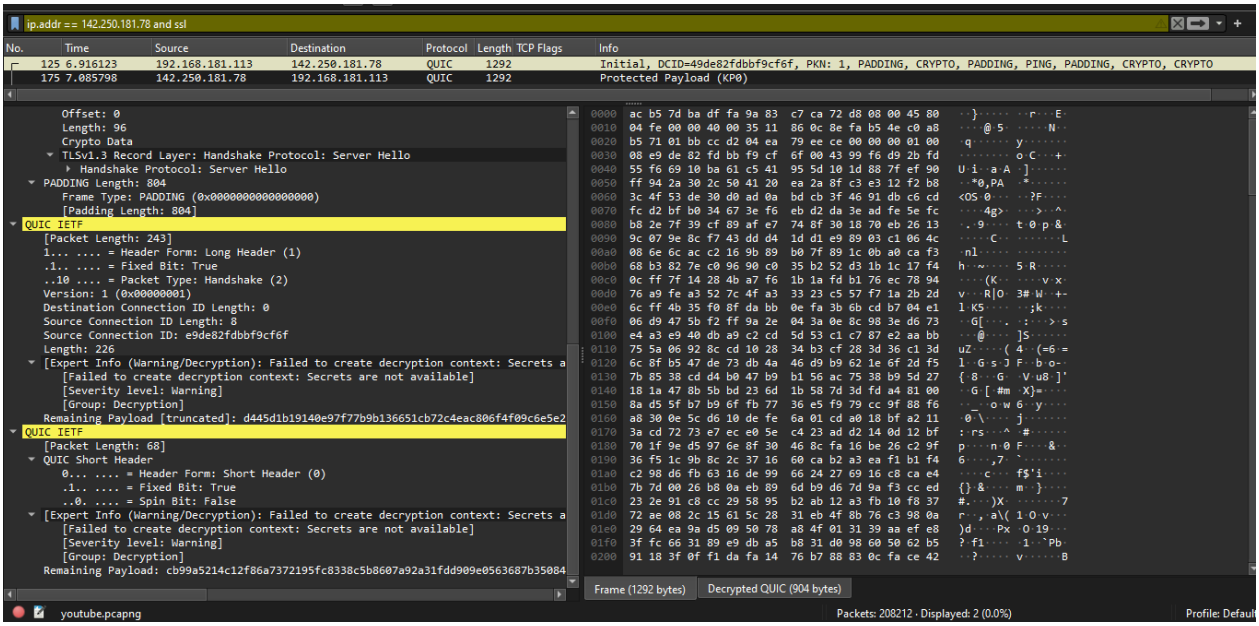
Filter for SSL/TLS traffic with specific SSL/TLS version:

```
ssl.record.version == 0x0303
```

Filters SSL/TLS traffic for packets using the specific SSL/TLS version 0x0303 (TLS 1.2) in the SSL record version field.




```
ip.addr == 142.250.181.78 and ssl
```



Filters and displays SSL/TLS traffic associated with the IP address 142.250.181.78.

FLAGS:

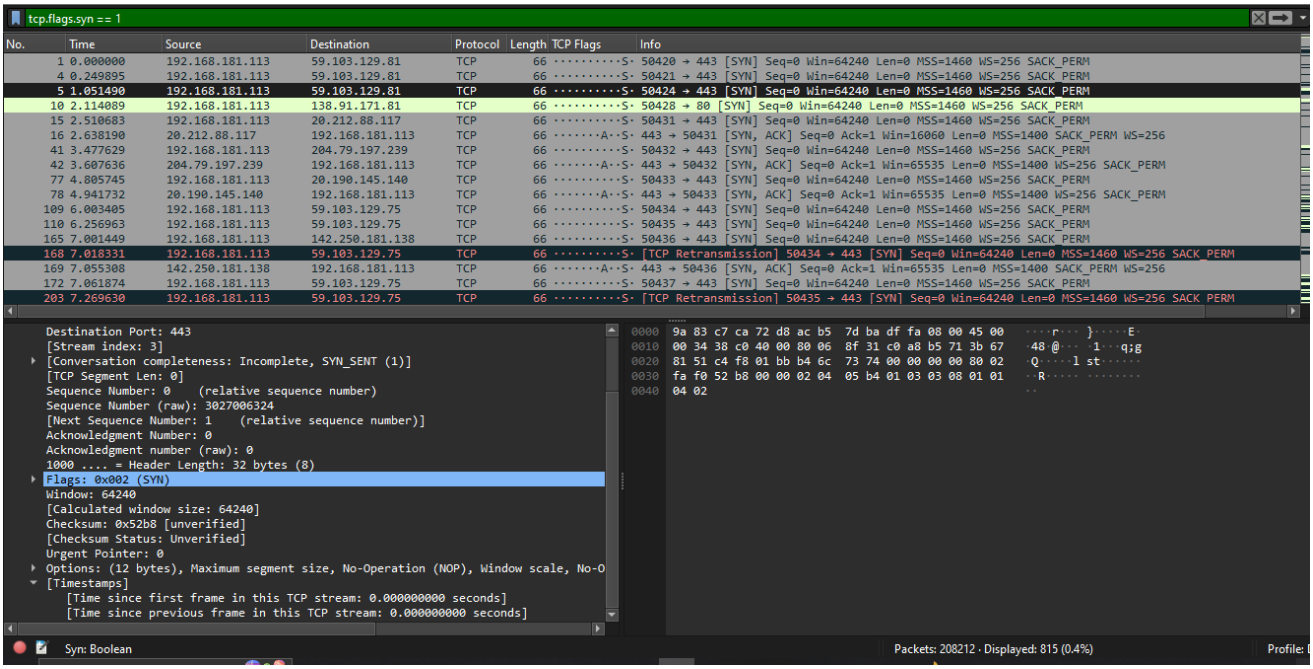
SYN FLAG:

(captured under item #1,#2,#3)

```
tcp.flags.syn == 1
```

Displayed: 815 Fraction: 0.4%

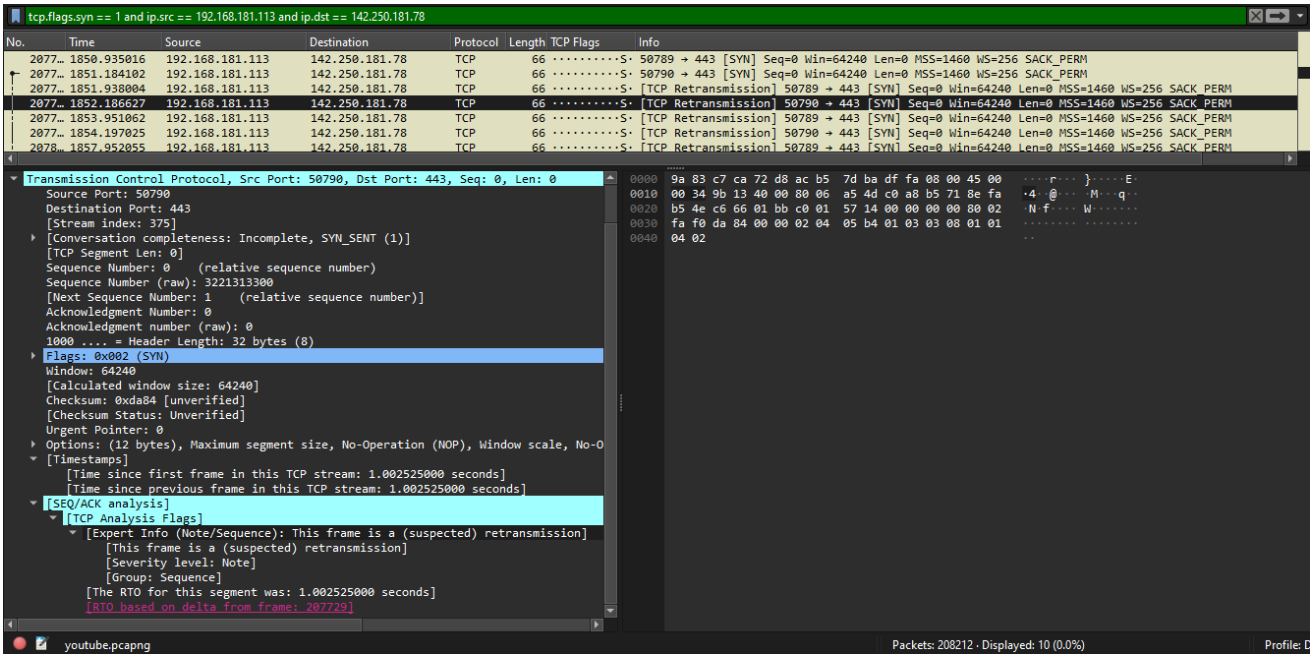
Captures and displays TCP packets with the SYN flag set, resulting in 815 packets, constituting 0.4% of the total captured traffic (items #1, #2, #3).



From PC to YouTube:

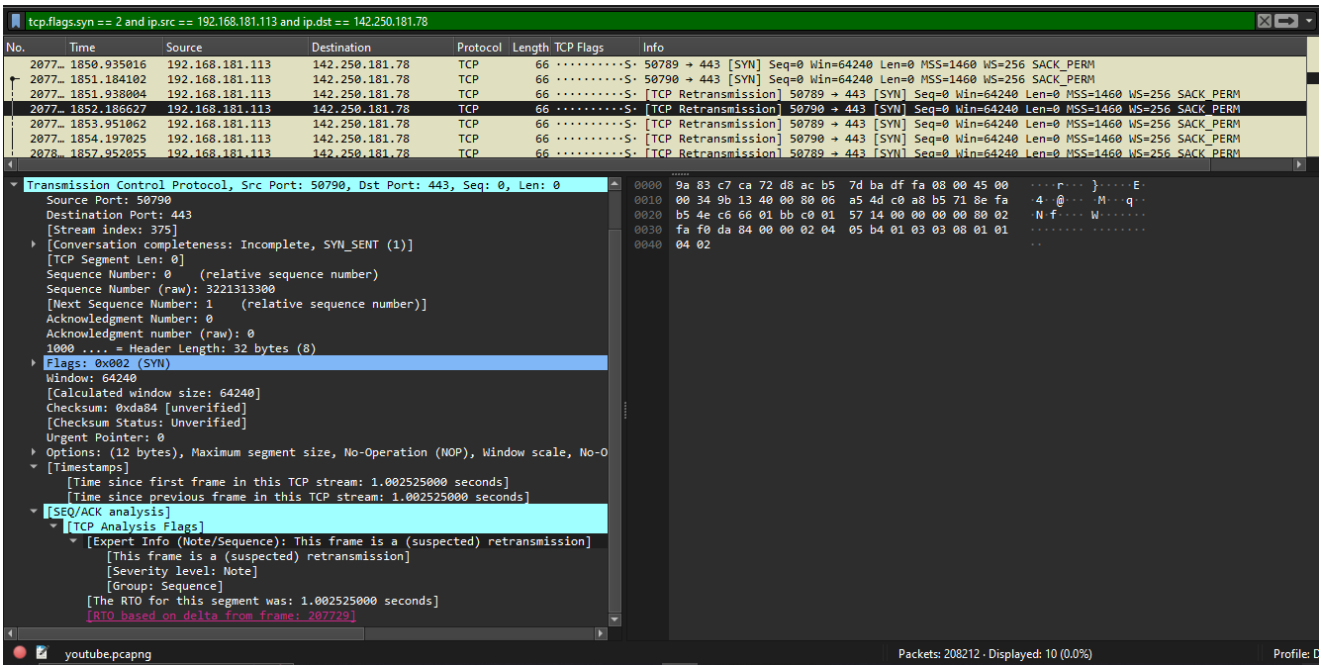
```
tcp.flags.syn == 1
and ip.src == 192.168.181.113 and ip.dst ==
142.250.181.78
Displayed: 10
Fraction: 0.0%
```

Captures TCP packets from a PC to YouTube with the SYN flag set, displaying 10 packets, representing 0.0% of the total captured traffic.



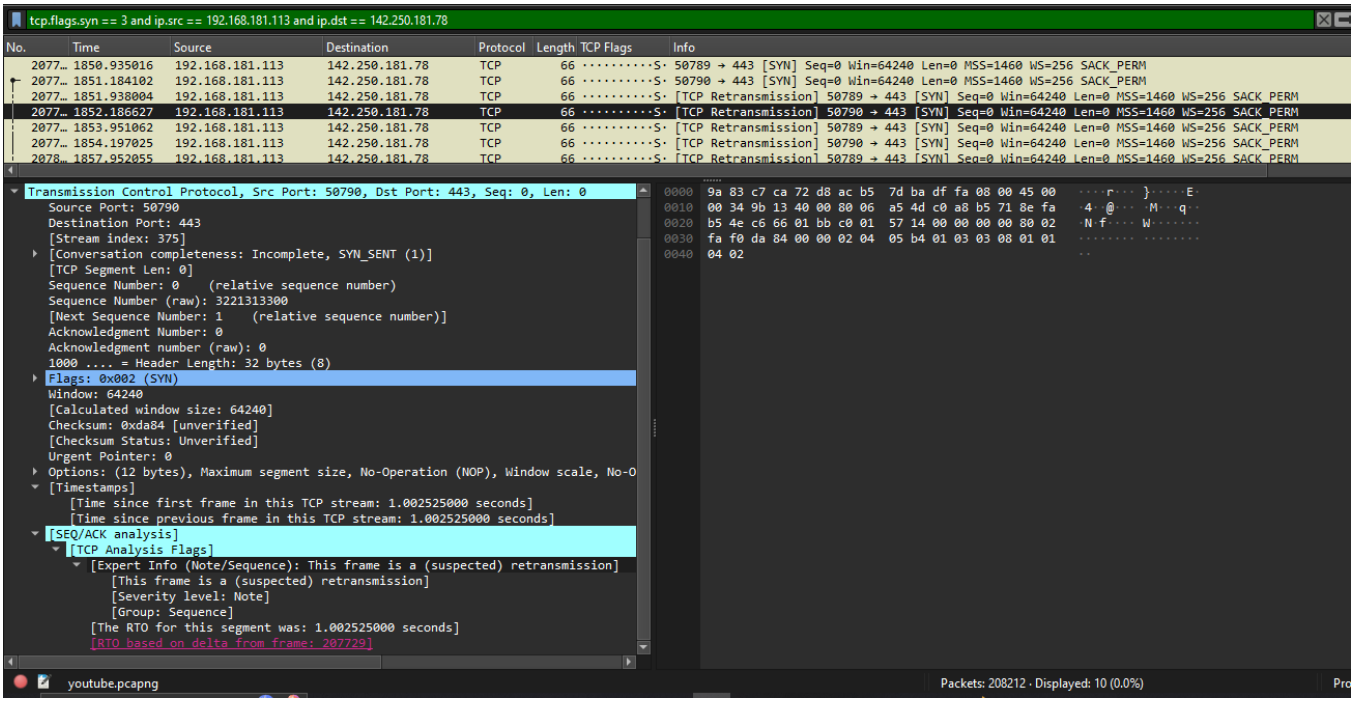

```
tcp.flags.syn == 2
and ip.src == 192.168.181.113 and ip.dst ==
142.250.181.78
```

Filters TCP packets from a PC to YouTube with the SYN flag set to 2, focusing on specific SYN values in the TCP handshake.




```
tcp.flags.syn == 3
and ip.src == 192.168.181.113 and ip.dst ==
142.250.181.78
```

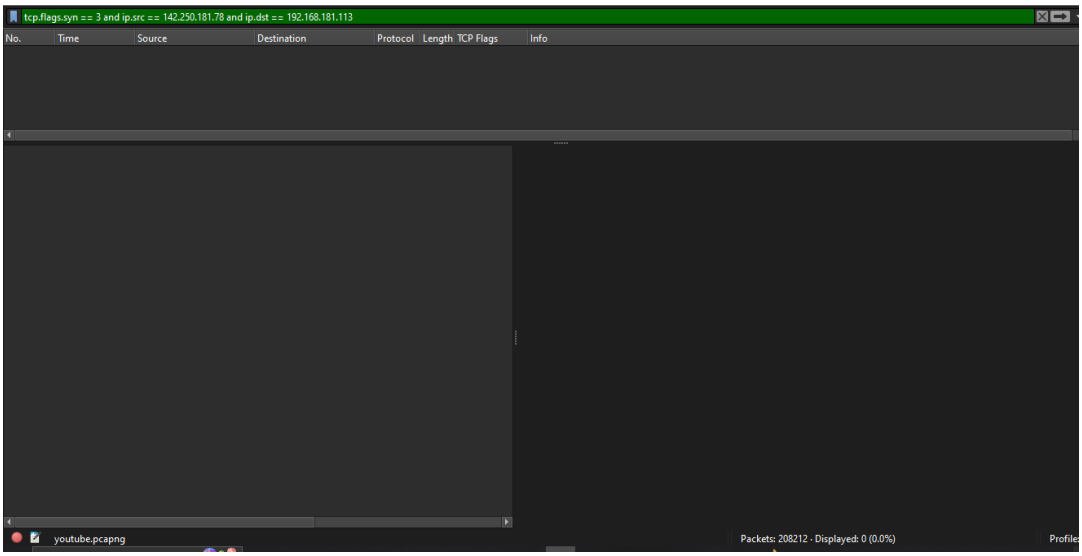
Filters TCP packets from a PC to YouTube with the SYN flag set to 3, targeting a specific condition in the TCP handshake.



FROM YouTube to PC:

0 for #3

The provided information "0 for #3" suggests that there were no TCP packets from YouTube to the PC with the specified conditions (SYN flag set to 3, source IP 192.168.181.113, and destination IP 142.250.181.78) in the captured data under item #3.



0 displayed for #2

0 displayed for #1

```
tcp.flags.syn == 1
and ip.src == 142.250.181.78 and ip.dst ==
192.168.181.113
```

Indicates that there were no displayed packets for item #2 and item #1 with the provided filters. The specific filter `tcp.flags.syn == 1 and ip.src == 142.250.181.78 and ip.dst == 192.168.181.113` focuses on capturing TCP packets with the SYN flag set from YouTube to the PC.

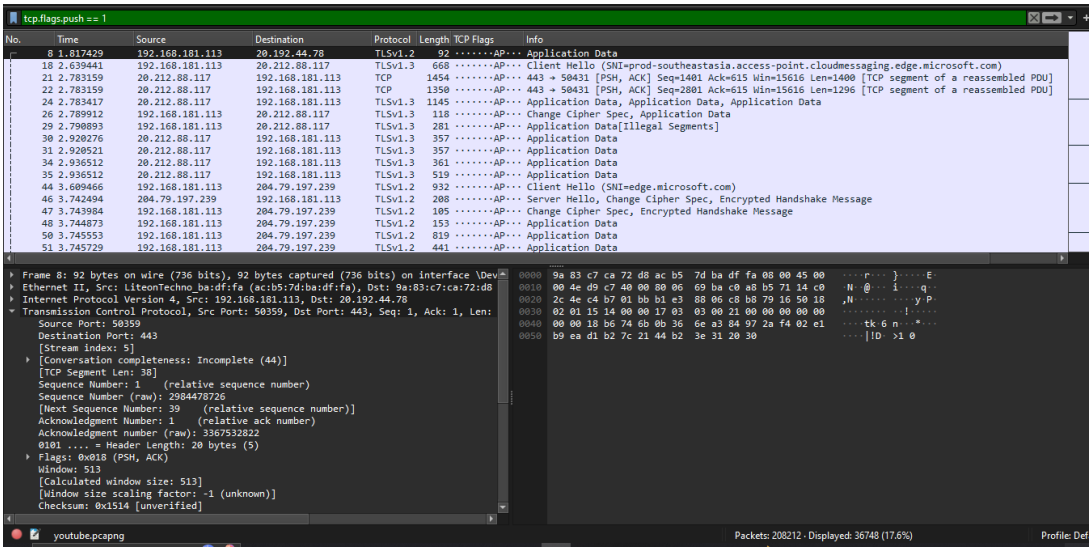
PUSH:

(captured under item #1,#2,#3)

From PC to YouTube:

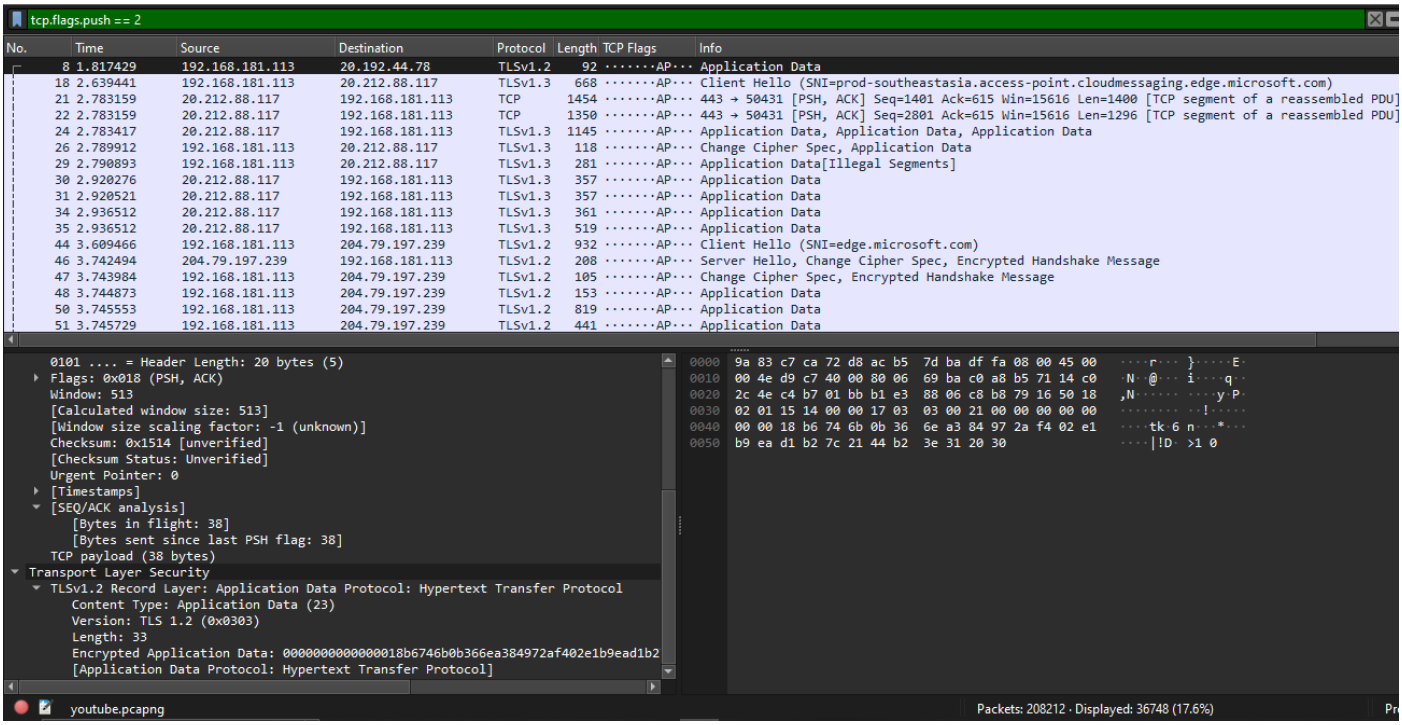
```
tcp.flags.push == 1
displayed: 36748 Fraction: 17.6%
```

Captures and displays TCP packets with the PUSH flag set from the PC to YouTube, resulting in 36,748 packets, representing 17.6% of the total captured traffic under items #1, #2, and #3.



`tcp.flags.push == 2`

The filter `tcp.flags.push == 2` focuses on capturing TCP packets where the PUSH flag is set to 2, targeting a specific condition related to the PUSH flag in the TCP protocol.



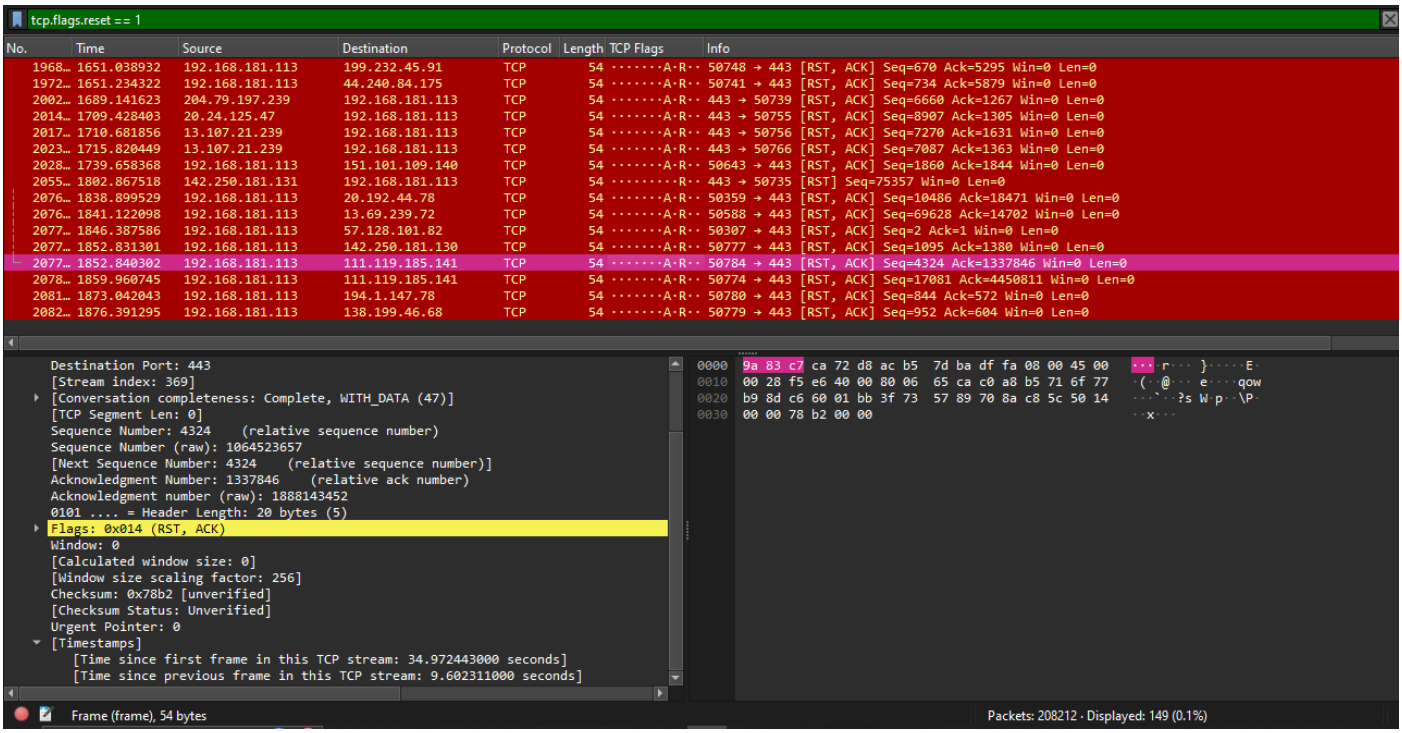
RESET:

`tcp.flags.reset == 1`

Dsiplayed: 149 Fraction:0.1%

Same display and fraction for the Capture Item #2 and #3.

Captures and displays TCP packets with the RESET (RST) flag set to 1. The displayed count is 149 packets, constituting 0.1% of the total captured traffic, and the same applies for Capture Items #2 and #3. Captures TCP packets with the RESET (RST) flag set to 1, displaying 149 packets, constituting 0.1% of the total traffic, consistent across Capture Items #2 and #3. This filter helps identify reset events in the TCP communication.



YouTube Table for set flags for capture item #1

FLAGS	COUNT	FRACTION
SYN	10	0.0%
PSH	36748	17.6%
RST	149	0.1%

Reference Videos:

https://youtu.be/u4ht-E-Kihk?si=4SBxFQ9y8TPN8S9e

https://youtu.be/5qecyZHL-GU?si=89GhFkp3VLl0bHWZ

VPN PROJECT

Introduction:

A Virtual Private Network (VPN) is a service designed to create a secure, encrypted connection to ensure privacy and anonymity while utilizing the internet. VPNs extend a private network across a public network, enabling users to securely send and receive data.

II. Why use a VPN?

1. **Privacy and Anonymity:** VPNs hide a user's browser history, IP address, and geographical location, enhancing security and preventing unauthorized access to personal data.
2. **Bypassing Geographical Restrictions***:* VPNs enable access to restricted content based on location, allowing users to visit otherwise unavailable websites.
3. **Remote Access***:* VPNs provide secure remote access to a company's network, facilitating work from home or while traveling.
4. **Public WiFi Security:** VPNs protect users on public WiFi networks from hackers and cybercriminals attempting to intercept sensitive information.

III. How VPNs Work:

1. **Tunneling Protocols***:* VPNs utilize tunneling protocols like OpenVPN, SSTP, and IKEv2 for data encryption and decryption.

2. **Encryption:** VPNs use encryption to secure data transmitted over the internet, preventing unauthorized reading without the decryption key.
3. **IP Address Masking**: VPNs replace a user's IP address with that of the VPN server, maintaining anonymity and preventing tracking.

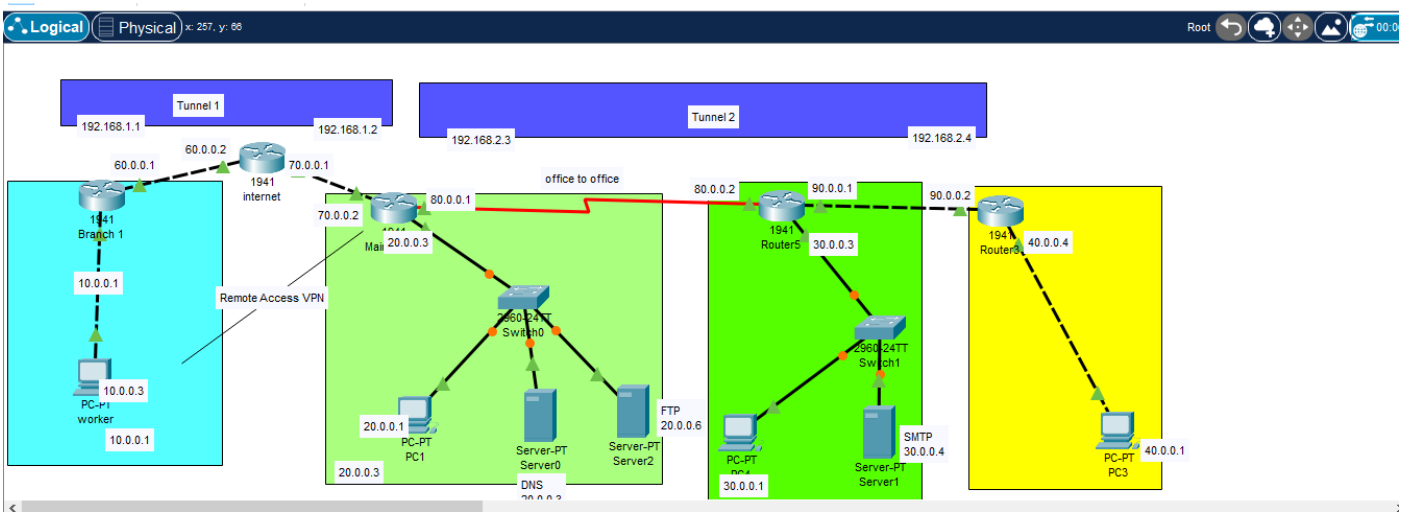
IV. Types of VPNs:

1. **Remote Access VPNs:** Allow individual users to securely connect to a private network over the internet.
2. **Site-to-Site VPNs**: Connect entire networks in different locations, ensuring secure communication between them.
3. **Hardware VPNs:** Physical devices providing VPN functionality, commonly used for added security in enterprise environments.

Software Used:

Cisco Packet Tracer

Visual Representation:



Process:

In this scenario, specific network devices were chosen, including Cisco 1941 routers, a 2960-24 switch with a battery backup, and a server. The reasons behind these choices are:

Cisco 1941 Routers:

- *High-Performance Routing:* Known for high-performance capabilities suitable for routing tasks.
- *Scalability:* Offers scalability to accommodate growing network needs.
- *Security Features:* Advanced security features for firewall capabilities, access control, and VPN support.
- *Modular Design:* Flexibility in adding modules for specific functionalities.
- *Reliability:* Ensures uninterrupted connectivity and minimizes downtime.

Cisco 2960-24 Switch:

- *Network Switching:* Used for local network switching, providing high-performance and low-latency communication.
- *Port Density:* With 24 ports, accommodates numerous devices within the local network.
- *Battery Backup:* Ensures continuous power supply, crucial for maintaining network operations during power outages.

Server:

The server serves various purposes, contributing to the overall functionality, efficiency, and management of the network:

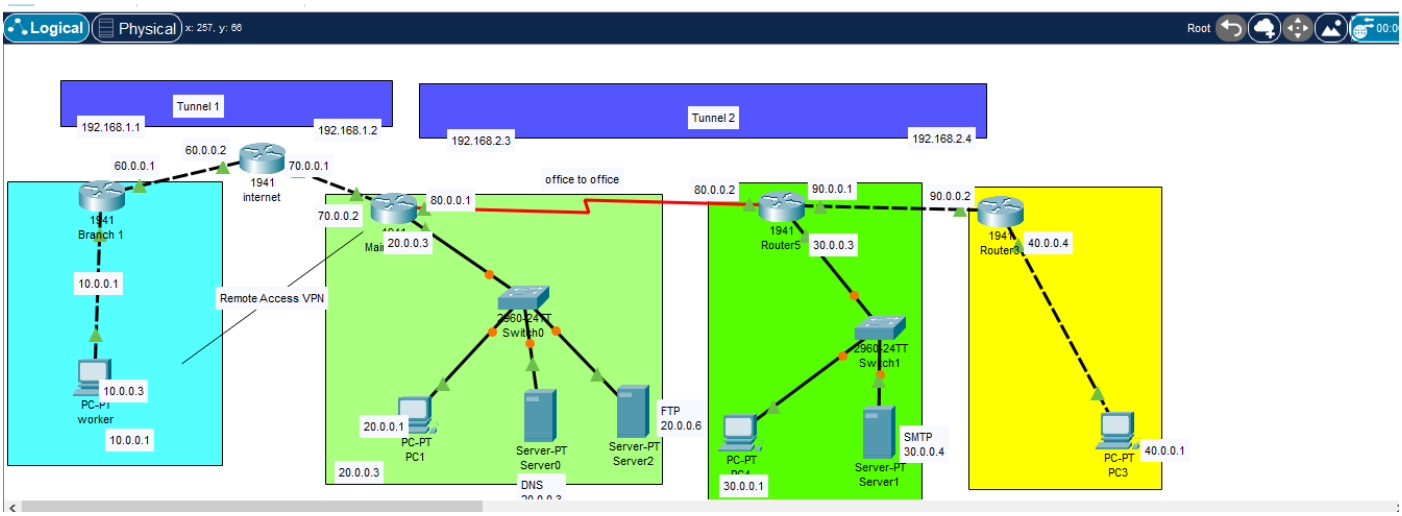
- *Centralized Resource Hosting*

- *User Authentication and Authorization*
- *Network Services*
- *Data Storage and Backup*
- *Application Hosting*
- *Security Management*
- *Power Resilience and Controlled Shutdown*
- *Centralized Updates and Patch Management*
- *Network Administration and Monitoring*

WORKING

Network Connectivity:

The Cisco 1941 routers are responsible for routing traffic within your network. They facilitate communication between different subnets and connect your local network to the internet.

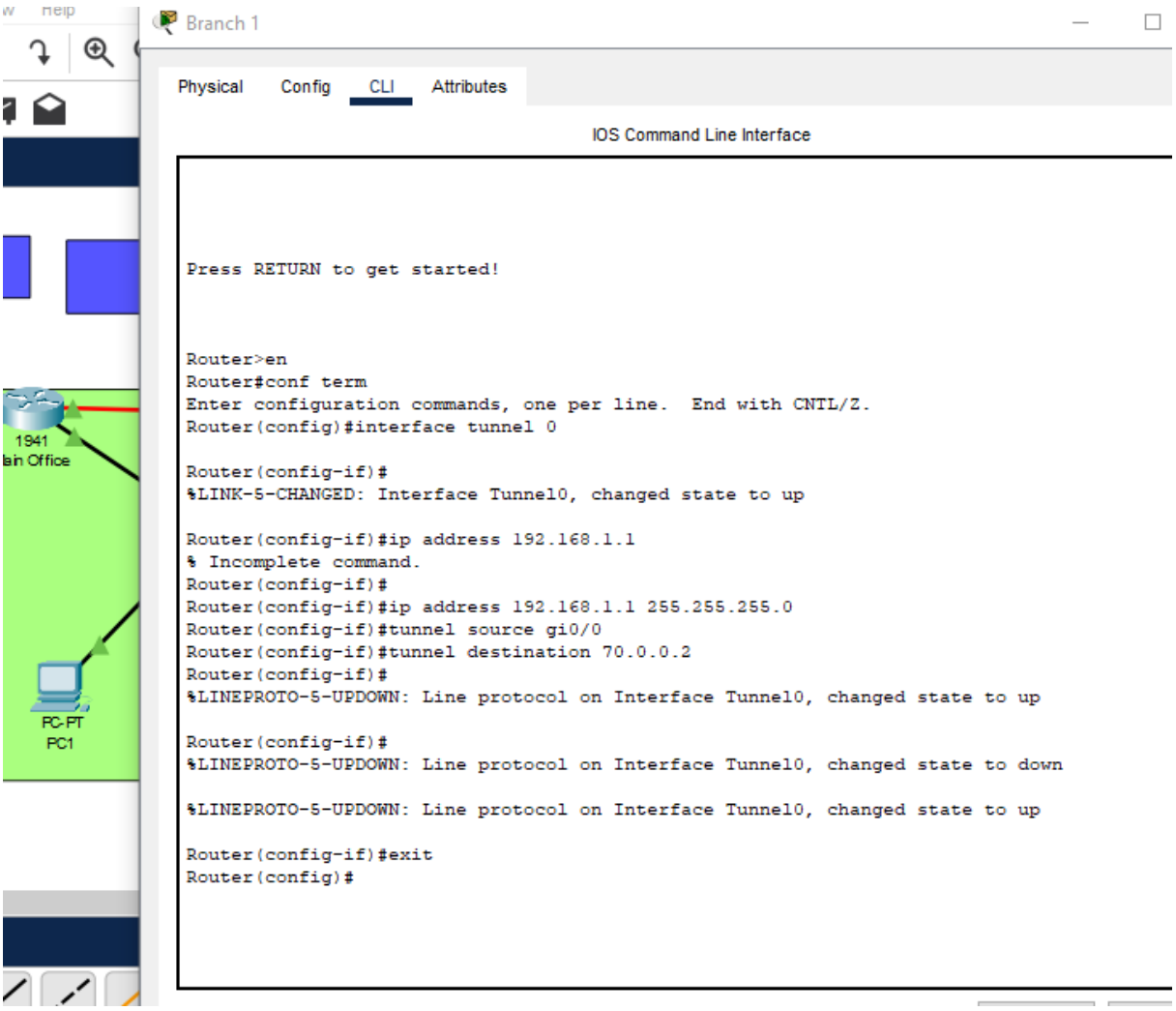


VPN Tunnel Configuration:

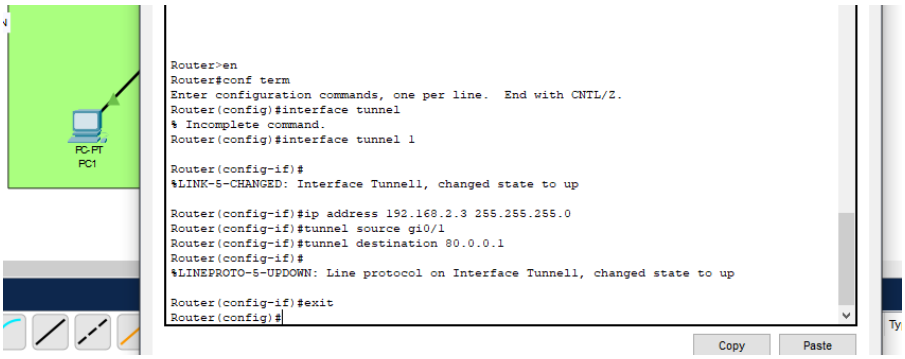
The routers are configured to establish an IPSec VPN tunnel between them. This tunnel allows secure communication between

the routers, even if they have duplicate LAN subnets. The VPN ensures that data exchanged between the routers is encrypted and secure.

BRANCH 1:



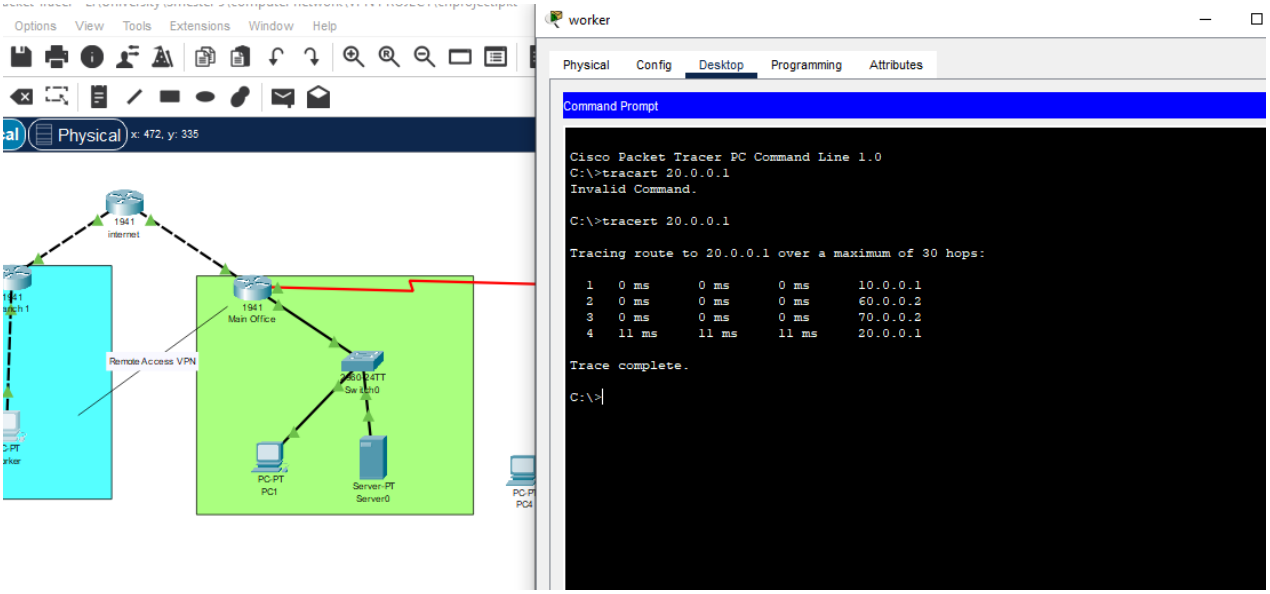
Last Router:

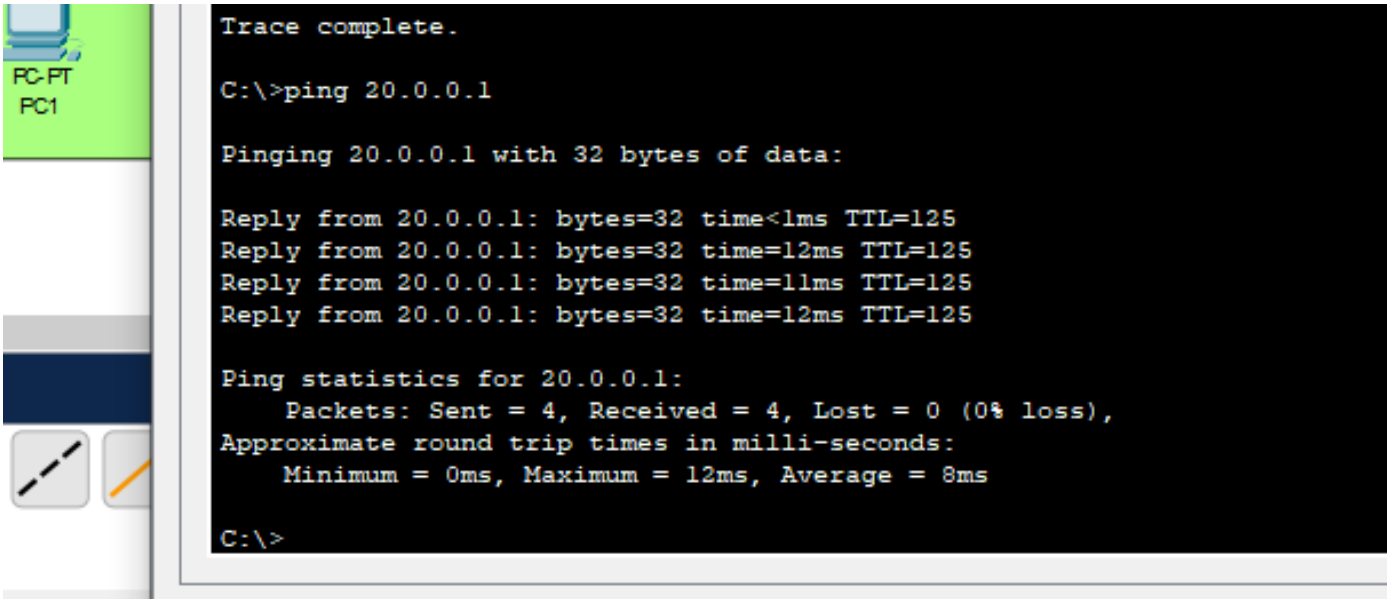


Switching and Local Network Connectivity:

The Cisco 2960-24 switch is used for local network switching. It provides efficient communication between devices within your local network, allowing devices to connect, communicate, and share resources seamlessly

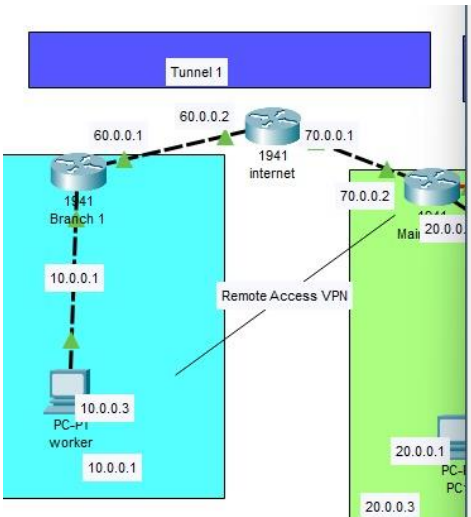
Worker 1 to Main Office:





Worker 1 is remotely accessing the main office, bypassing the router for direct connectivity. This streamlined connection enhances efficiency in accessing essential resources from the main office.

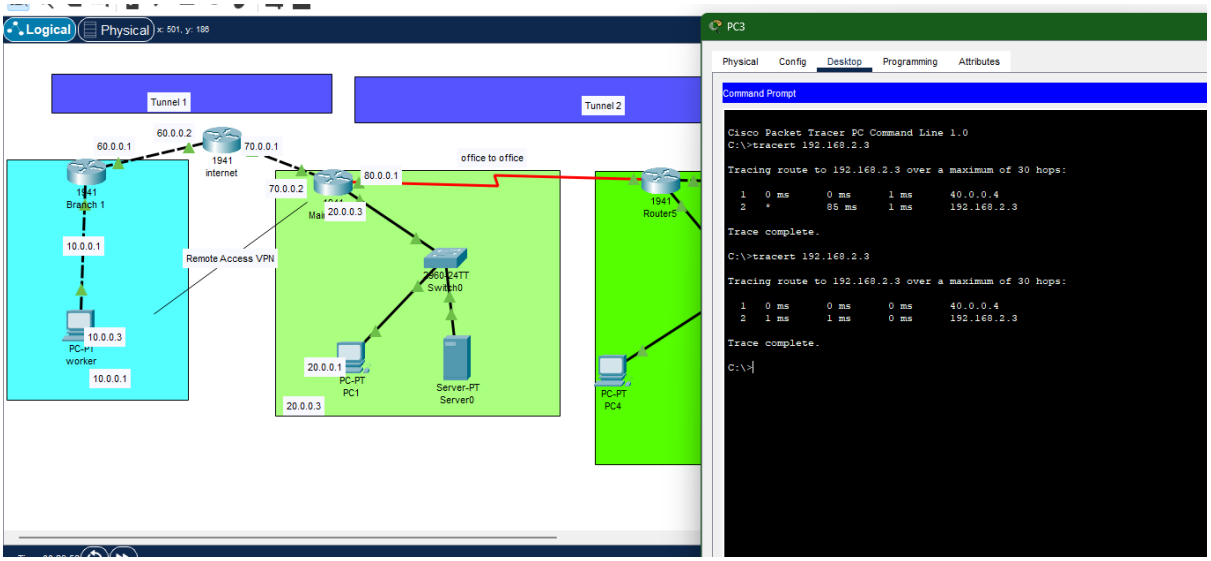
```
Reply from 20.0.0.2: bytes=32 time<1ms TTL=124
Reply from 20.0.0.2: bytes=32 time=20ms TTL=124
Reply from 20.0.0.2: bytes=32 time=10ms TTL=124
Reply from 20.0.0.2: bytes=32 time=1ms TTL=124

Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 20ms, Average = 7ms

C:\>tracert 192.168.1.2

Tracing route to 192.168.1.2 over a maximum of 30 hops:

  1    0 ms      0 ms      0 ms      10.0.0.3
  2    *         *         *         Request timed out.
  3    *         *         *         Request timed out.
  4    *         *         *         Request timed out.
  5    *         *         *         Request timed out.
  6    *         *         *         Request timed out.
  7    *         *         *         Request timed out.
  8    16 ms     12 ms     12 ms     192.168.1.2

Trace complete.

C:\>tracert 192.168.1.2

Tracing route to 192.168.1.2 over a maximum of 30 hops:

  1    0 ms      0 ms      0 ms      10.0.0.3
  2    0 ms      0 ms      0 ms      192.168.1.2

Trace complete.

C:\>
```



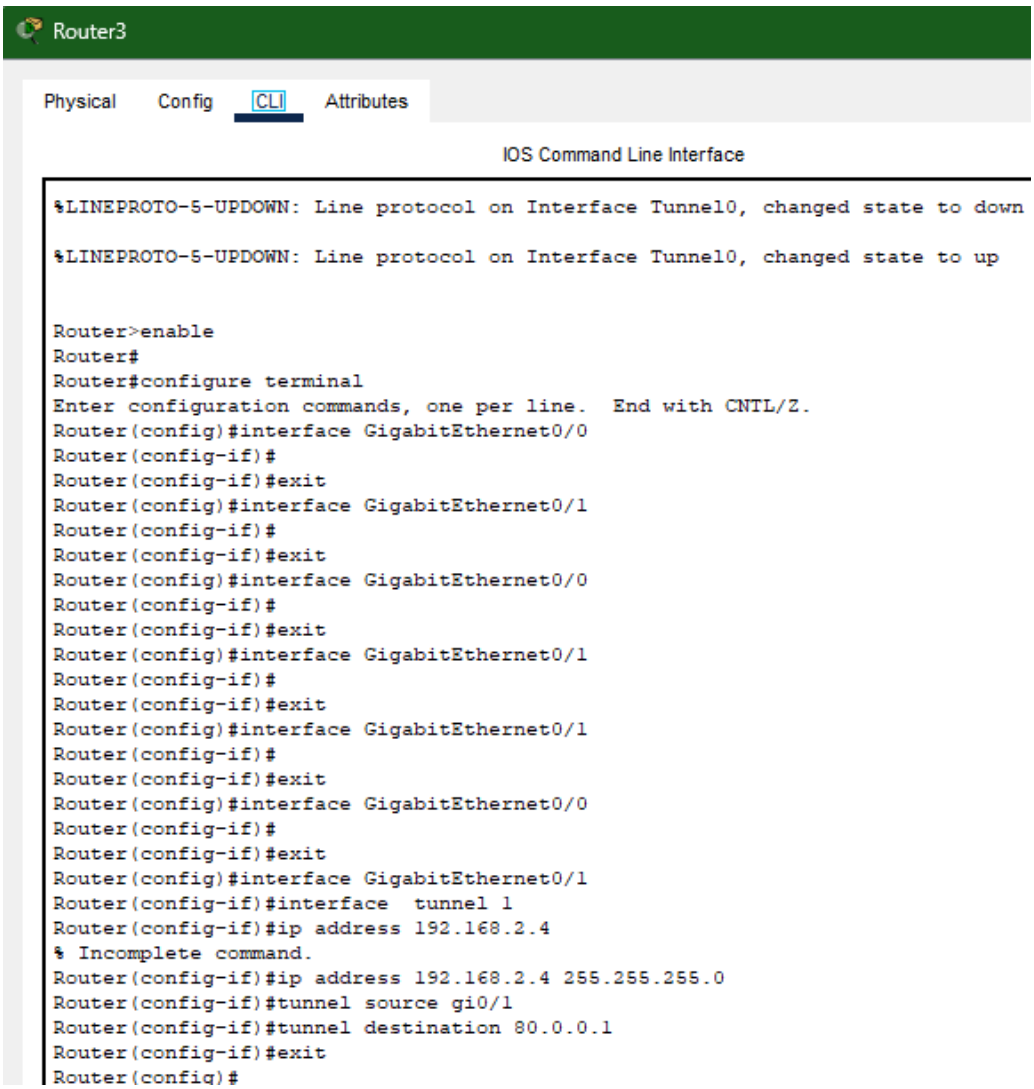

```
  1    0 ms      0 ms      0 ms      10.0.0.1
  2    0 ms      0 ms      0 ms      60.0.0.2
  3    1 ms      0 ms      0 ms      70.0.0.2
  4    0 ms      0 ms      0 ms      20.0.0.1

Trace complete.

C:\>tracert 20.0.0.1

Tracing route to 20.0.0.1 over a maximum of 30 hops:

  1    0 ms      0 ms      0 ms      10.0.0.1
  2    0 ms      1 ms      0 ms      60.0.0.2
  3    0 ms      0 ms      0 ms      70.0.0.2
  4    21 ms     0 ms      10 ms     20.0.0.1

Trace complete.

C:\>tracert 192.168.1.2

Tracing route to 192.168.1.2 over a maximum of 30 hops:

  1    0 ms      0 ms      0 ms      10.0.0.1
  2    0 ms      0 ms      0 ms      192.168.1.2

Trace complete.

C:\>tracert 192.168.1.2

Tracing route to 192.168.1.2 over a maximum of 30 hops:

  1    0 ms      0 ms      0 ms      10.0.0.1
  2    0 ms      0 ms      0 ms      192.168.1.2

Trace complete.

C:\>
```

Top

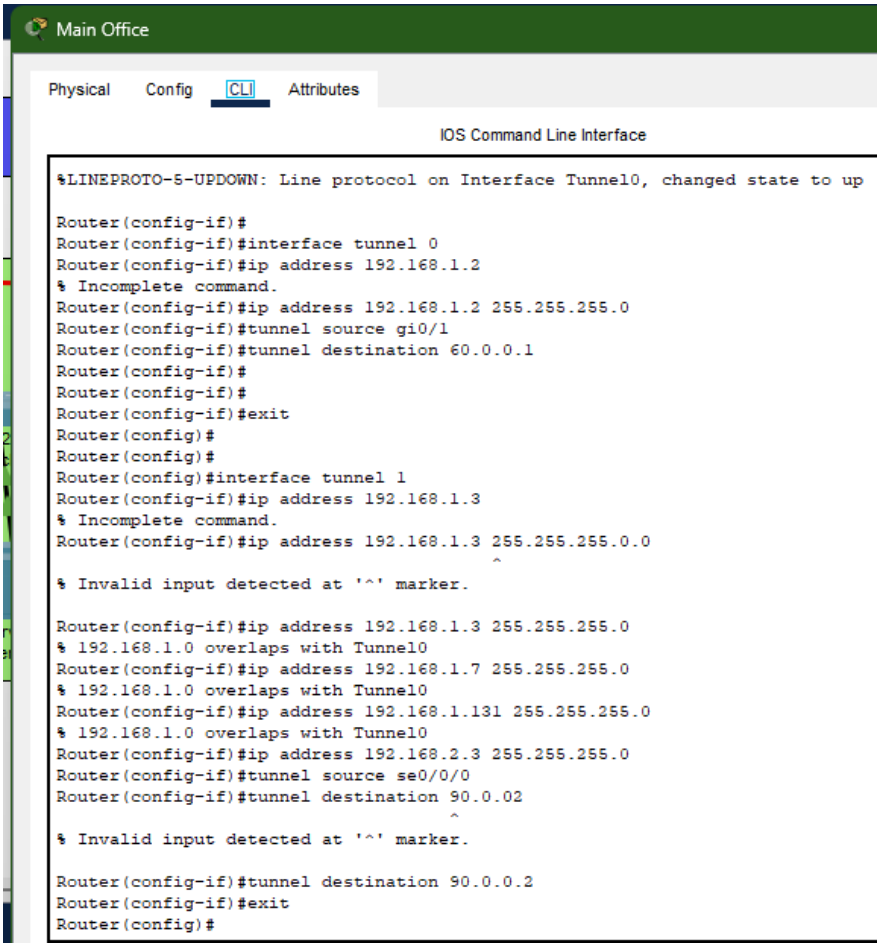
TUNNEL 2 ACCESSING FROM LAST Branch/WORKER:



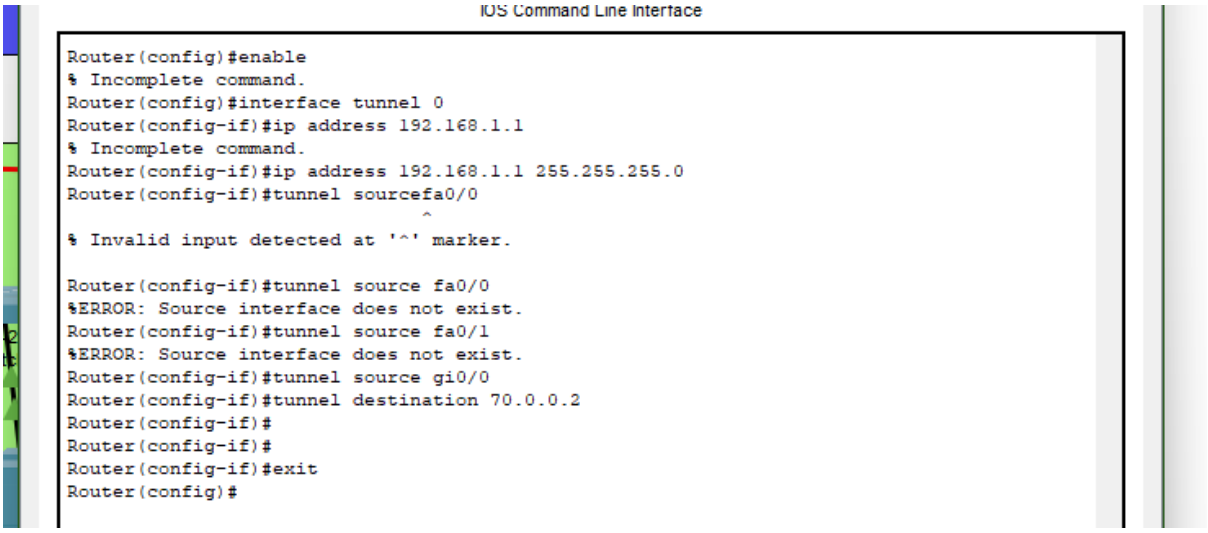
Setting Tunnel 2 for Last worker (PC3) and Main Office:



Main Office Tunnel Configuration:



Setting Tunnel 0 configuration in Branch 1:



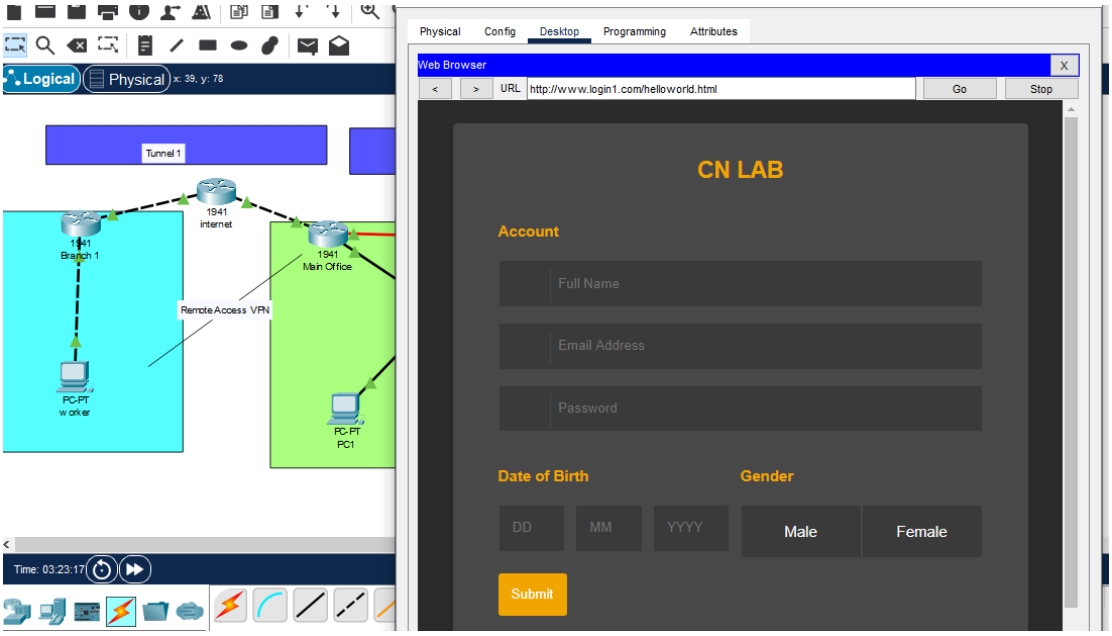
Server Functionality:

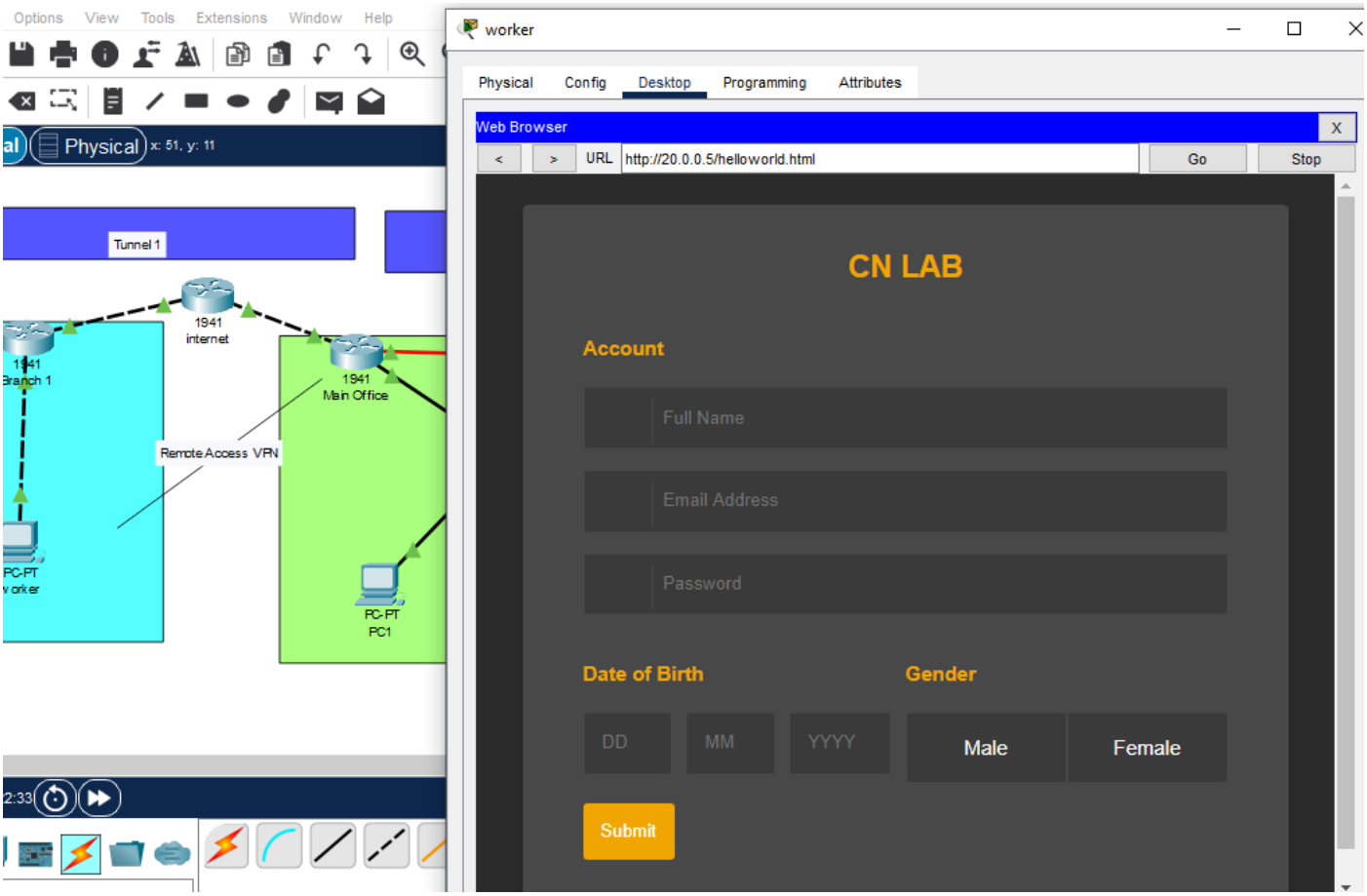
The server serves as a centralized hub for hosting various resources, including applications, databases, and shared files. It

provides a platform for user authentication and authorization,

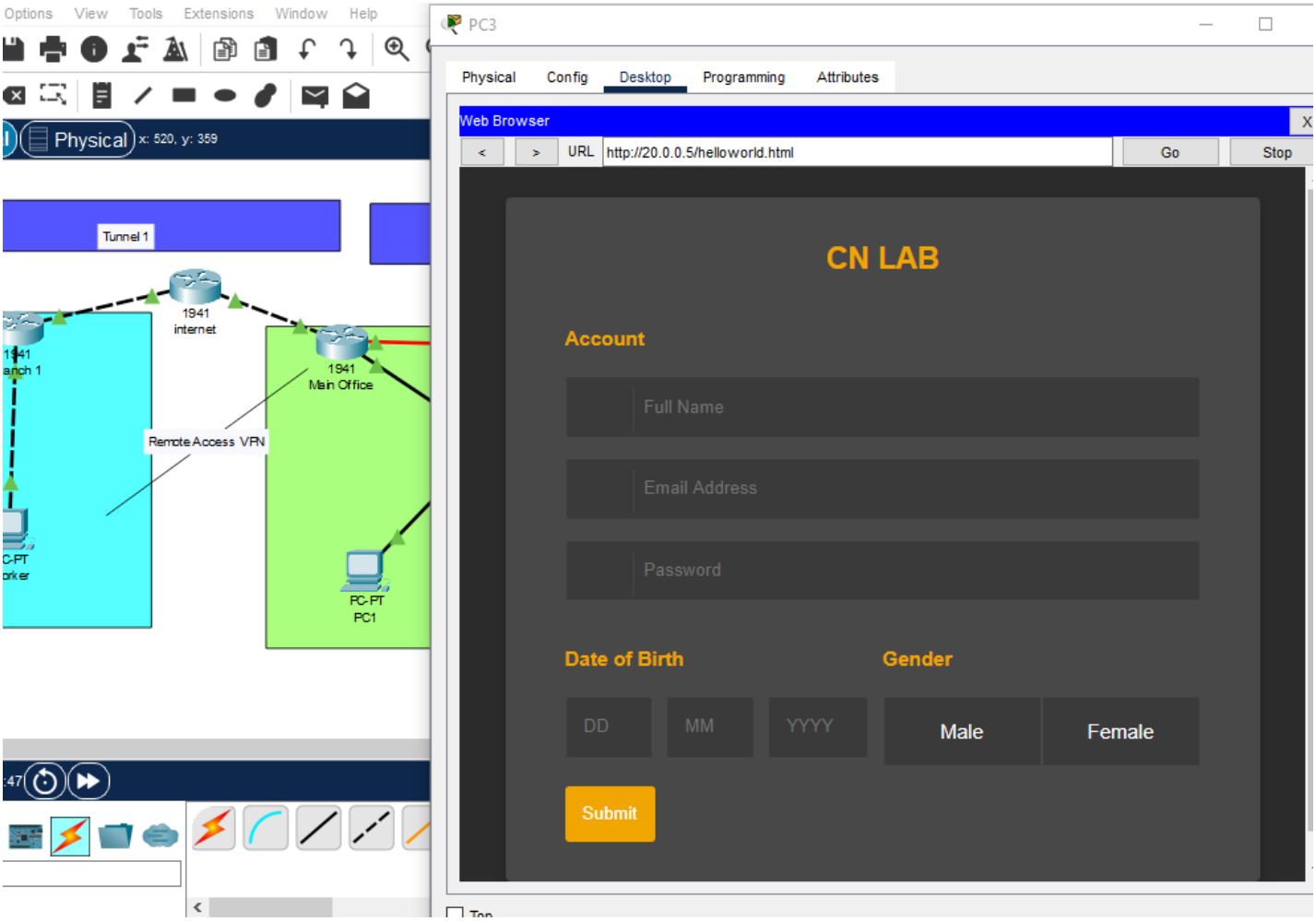
ensuring secure access to network resources.

ACCESSING SERVER FROM WORKER 1:



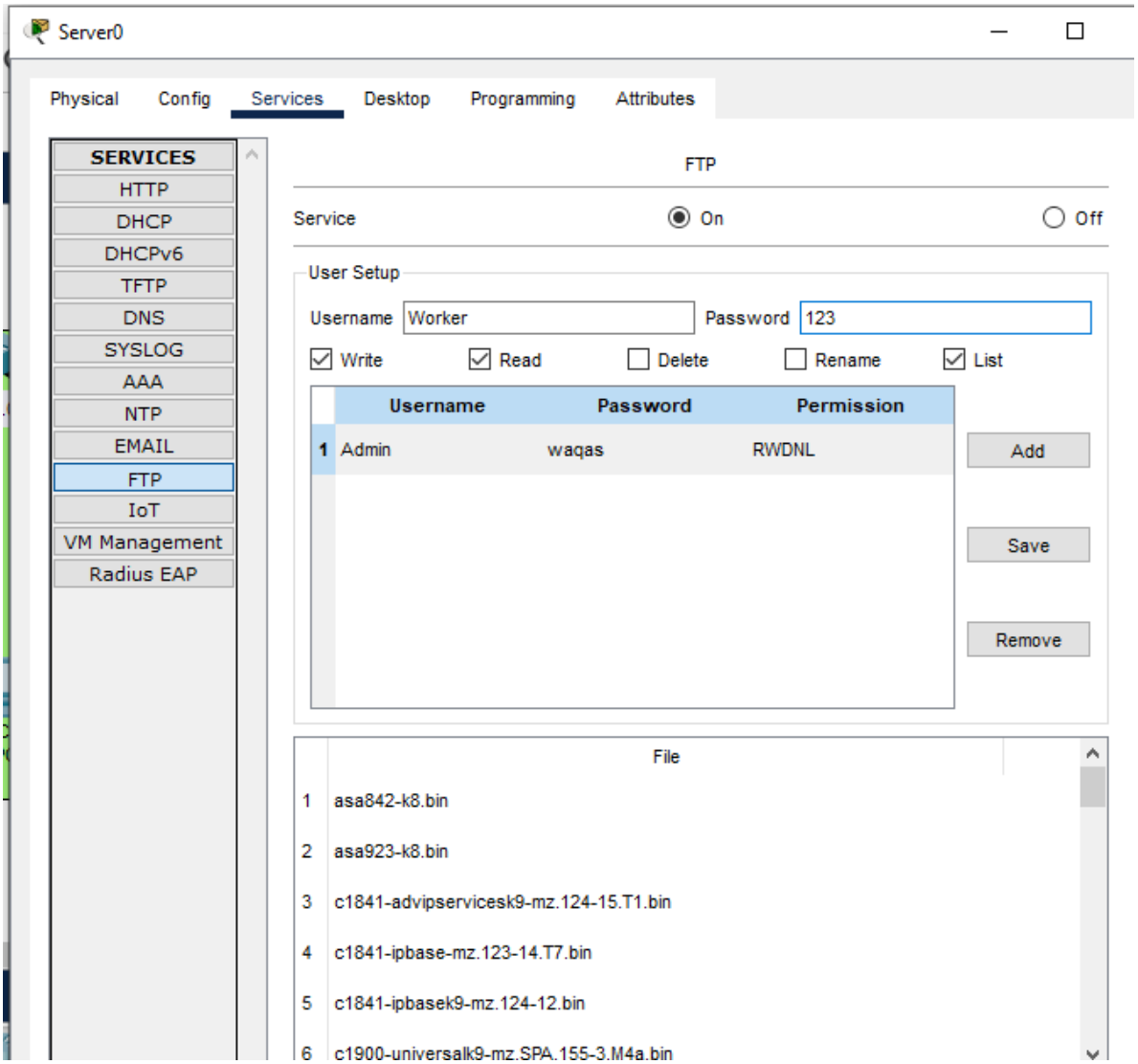


Worker 3 to Main Office:

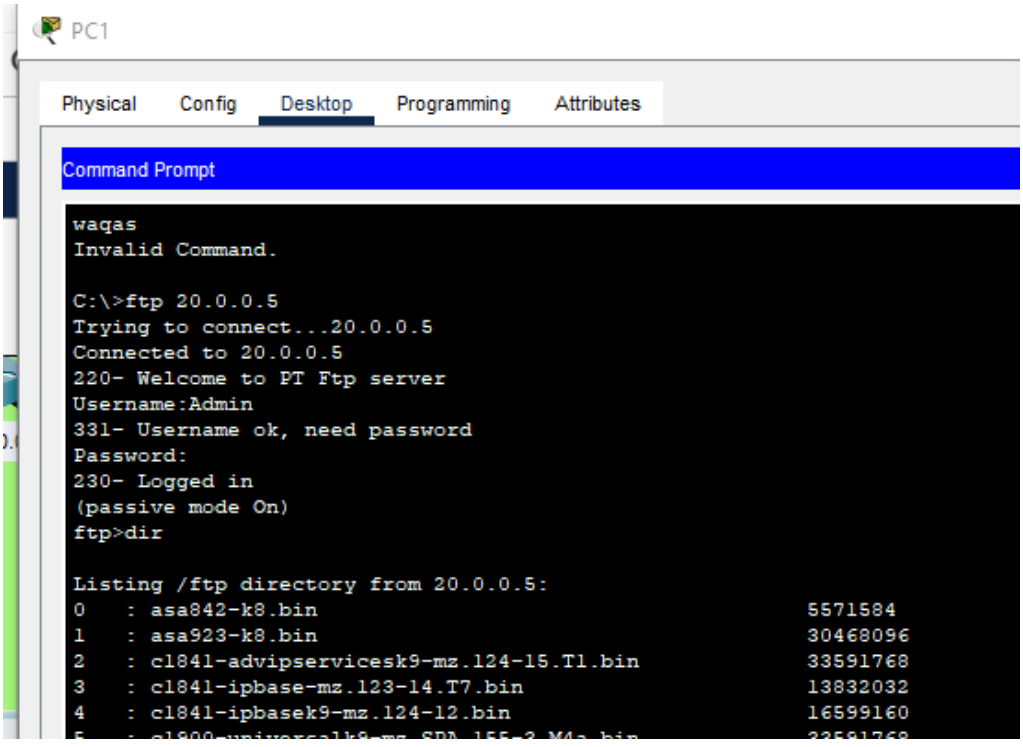


Setting Up the FTP (File Transfer Protocol):

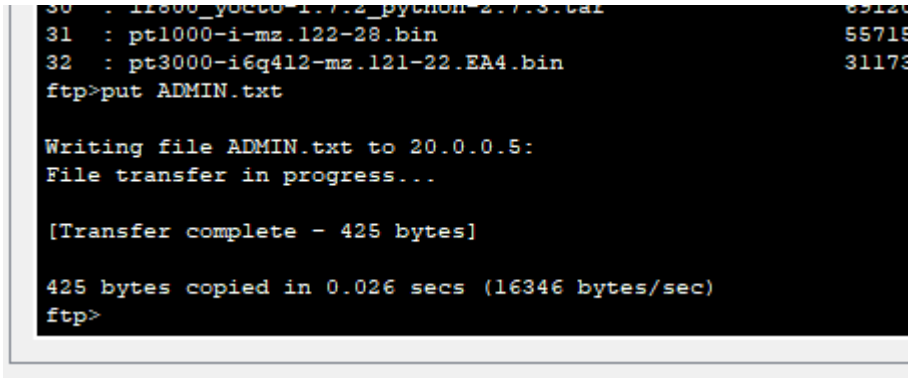
FTP (File Transfer Protocol) is a standard network protocol used for the seamless transfer of files between a client and a server on a computer network. It provides a reliable method for sharing files over the internet or an intranet, allowing users to upload, download, and manage files efficiently. FTP is commonly employed in scenarios where large files or numerous files need to be transferred between devices or servers.

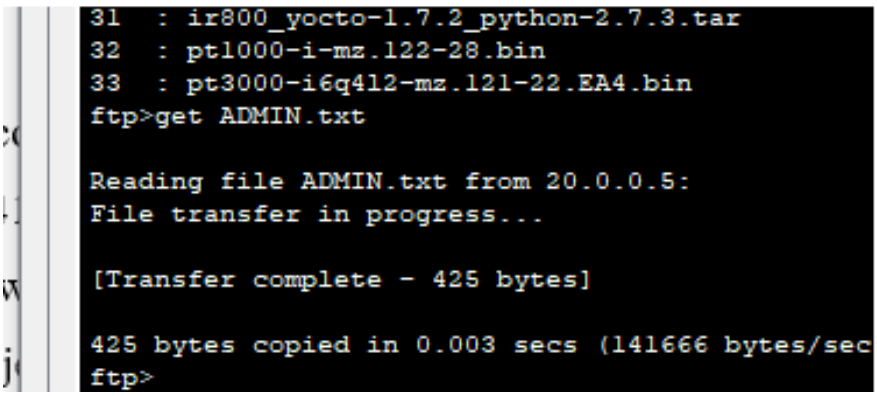


Accessing Directory through Admin PC:



File Uploading & Downloading from through admin pc:





Setting up the email (SMTP) on Server 2:

SMTP stands for Simple Mail Transfer Protocol. It is a protocol used for sending and receiving email messages over a network. SMTP is essential for the delivery of emails from the sender's email client to the recipient's email server. To set up SMTP on Server 2, you'll need the following information:

1. SMTP Server Address: The address of the server that will handle outgoing emails.

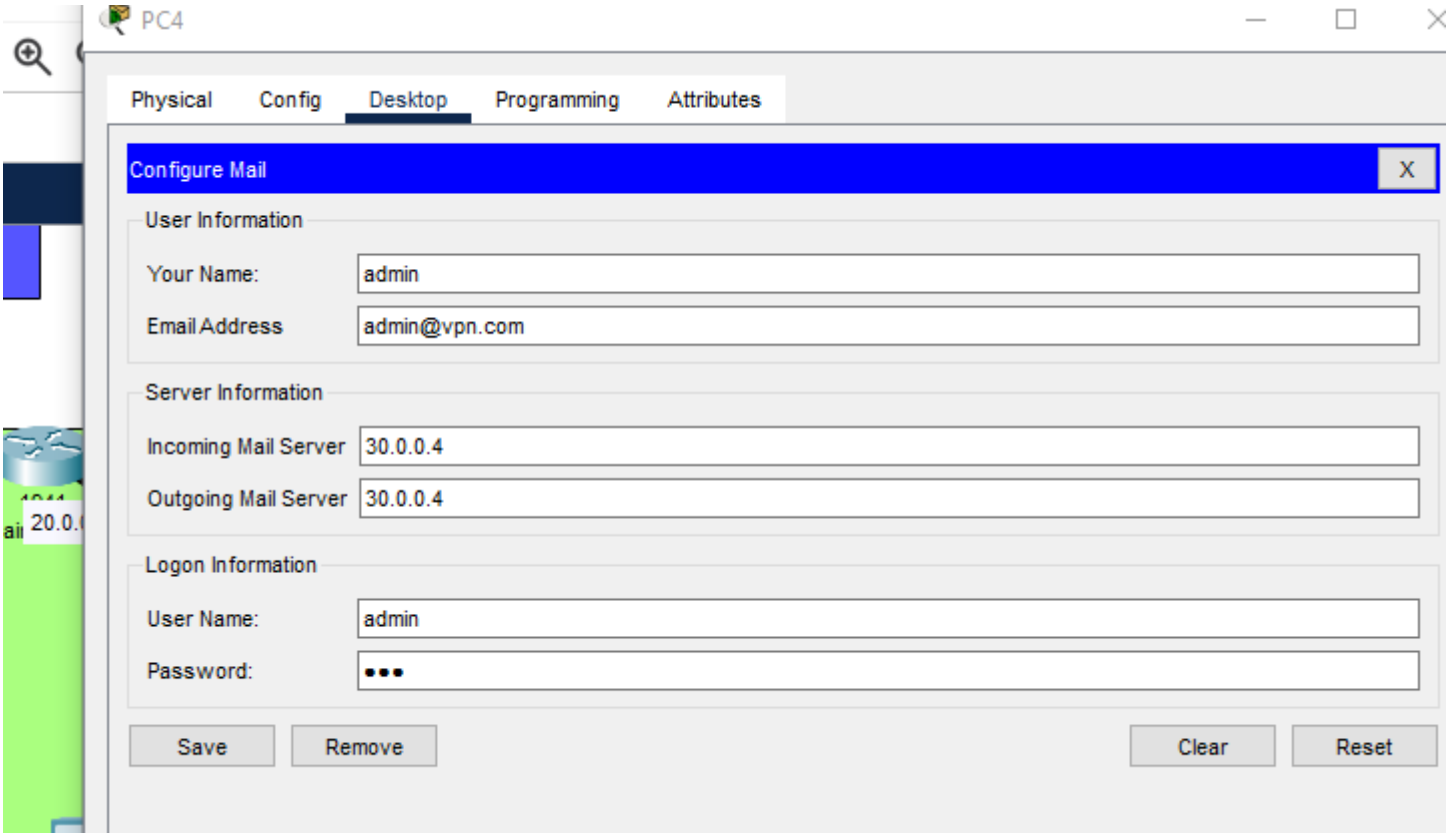
2. SMTP Port: The port number used for SMTP communication. Common ports are 25, 587, and 465.

3. Username and Password: Credentials required for authentication with the SMTP server.

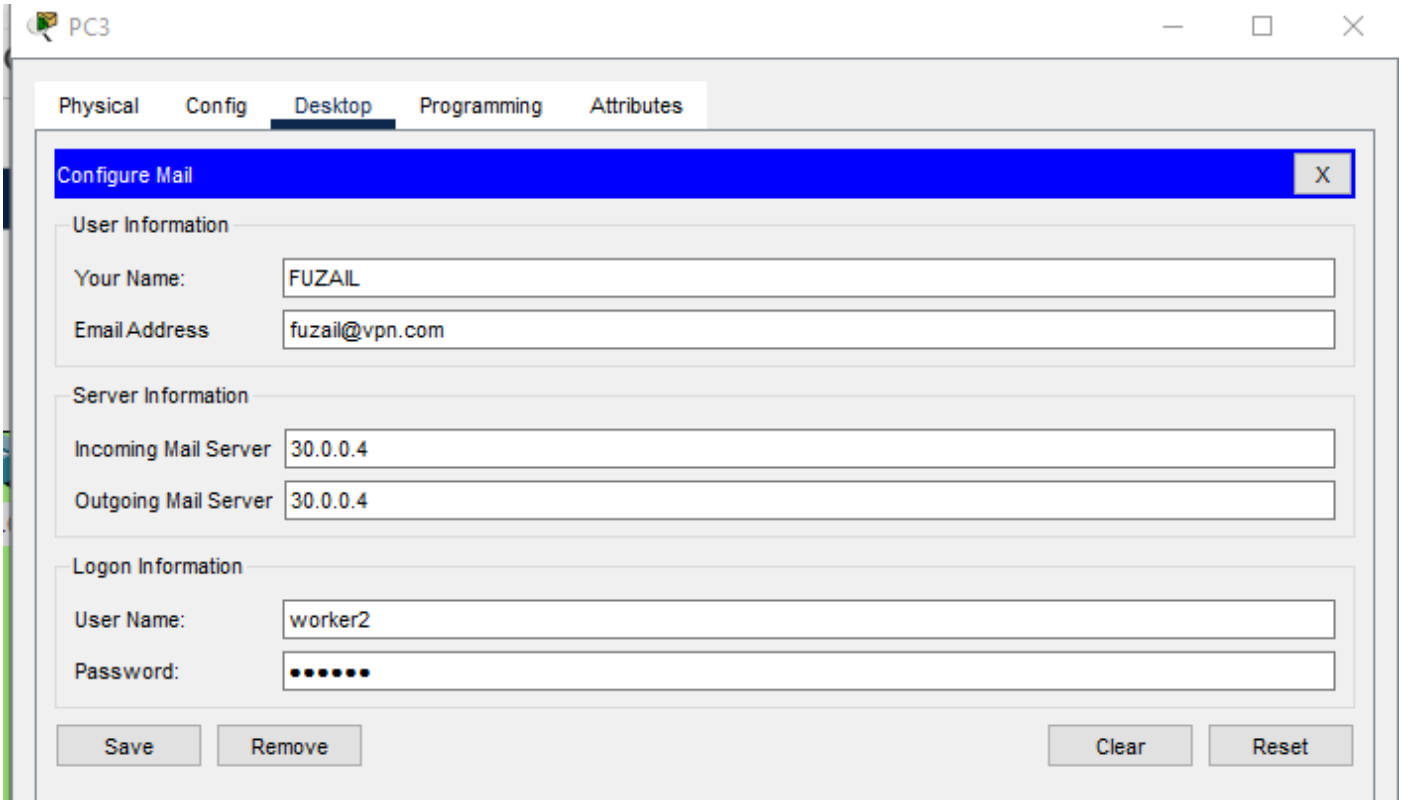
4. Encryption Method: Whether the communication with the SMTP server is encrypted, often specified as SSL/TLS.

ADMIN:

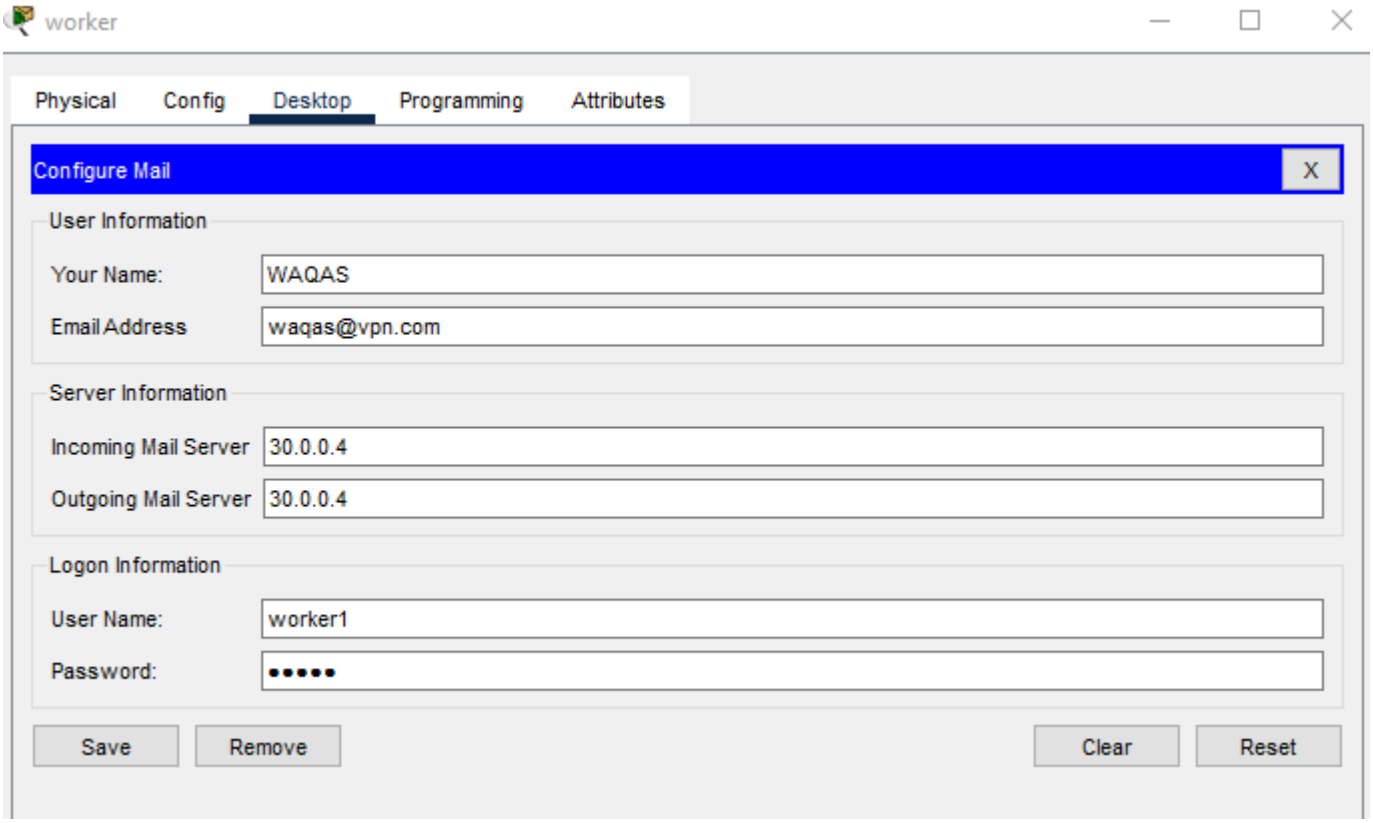
Admin initiates email communication between Worker 1 and Worker 2, leveraging the SMTP setup on Server 2. Admin coordinates email transmission from Worker 1 to Worker 2, utilizing the configured SMTP settings on Server 2 for seamless communication within the network.



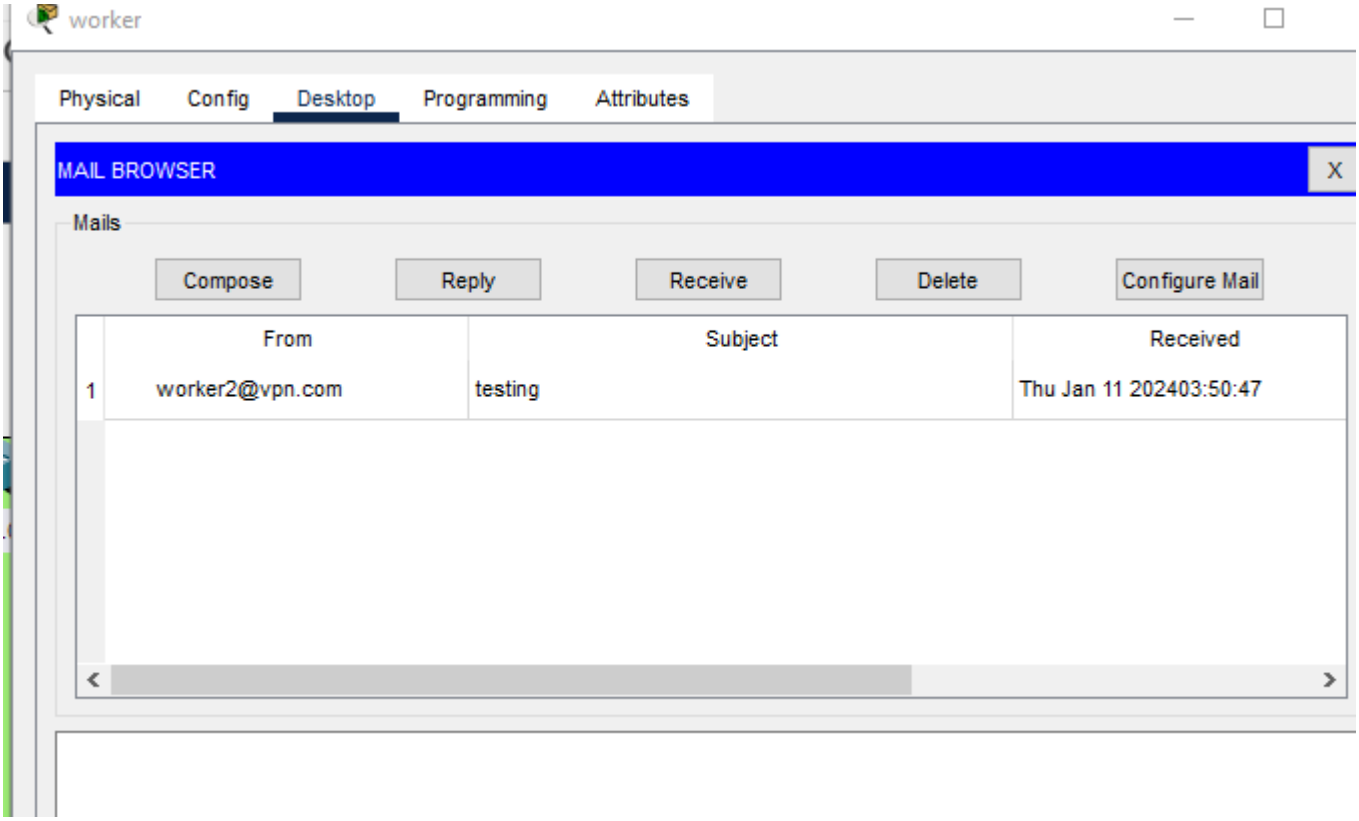
Worker1:

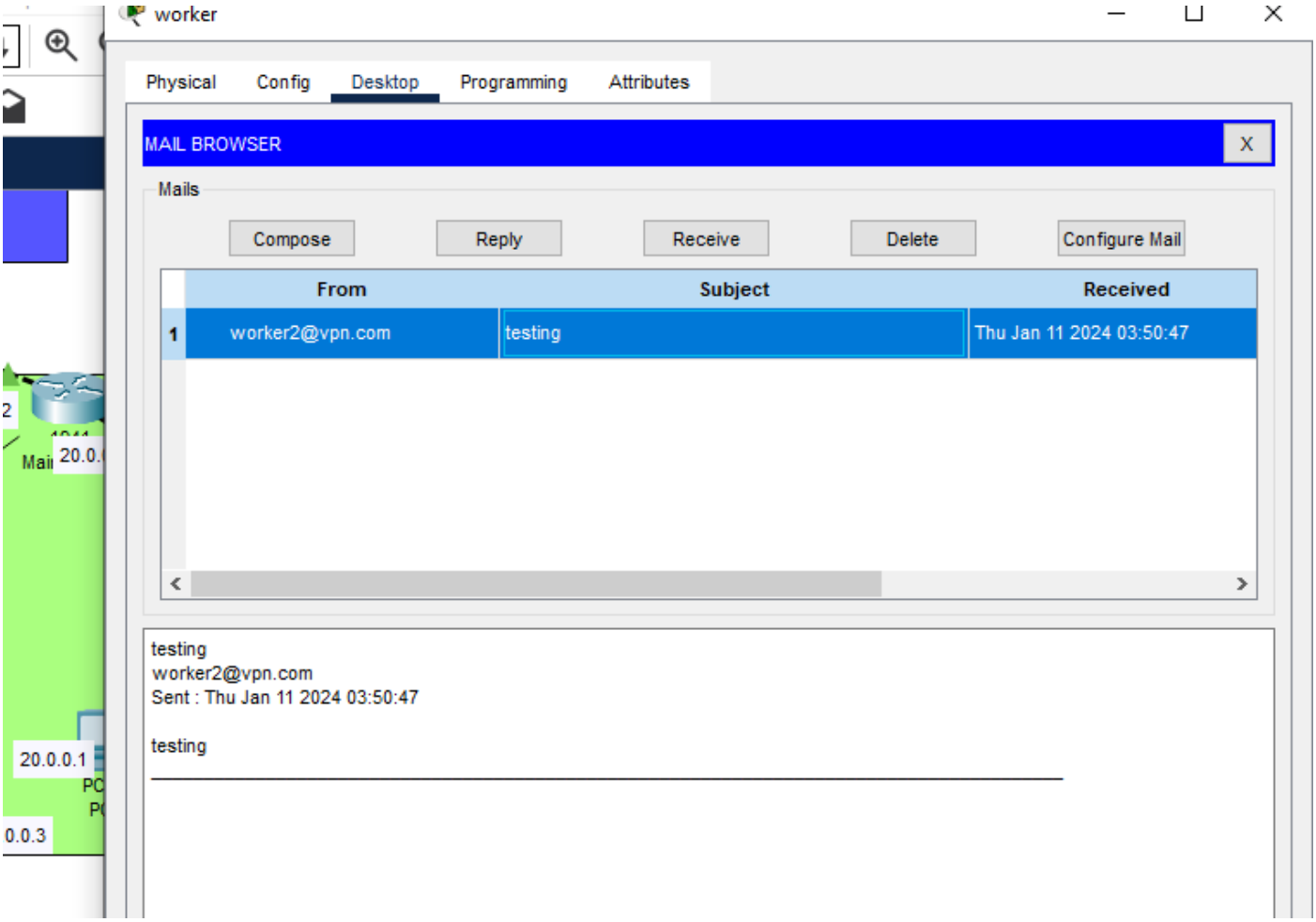


Worker2:

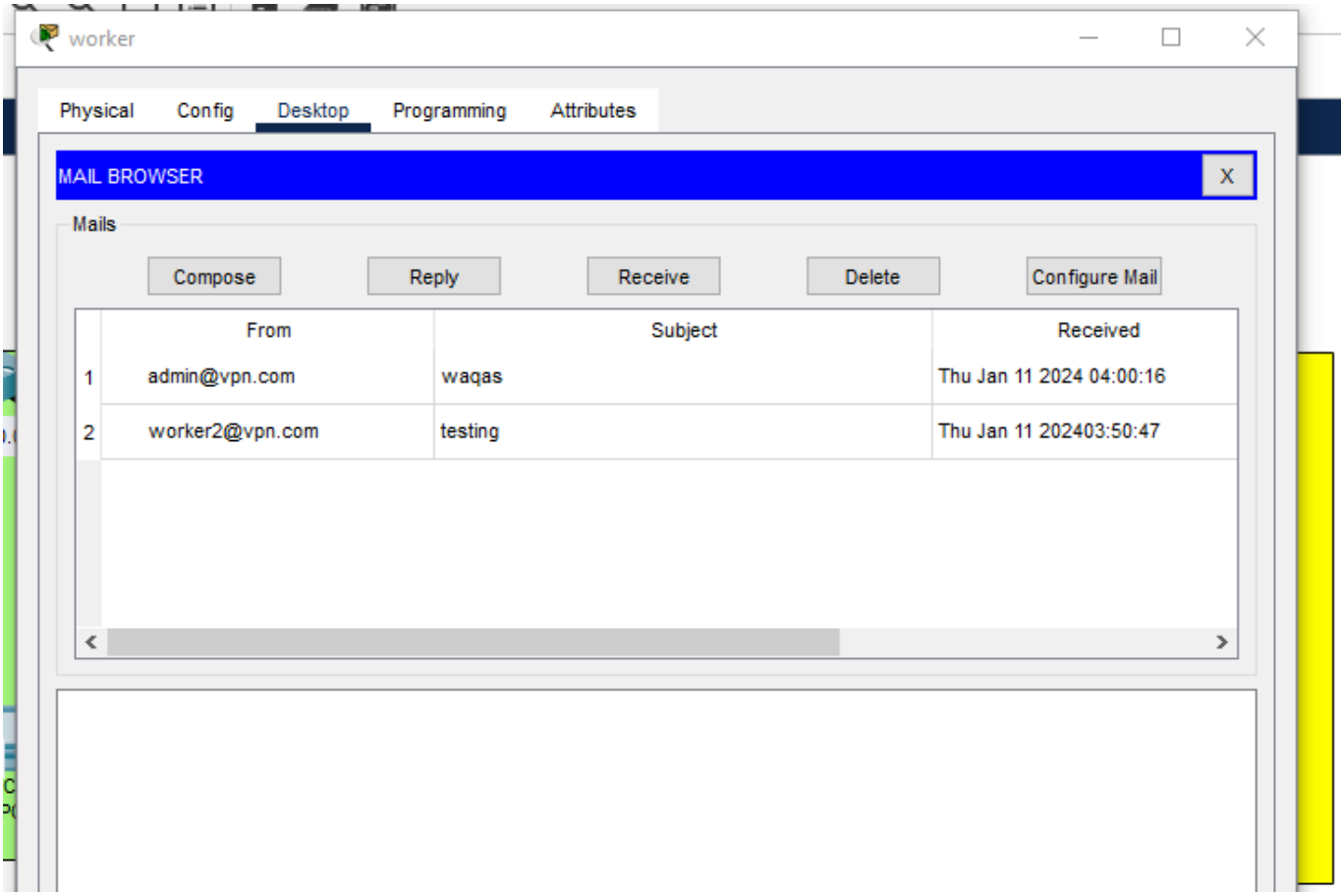


Sending mail from worker 1 to worker2: Worker 1 sends an email to Worker 2 using the established SMTP configuration on Server 2.





From Admin to waqas:



Conclusion:

In conclusion, the successful implementation of this project relies on the Cisco 1941 routers, which play a pivotal role in establishing a secure and efficient network infrastructure. Through the deployment of IPsec VPN tunnels, the project ensures data confidentiality and integrity, addressing challenges related to duplicate LAN subnets. The routers excel in routing tasks, enhancing network scalability and reliability. Emphasizing the strategic use of Cisco 1941 routers, the project showcases their adaptability, high-performance routing, and security features, particularly in firewall configurations and intrusion prevention. This streamlined approach underscores the routers' cost-effectiveness, providing a resilient and secure network infrastructure.

Reference Video:

Concept From: **https://youtu.be/LlL2DkFkACo?si=IPsOMUctbyxPouGO**

https://youtu.be/SYGdxsDApyM?si=zbzzzsL6C5M8rKBF

https://youtu.be/lkUq6Pl6his?si=1F5TJhH2QunpZBRV

https://youtu.be/8uWmFkrn6qE?si=eLMZ46x2v4IG6UuO