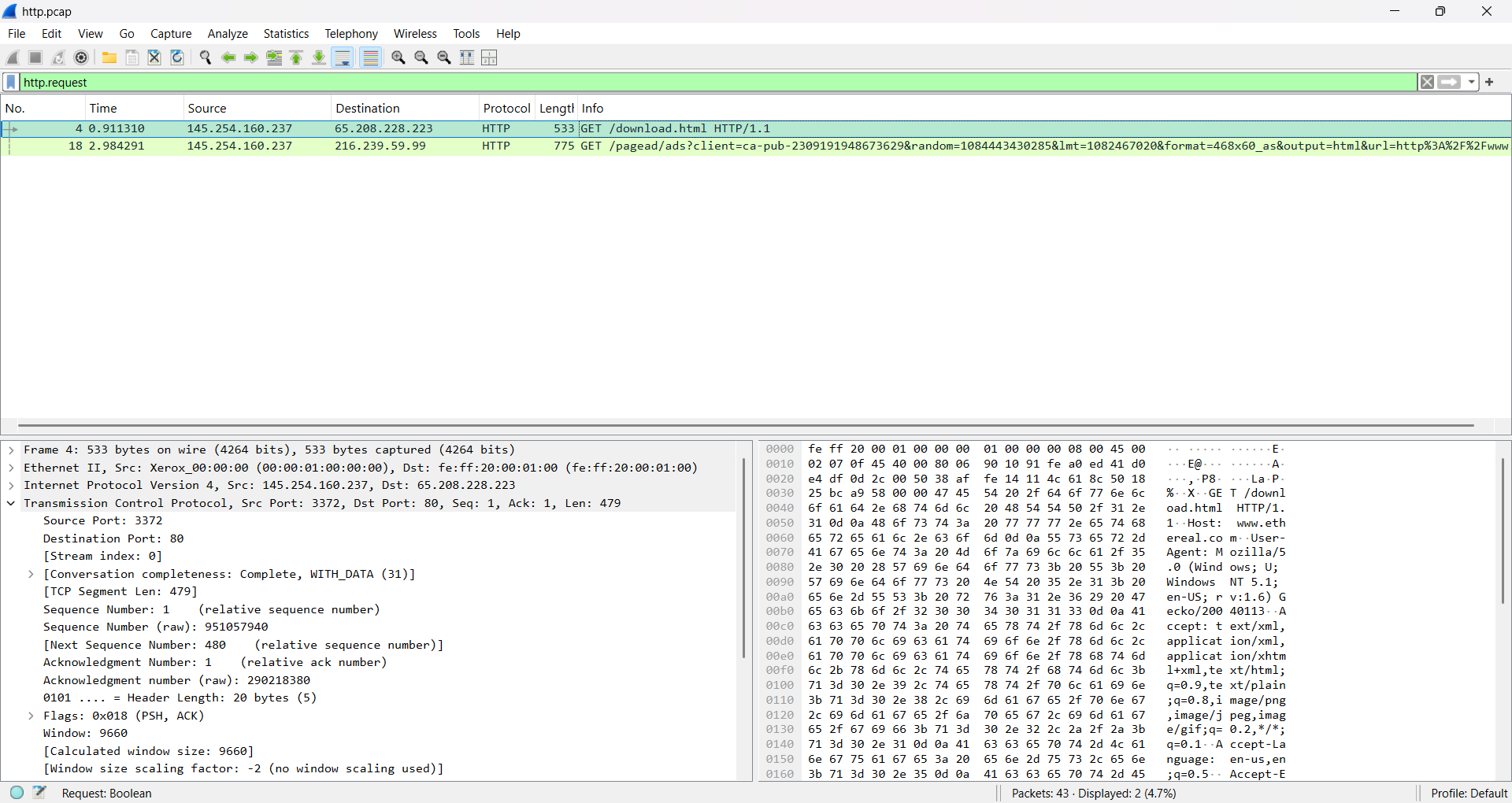
**HTTP pcap Documentation**

**Answering Objectives Process**

This whole project uses the http.pcap file for the Objective Questions and for the Analysis. Naturally, I started with the Objective Questions. For the first question to find the IP Address of the Main User, I did a search with http.request to see who has been sending the most requests. For the second question, I did a search for dns for the DNS Packets, to answer those set of questions. For the third question, I did a search for http, so I can easily answer that set of questions. For the fourth question, the answer was within the http requests, so I didn’t need to make a new search. For the last question, I just searched for tcp and scrolled down to the black lines. Then, I searched for the information in the packets for the questions in that set.



**Inspect Packets**

After looking through the packets, they all look like they were successful, except for two TCP packets. There are three different Protocols being used in these transactions, TCP, HTTP, and DNS. TCP is the Transmission Control Protocol, which ensures that your data will arrive safely and in order. HTTP is the Hypertext Transfer Protocol, which is used to fetch websites. DNS is the Domain Name System, which translate domain names into IP Addresses.

**Detect Anomalies**

Packet 36 and 37 are the two packets that were not successful with their transactions. Both are colored black, which means that there was a Bad TCP transaction. Number 36 is a Spurious Retransmission of Number 26, which means that the server didn’t received the Acknowledgment, so it transmitted the data again. Number 37 is a Duplicate Transmission of Number 28, which means that data was lost.

**Recommendations to fix the Anomalies**

To fix Spurious Retransmissions, first look at the Round Trip Time (RTT). High RTT can mean there is a lot of network congestion. Occasional incidents isn’t really something to worry about, but if it happens a lot, it would be good to look at the Sender’s Retransmission Timer. It could be set too low, causing this to be a reoccurring problem.

To fix Duplicate Transmissions, implement a timeout mechanism for the sender to transmit again if no acknowledgment is received within a certain time frame. Another fix is to use a Duplicate Acknowledgment Detection Mechanism to identify lost of a TCP ACK message by the receiver. The last fix would be to avoid congestion with a Congestion Control Algorithm.

