Lab 09 – TCP Filters

Author: Raymond Ng
Course Number/Section: IS 3413-006
Date: November 8, 2022

INTRODUCTION

The purpose of this lab was to allow the user to experiment with TCP filters in Wireshark. Moreover, it allows the user to become more familiarized with the use of Wireshark.

PROCESS

I will be using the `PCAP` file, `THMWeb.pcapng`, provided by *TryHackMe.com* for the entirety of this lab in Wireshark. [1]

1. Applying `tcp.flags.syn == 1` filter in Wireshark, resulted in `22` packets with the TCP SYN flag set (*Figure 1*).

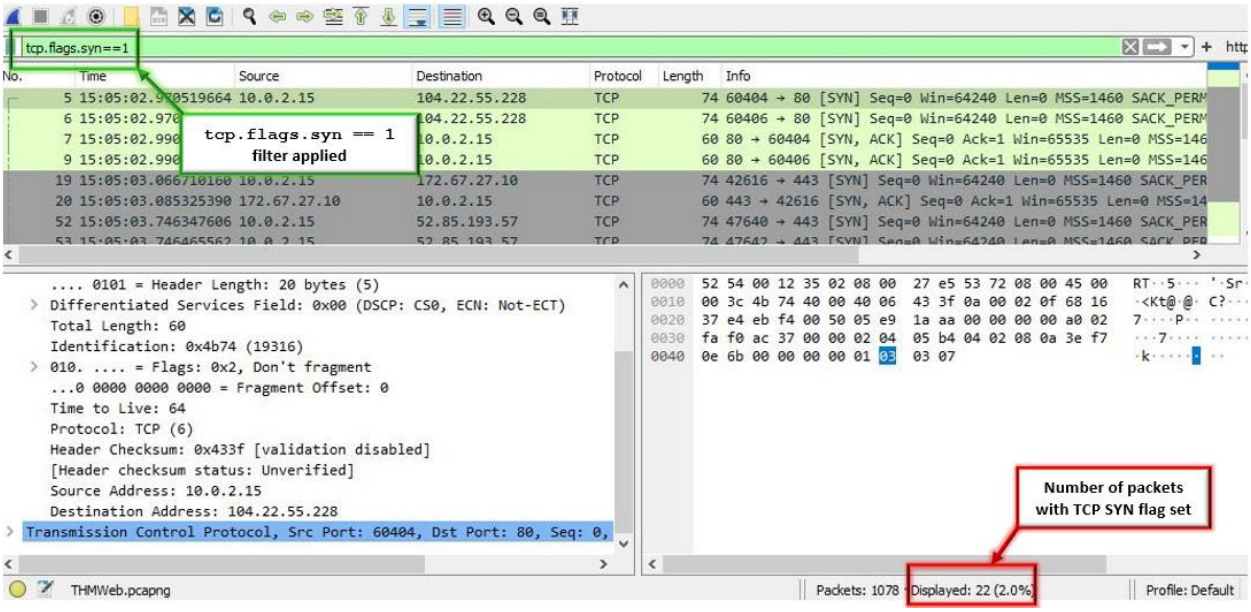

Figure 1: Depicting all packets with TCP SYN flag set via Wireshark

2. Applying `tcp.flags.fin == 1` filter in Wireshark, resulted in `11` packets with the FIN flag set (*Figure 2*).

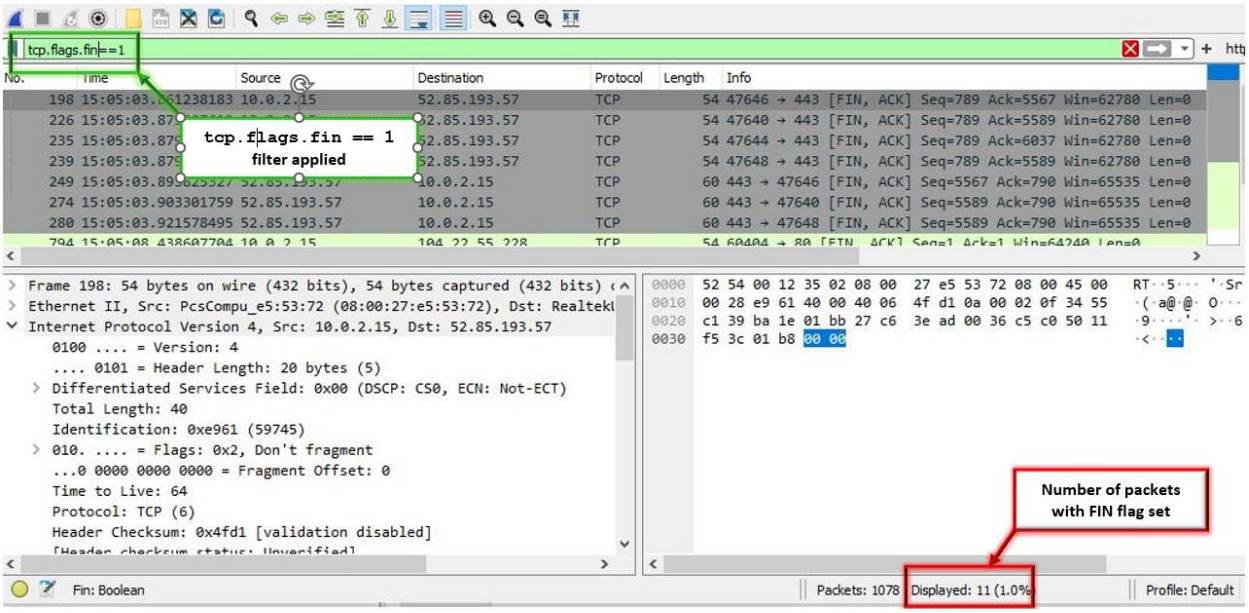

Figure 2: Depicting all packets with FIN flag set via Wireshark

3. Applying `tcp.flags.reset == 1` filter in Wireshark, resulted in `4` packets with the TCP resets (*Figure 3*).

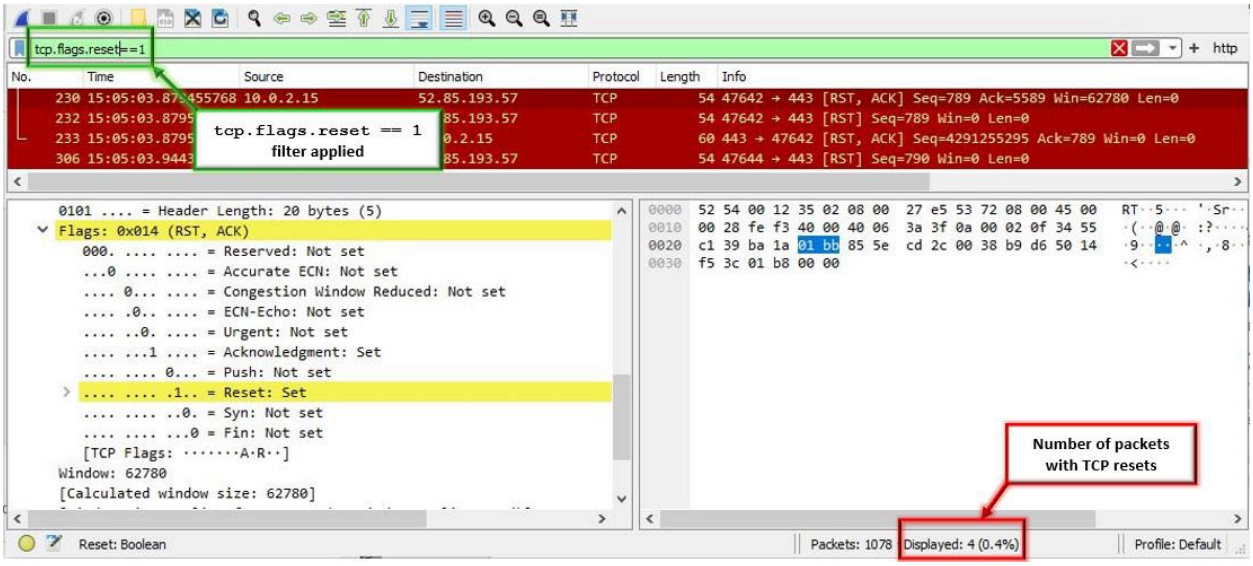

Figure 3: Depicting all packets with TCP resets via Wireshark

4. Applying `tcp.port==80 and tcp.flags.syn==1 and tcp.flags.ack==0` filter in Wireshark, I discovered 4 SYNs where sent on TCP port 80, not including TCP SYN/ACKs (*Figure 4*).

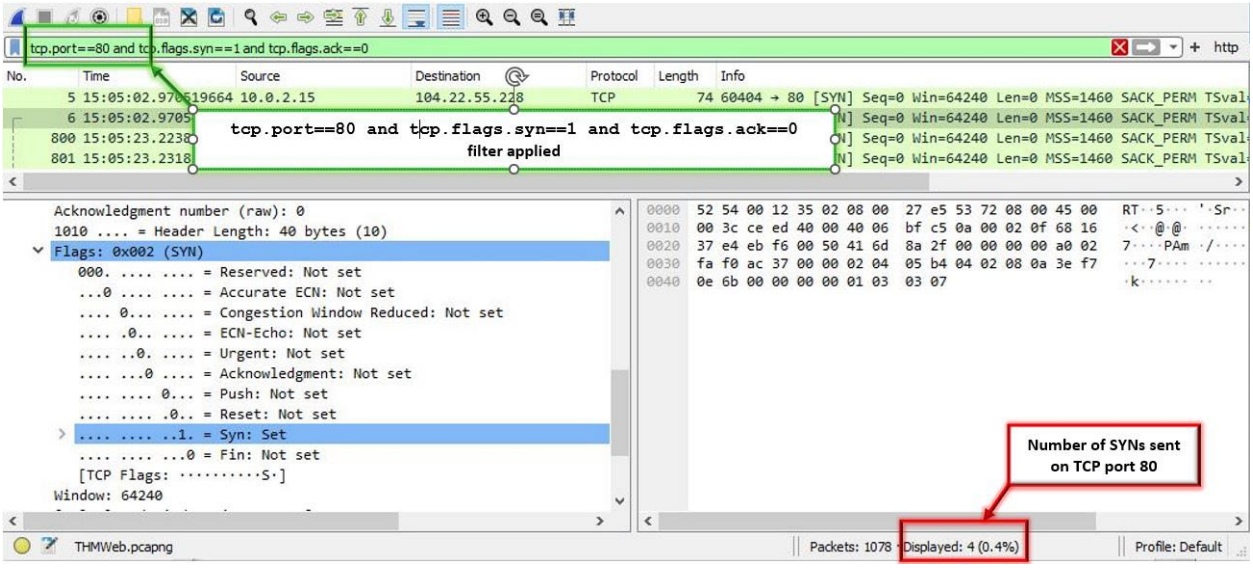

Figure 4: Depicting SYNs sent on TCP port 80 via Wireshark

Of note, per the Chris Greer video via YouTube, the `tcp.port==80 and tcp.flags==2` filter could have been used as well because of the minimal results and by observing the `Flag` contents in `Packet Details Pane` in Wireshark. However, applying the aforementioned filter is discouraged because the user may miss some of the desired results from the original query via Wireshark. [2]

5. To find the busiest TCP connection in the `PCAP` file, I went to the `Statistics` menu and selected `Conversations`, which opens a new window. In the `Wireshark·Conversations·THMWeb.pcapng` window, I clicked on the `TCP` tab and filtered the packets by clicking in the `Packets` column, which revelead `47650` client port number as the busiest TCP connection. (*Figure 5*)

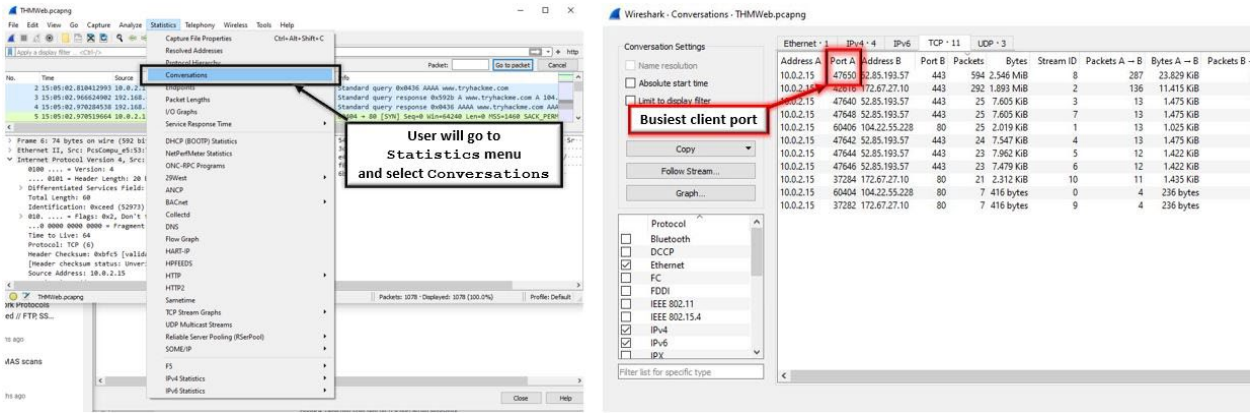

Figure 5: Depicting user's steps to find the busiest TCP connection, client port number 47650

6. Observing the same window from the previous step, looking at the `TCP` tab, I could see that there were `11` unique TCP connections/conversations in the `PCAP` file (*Figure 6*).

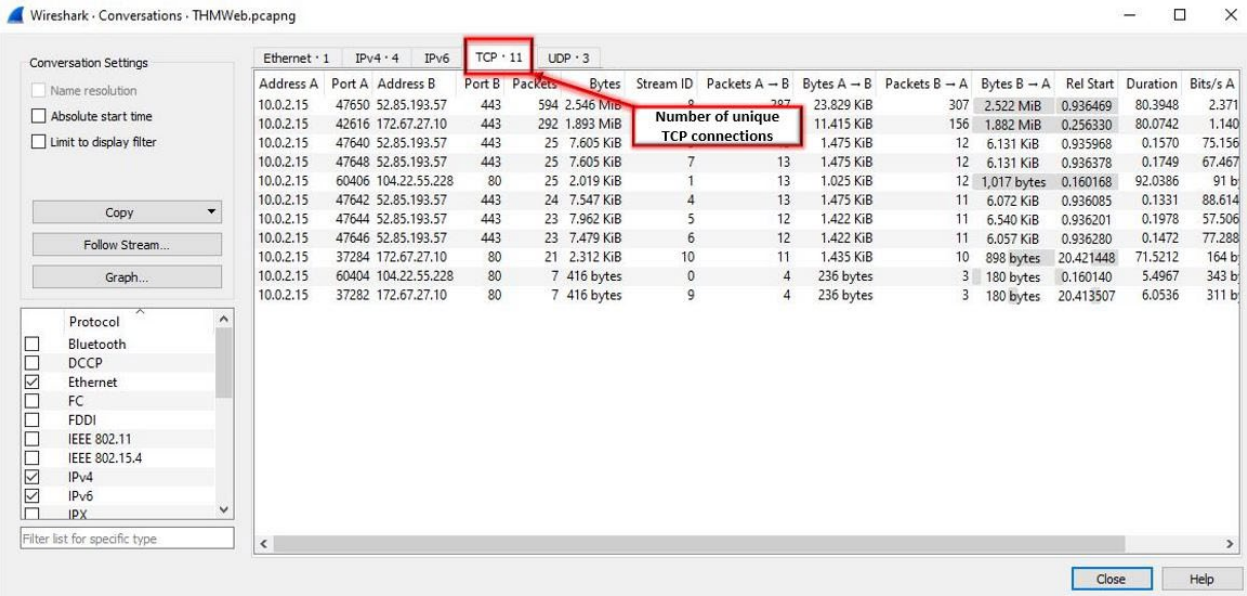

Figure 6: Depicting the number of unique TCP connections via Wireshark

7. `tcp.analysis.flags` would be the filter used to find TCP errors (*Figure 9*). In this PCAP file, there 32 packets with TCP errors (*Figure 7*). [2]

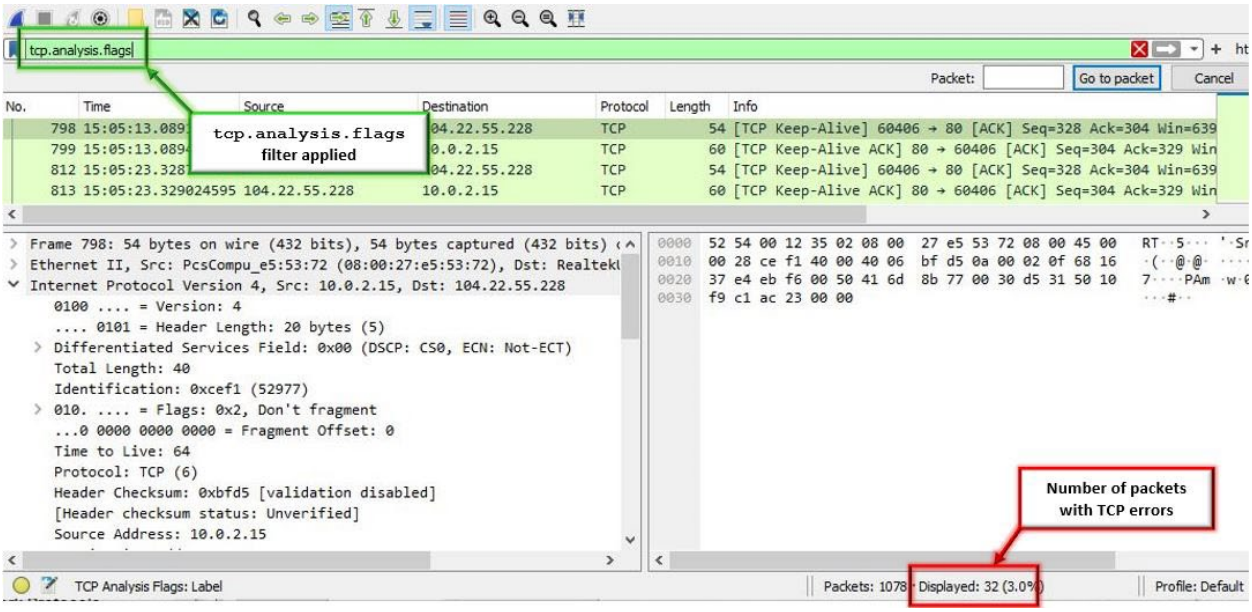

Figure 7: Depicting the number of packets with TCP errors via Wireshark

8. The `in` operator allows users to quickly filter for several different ports (*Figure 9*). For example, to query for packets in Ports 21, 23, 25, I would apply `tcp.port in {21,23,25}` filter. [2]

9. Applying `tcp.port in {47640..47650}`, resulted in `714` packets, which are all the traffic on Ports 47640-47650. (*Figure 8*).

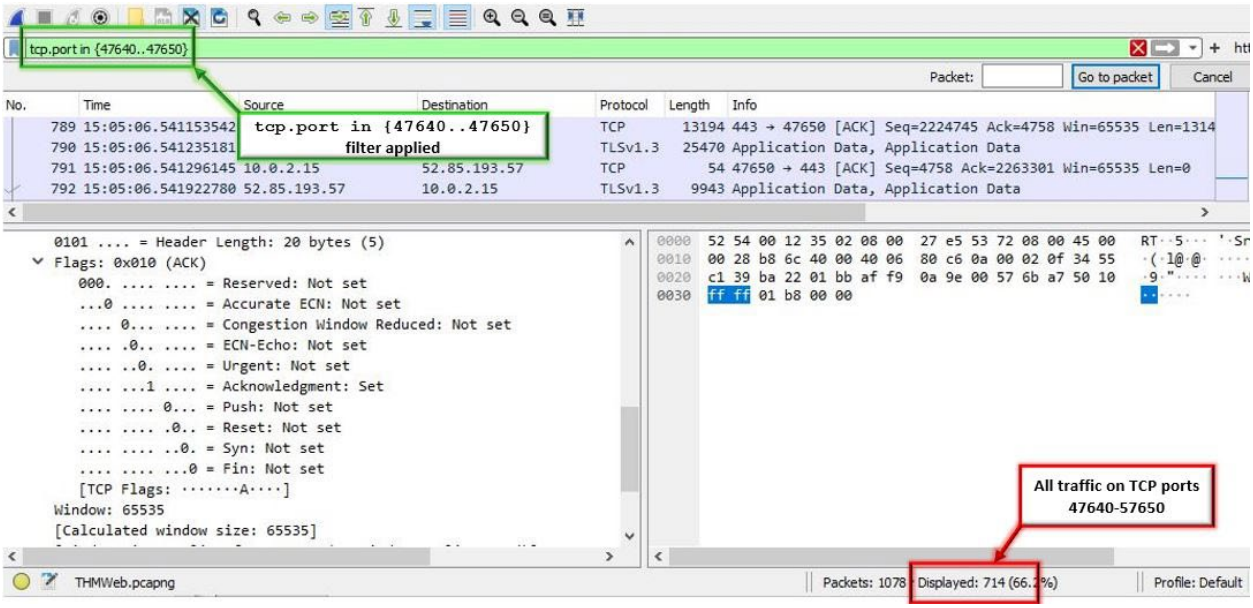

Figure 8: Depicting all traffic on TCP ports 47640-47650 via Wireshark

Figure 9: Validating user's responses/answers in Steps 7 & 8 via TryHackMe.com

LIMITATIONS/CONCLUSION

As an introductory experiment for the novice user like myself, I thought the lab's difficulty was simple. I do not assess there were any limitations because everything was executed in a live environment versus a controlled environment, like on a virtual machine. The biggest takeaway from this lab was learning how to use proper symbols and syntax to execute filters in Wireshark for the user's desired results.

REFERENCES

[1] TryHackMe [Online]. "Wireshark Filters", 2022. Available: https://tryhackme.com/room/wiresharkfilters [Accessed: 08-Nov-2022]

[2] Greer, *YouTube* [Online]. "TryHackMe WIRESHARK Filters Walkthrough", September 6, 2022. Available: https://www.youtube.com/watch?v=-MLkdg4s4ew [Accessed: 08-Nov-2022]

COLLABORATION

The entirety of this lab was executed independently by the author. No additional collaboration to report.