COMP 4108 – COMPUTER SYSTEMS SECURITY

Experience 3 – WireShark - Network Auditing For Anomaly Detection and Network Security

Student Name: Ben Cendana
Student #: 100811212
Date: April 10 2018

Table of Contents

1.0: Introduction

An underappreciated yet effect method in preventing the spread of viruses thought the network is anomaly detection. But In order to implement effective anomaly detection we must be able to perform some kind of network audit that collects and intercept all incoming network traffic. Once this data is collected we can then determine what the networks behaviour is.

The purpose of this experience is to become better familiar with a network packet analyser called Wireshark and understand how to use it to its full potential, once we fully understand how to use the software we can apply it to intrusion detection.

By the end of this experience we should have mastered all of the main Wireshark features.

2.0: Performing A Capture

The very first step is to setup the capture interface for the environment (Figure 1), since the primary internet connection is wifi we will use WI-FI as the interface type and make sure *Use promiscuous mode* is selected to allow all packets to be captured.

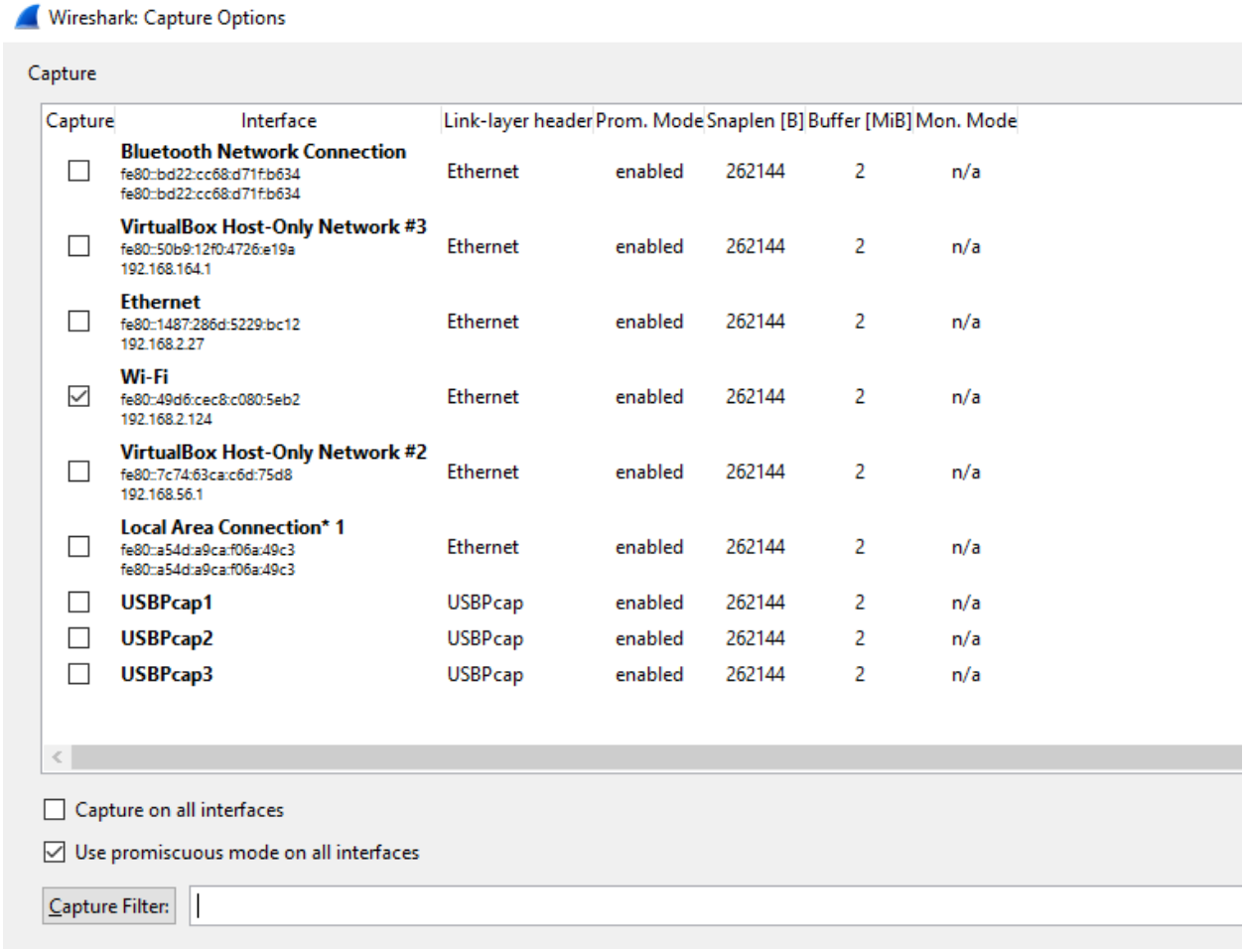

Figure 1: Setting Up The Capture Interface

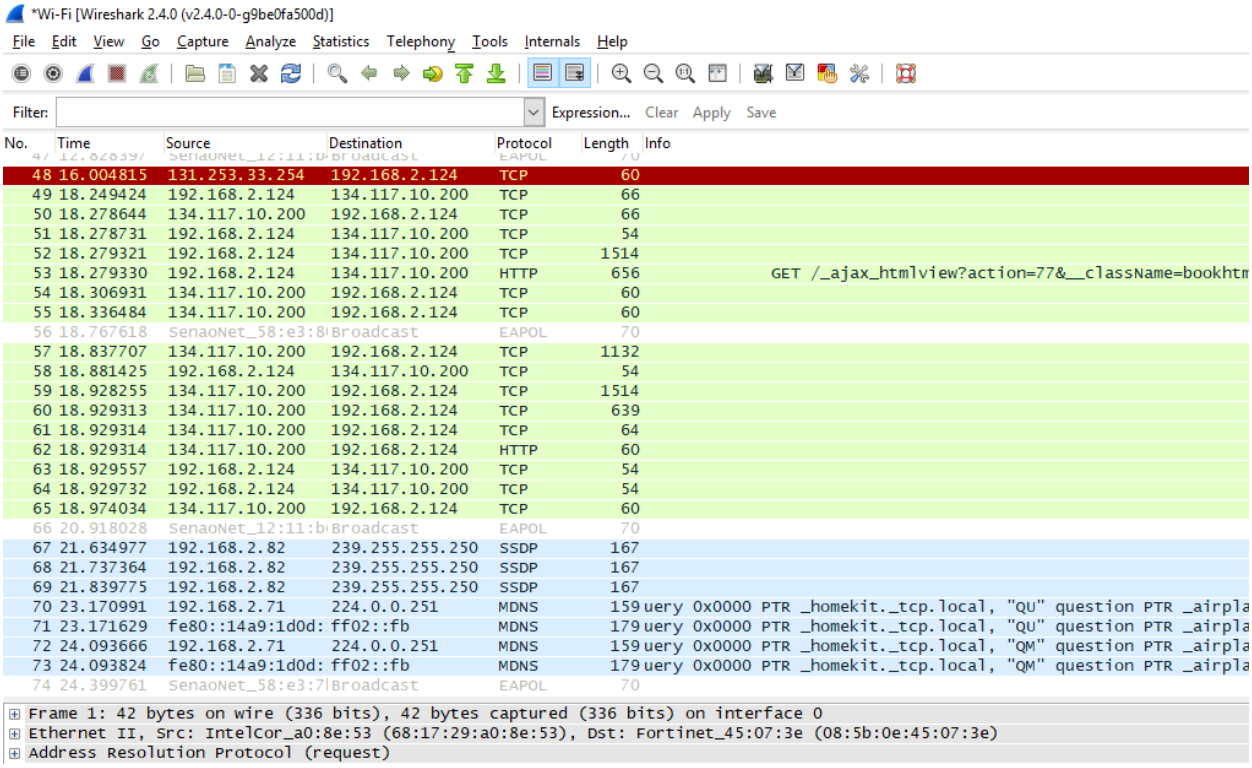

Figure 2: A successful Capture

As we can see in Figure 2 we now have a successful capture of all the incoming packets into the network.

Furthermore the capture provides us with some packet information: these are time of the packet, source IP, destination IP, protocol, port, length and info.

We can gets some hint of what the packet is doing based on the protocol information, If the protocol for the packet says TCP we know that the packet is just setting up the network communication and if the packet is entering through port 8080 we know it must be a server application.

Putting together what we now know based on the port being used and what protocol is being used we know that it's a server application trying to establish communication.

3.0: Capture Filtering

As we can see in Figure 2 we are left with many packets, most of which we don't need. To focus on the packets we do need we will use filters to filter out the packets we only want.

3.1: Filtering By IP

Just below the main menu we can see there exists a text field called Filter(Figure 3), in this text field we will type in the filter commands we want.

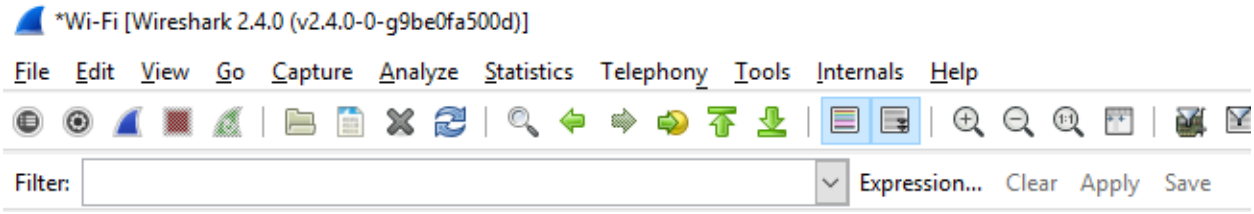

Figure 3: Wireshark Filter

Since we are interested in filtering the packets based on a particular source and destination IP, we will apply a filter that filters out the packets based on this criteria.

We start by typing *ip,* if we don't exactly remember the syntax Wireshark features an intellisense that will guide us (Figure 4) by providing a list of available filters.

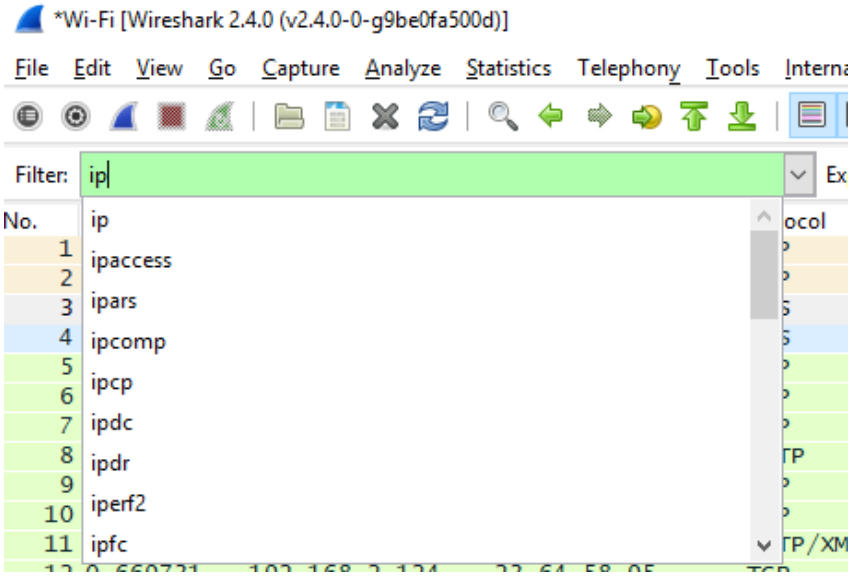

Figure 4: The Wireshark Filter Intellisense

Furthermore If the syntax is valid it will remain green, otherwise it will changed to red to indicate a syntax error.

We type in the command *(ip.src.host == 192.168.2.124) && (ip.addr == 134.117.10.200)* (Figure 5) to filter packets coming from the source IP 192.*168.2.124 to* the destination IP 134.117.10.200.

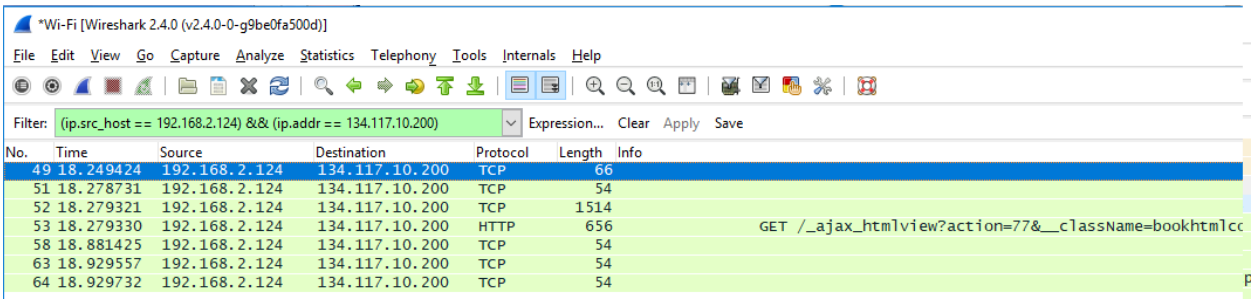

Figure 5: The Wireshark Filter Intellisense

As we can see we are now left with only the packets from the source IP address 192.168.2.124 and destination IP 134.117.10.200 (Figure 5).

3.2: Filtering By Protocol and Port Number

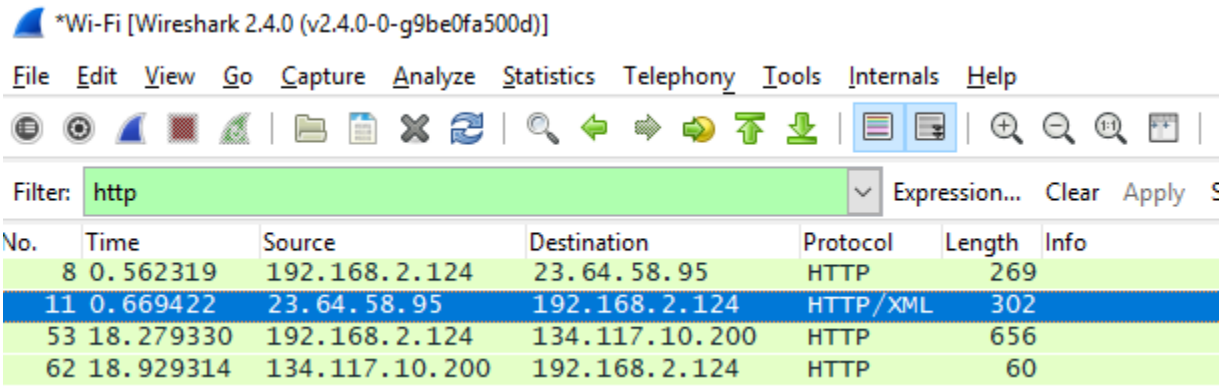

Figure 6: Filtering By Protocol

We can also filter by the protocol type by typing in the protocol type (Figure 6) and by the port number (Figure 7).

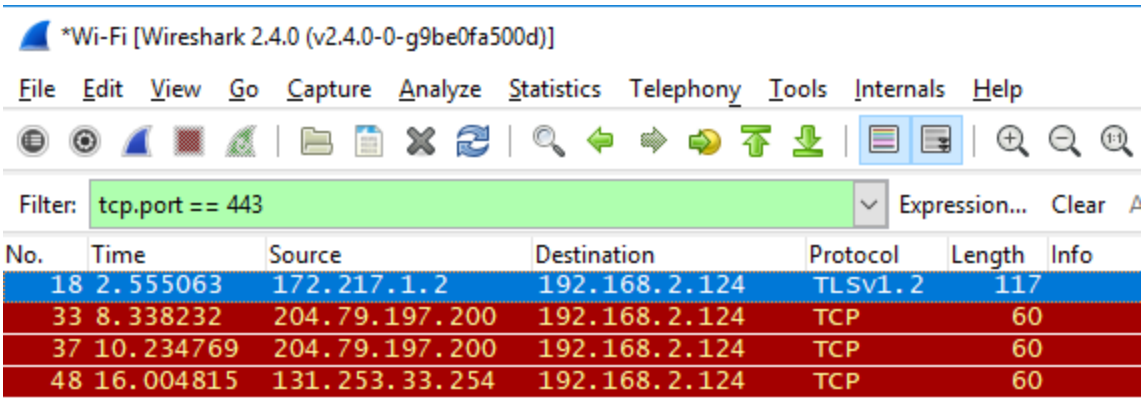

Figure 7: Filtering By Port Number

3.3: Combined Filtering

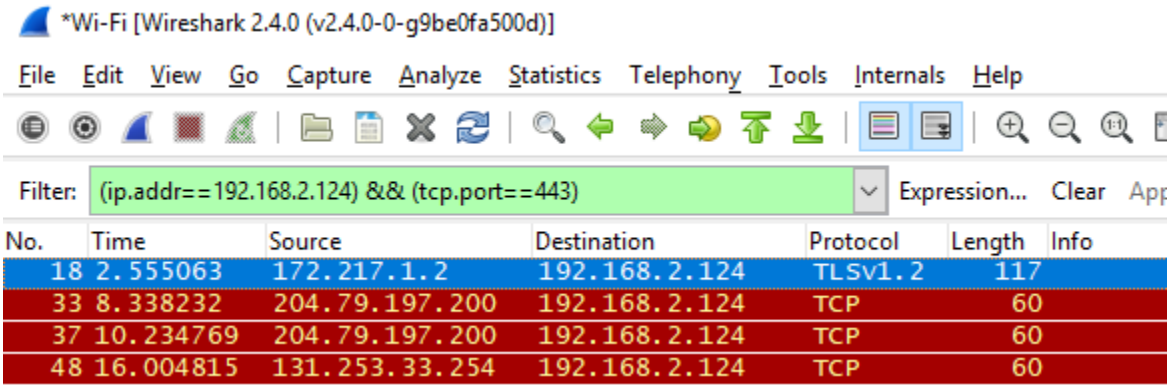

Figure 8: Filtering By Port Number And IP Address

We can also do a combined filter where we can search for all IP address using a particular port (Figure 8)

4.0: Wireshark Statistics For Anomaly Detection

Anomaly detection works is by having some form of statistic to determine the networks behaviour. By clicking on the packet and selecting *statistics* Wireshark provides a statistically audit for that packet.

We can then use this information to profile the packets behaviour and look for anomalies on an on going basis.

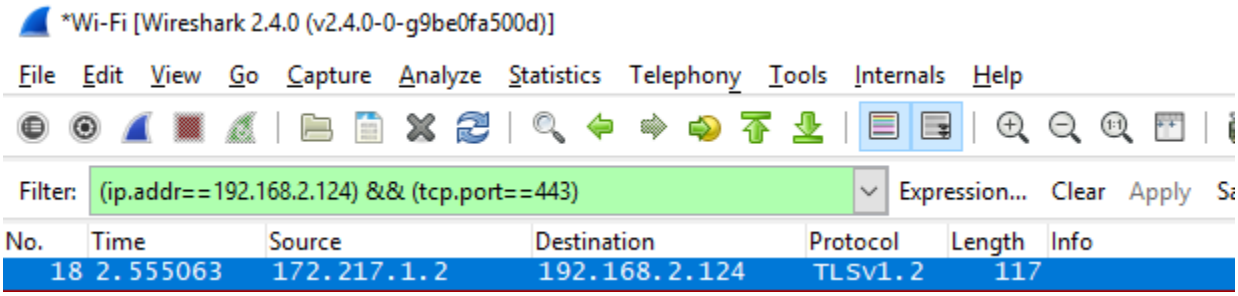

Figure 9: The Packet With A Statistic Applied To

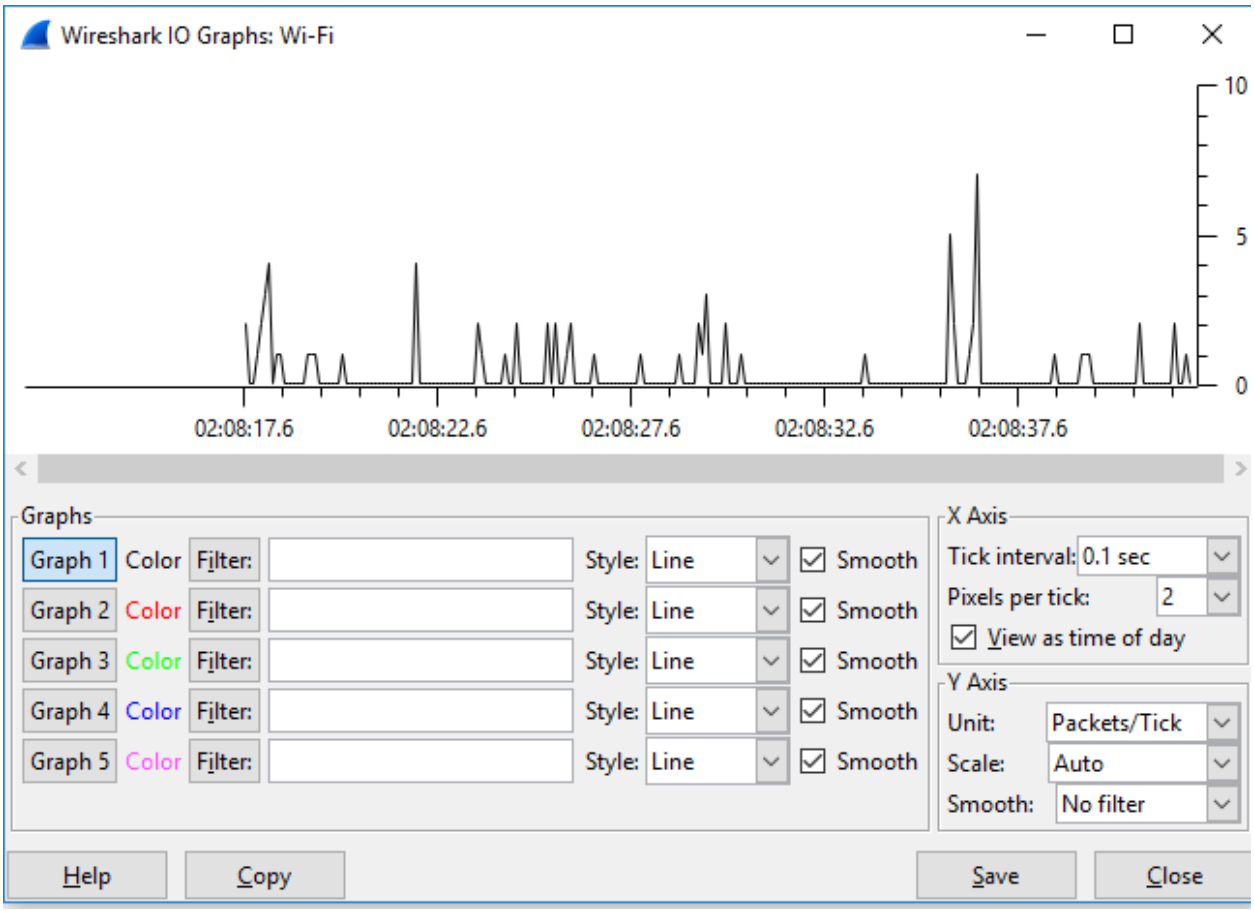

Figure 10: Displaying The Statistic Information For The Particular Packet

As can be seen in Figure 11 the x axis displays the time indicated by 0.1 sec and the number of packets, based on this statistic we can spot a pattern where a packet burst occurs every 0.5 seconds. If we observer a change in this behaviour then this that might be an indication of an intrusion.

(Next Page)

5.0: Wireshark As A Network Security Mechanism

Since we now understand how to capture packets and analyze them we can now understand how to use Wireshark for viruses and malware detection.

Lets assume that we monitor our network with Wireshark everyday and we suddenly find a new protocol called *arp* included with a packet, we deem this with some suspicion and decide to investigate.

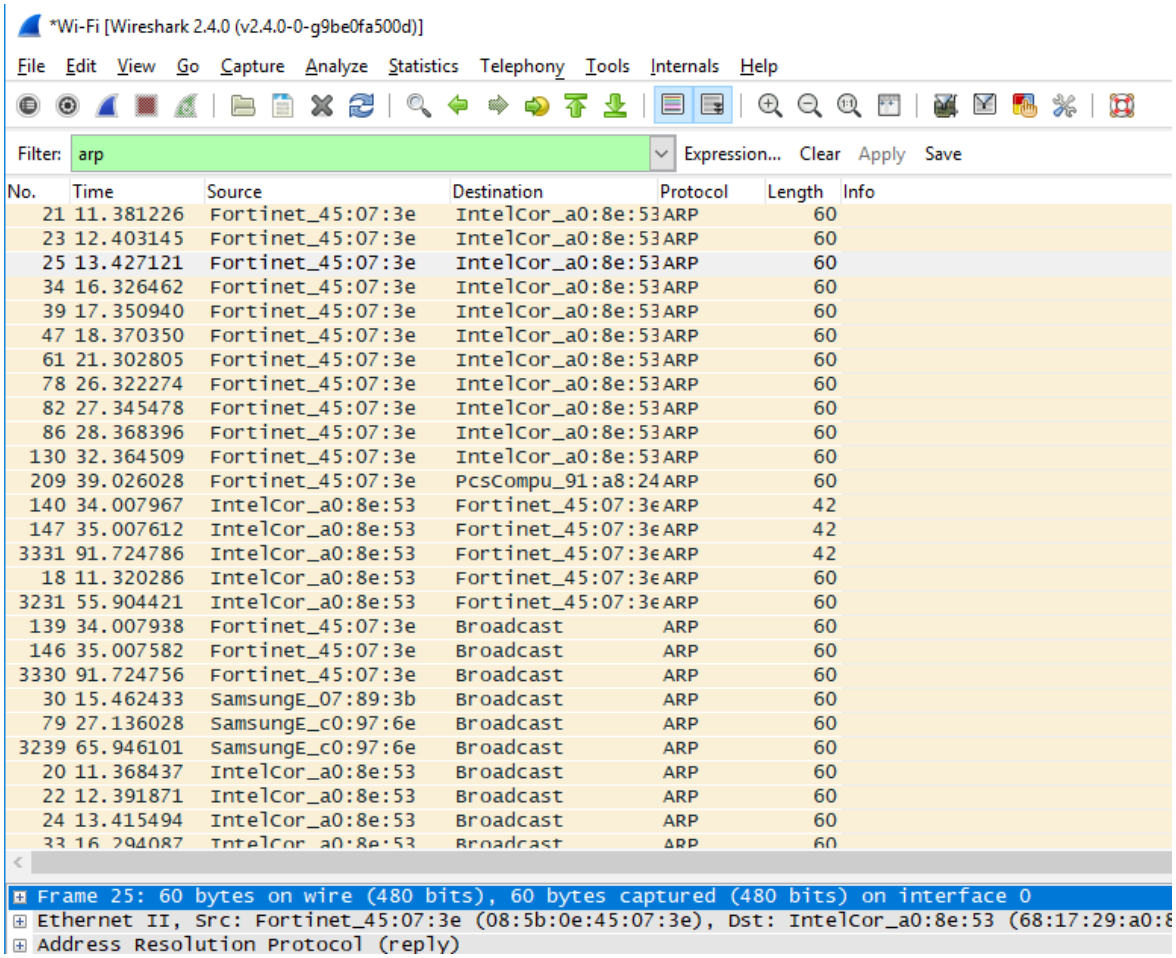

Figure 11: ARP Protocol Captured

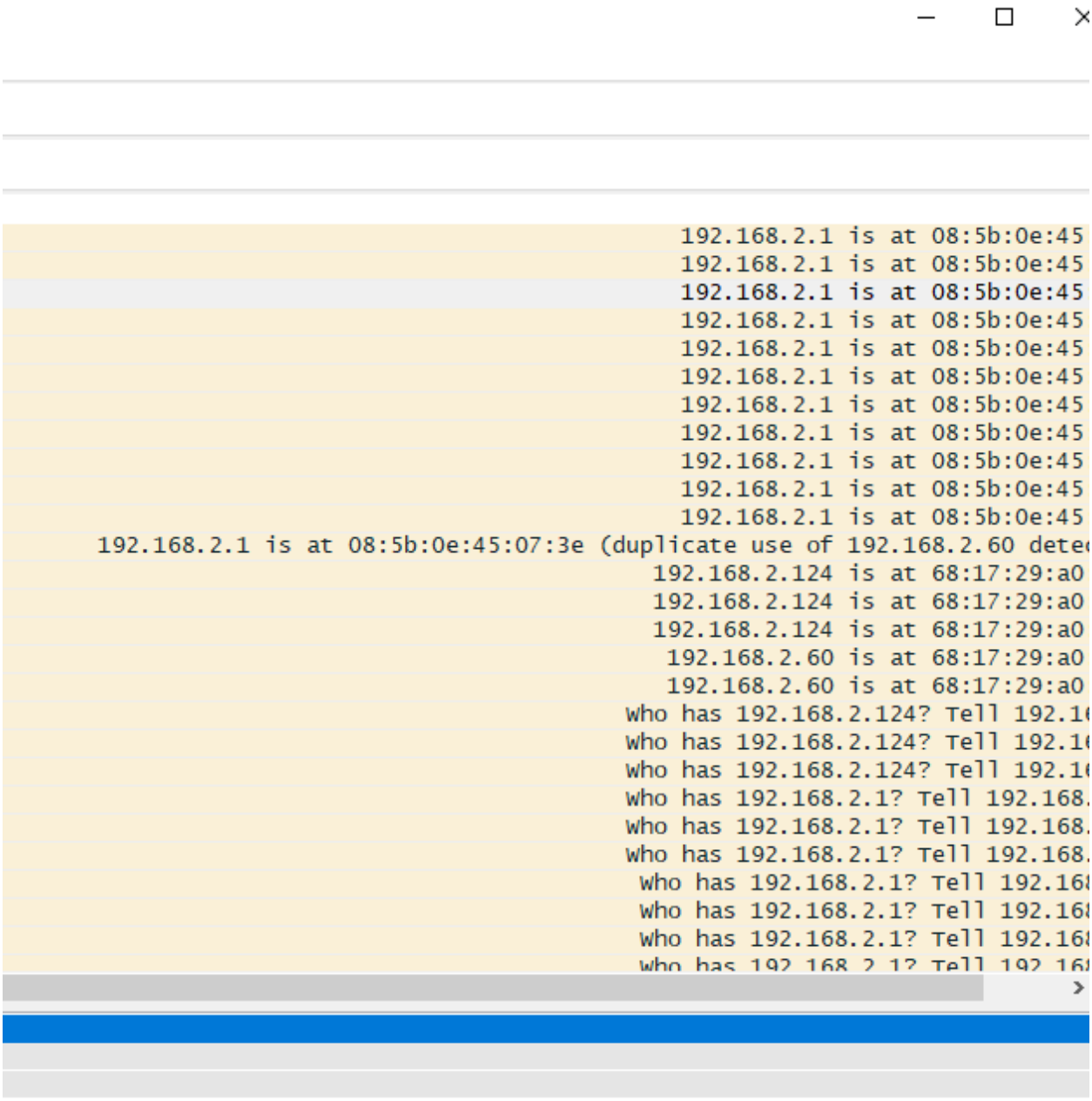

Figure 12: ARP Protocol Captured Information

After looking at the packet information we find a whole list of IPs being scanned and the source of the IP scan was originating from a source outside of our network (Figure 11).

We then realize the ARP protocol was being used by someone outside of are network to sniff the network to determine all the computer connected to the network .

As this example as demonstrated Wireshark was effectively used as a security tool to observe unusually network patterns and avoid a potential attack.