IS IA-1

Implementation of Wireshark

Presented by:

Om Topale - 16010120055

Atharva Bet - 16010120059

Deven Sharma - 16010120049

Sudhanshu Shrirao -16010120051

SOMAIYA VIDYAVIHAR UNIVERSITY

Somaiya TRUST

Introduction

- Wireshark is a network packet analyzer. A network packet analyzer presents captured packet data in as much detail as possible.
- You could think of a network packet analyzer as a measuring device for examining what's happening inside a network cable, just like an electrician uses a voltmeter for examining what's happening inside an electric cable (but at a higher level, of course).
- In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Wireshark, that has changed. Wireshark is available for free, is open source, and is one of the best packet analyzers available today.

Purpose

- Network administrators use it to troubleshoot network problems
- Network security engineers use it to examine security problems
- QA engineers use it to verify network applications
- Developers use it to debug protocol implementations
- People use it to learn network protocol internals

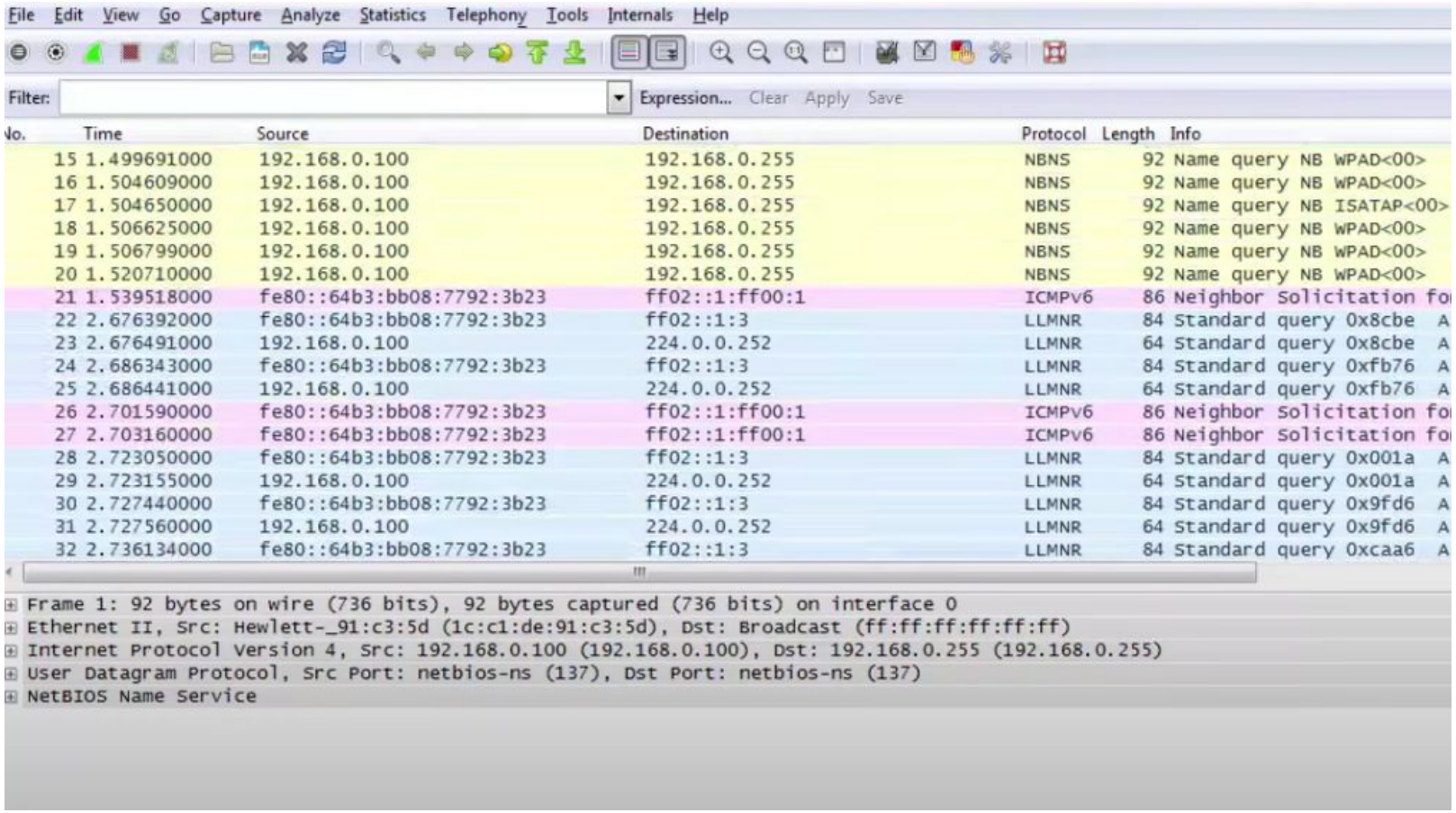
Features

- Capture live packet data from a network interface.
- Open files containing packet data captured with tcpdump/WinDump, Wireshark, and many other packet capture programs.
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- Save packet data captured.
- Export some or all packets in a number of capture file formats.
- Filter packets on many criteria.
- Search for packets on many criteria.
- Colorize packet display based on filters.
- Create various statistics.
- ...and a lot more!

Packet Sniffer

- Packet sniffer is a basic tool for observing network packet exchanges in a computer. As the name suggests, a packet sniffer captures ("sniffs") packets being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured packets.
- A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself.
- The second component of a packet sniffer is the packet analyzer, which displays the contents of all fields within a protocol message. In order to do so, the packet analyzer must "understand" the structure of all messages exchanged by protocols.

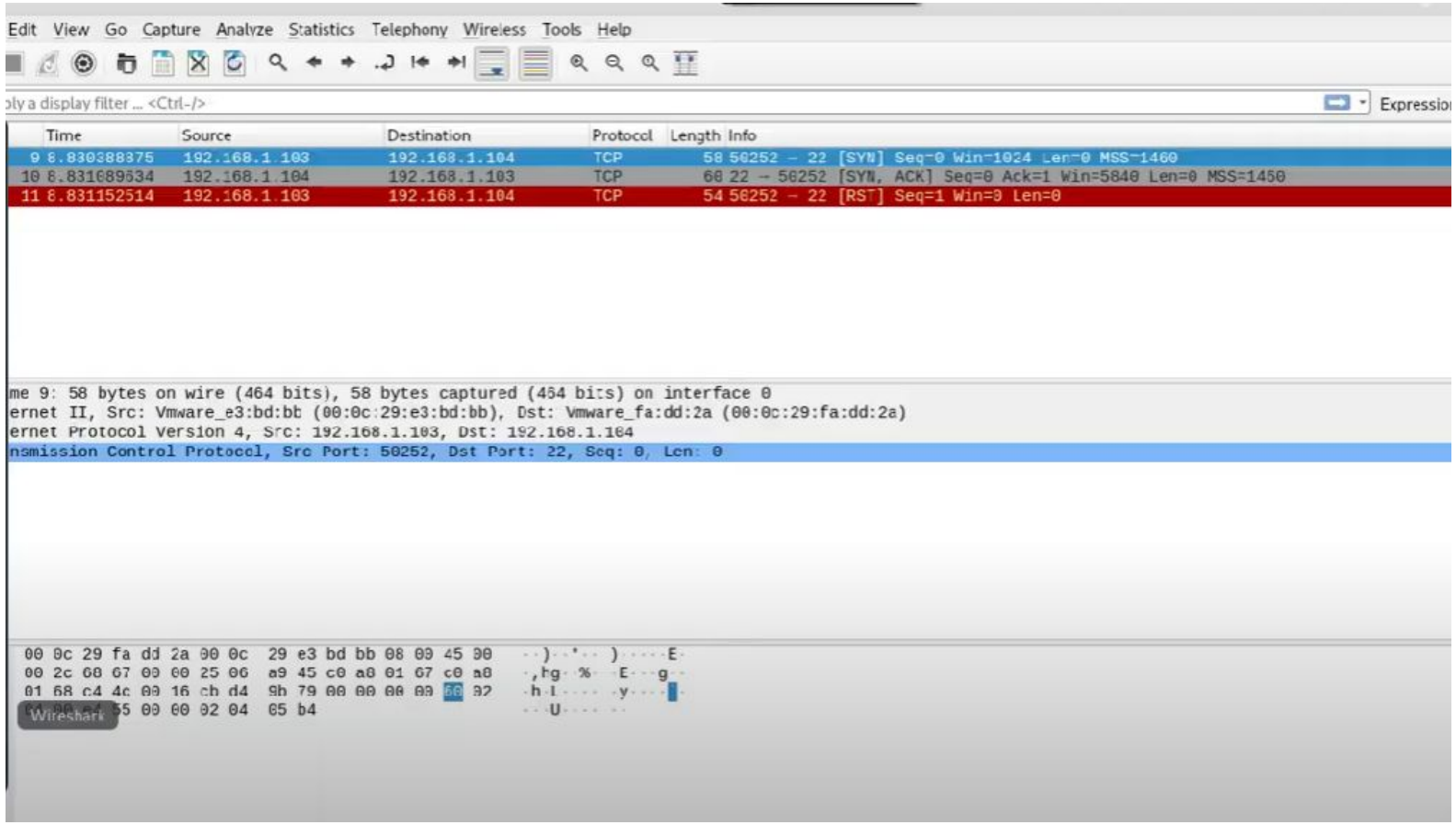
Packet Sniffer



Port Scan

- Packet sniffer is a basic tool for observing network packet exchanges in a computer. As the name suggests, a packet sniffer captures ("sniffs") packets being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured packets.
- A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself.
- The second component of a packet sniffer is the packet analyzer, which displays the contents of all fields within a protocol message. In order to do so, the packet analyzer must "understand" the structure of all messages exchanged by protocols.

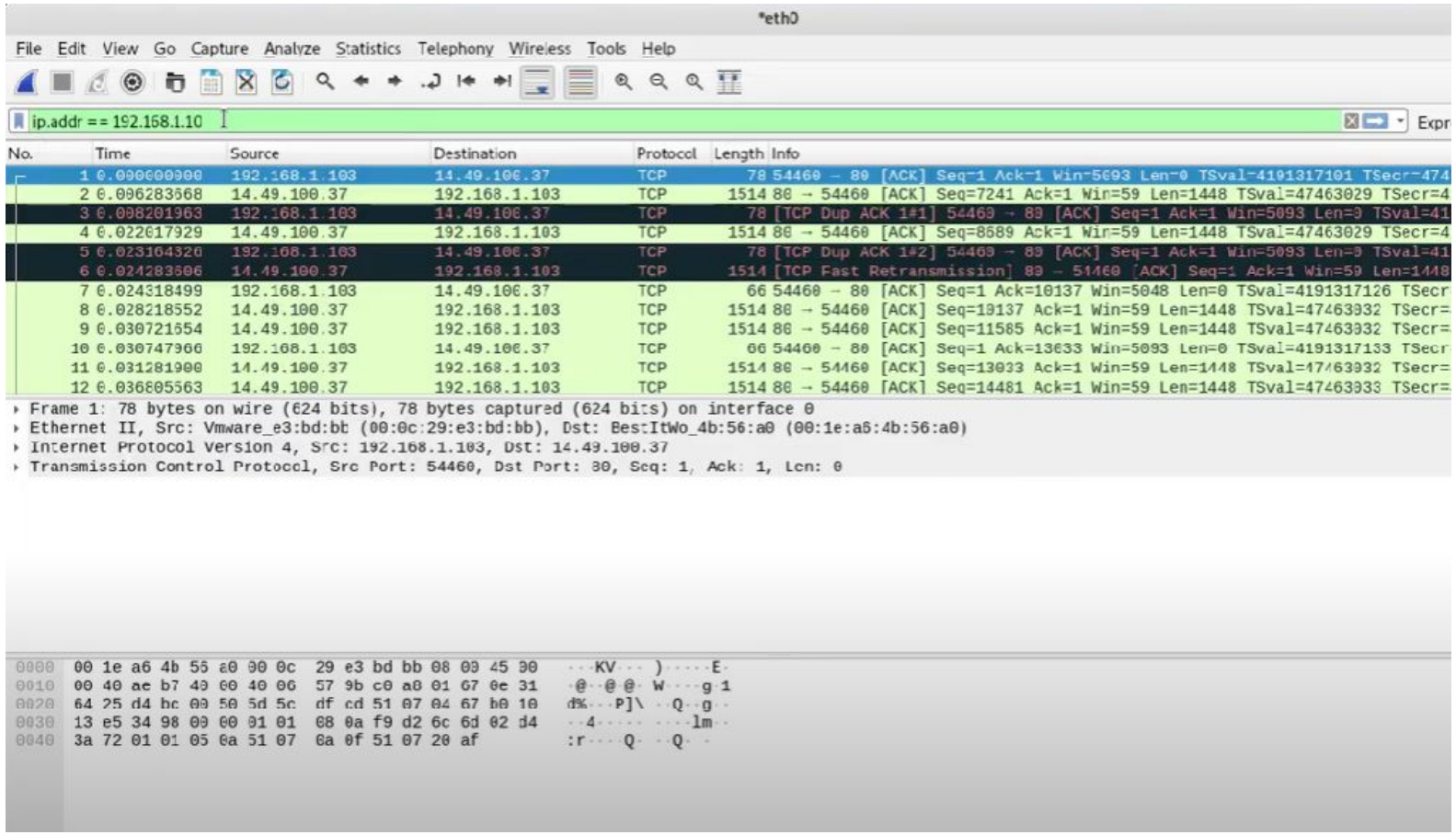
Port Scan



SOMAIYA VIDYAVIHAR UNIVERSITY

Somaiya TRUST

Port Scan



SOMAIYA VIDYAVIHAR UNIVERSITY

Somaiya TRUST

Traffic Analyser

- Traffic analyzers are tools that we use to analyze the network traffic coming in and out of a specific host computer.
- Sniffer grabs all this information, and then the sniffer's going to do one of two things. It's either going to save it into a file or it's going to make a live feed directly into the traffic analyzer. The traffic analyzer really just reads pcap data and then — here's where the term comes from — analyzes it in a way that we can look at it.

SOMAIYA VIDYAVIHAR UNIVERSITY

Somaiya TRUST

Protocol Analyser

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.254	255.255.255.255	UDP	215	32822 → 7423 Len=173
2	2.967285	192.168.0.254	255.255.255.255	UDP	215	32822 → 7423 Len=173
3	5.430885	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.32? Tell 192.168.0.1
4	5.430886	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.33? Tell 192.168.0.1
5	5.430886	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.34? Tell 192.168.0.1
6	5.430886	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.35? Tell 192.168.0.1
7	5.430887	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.36? Tell 192.168.0.1
8	5.430887	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.37? Tell 192.168.0.1
9	5.430887	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.38? Tell 192.168.0.1
10	5.430887	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.39? Tell 192.168.0.1
11	5.430887	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.40? Tell 192.168.0.1
12	5.430887	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.41? Tell 192.168.0.1

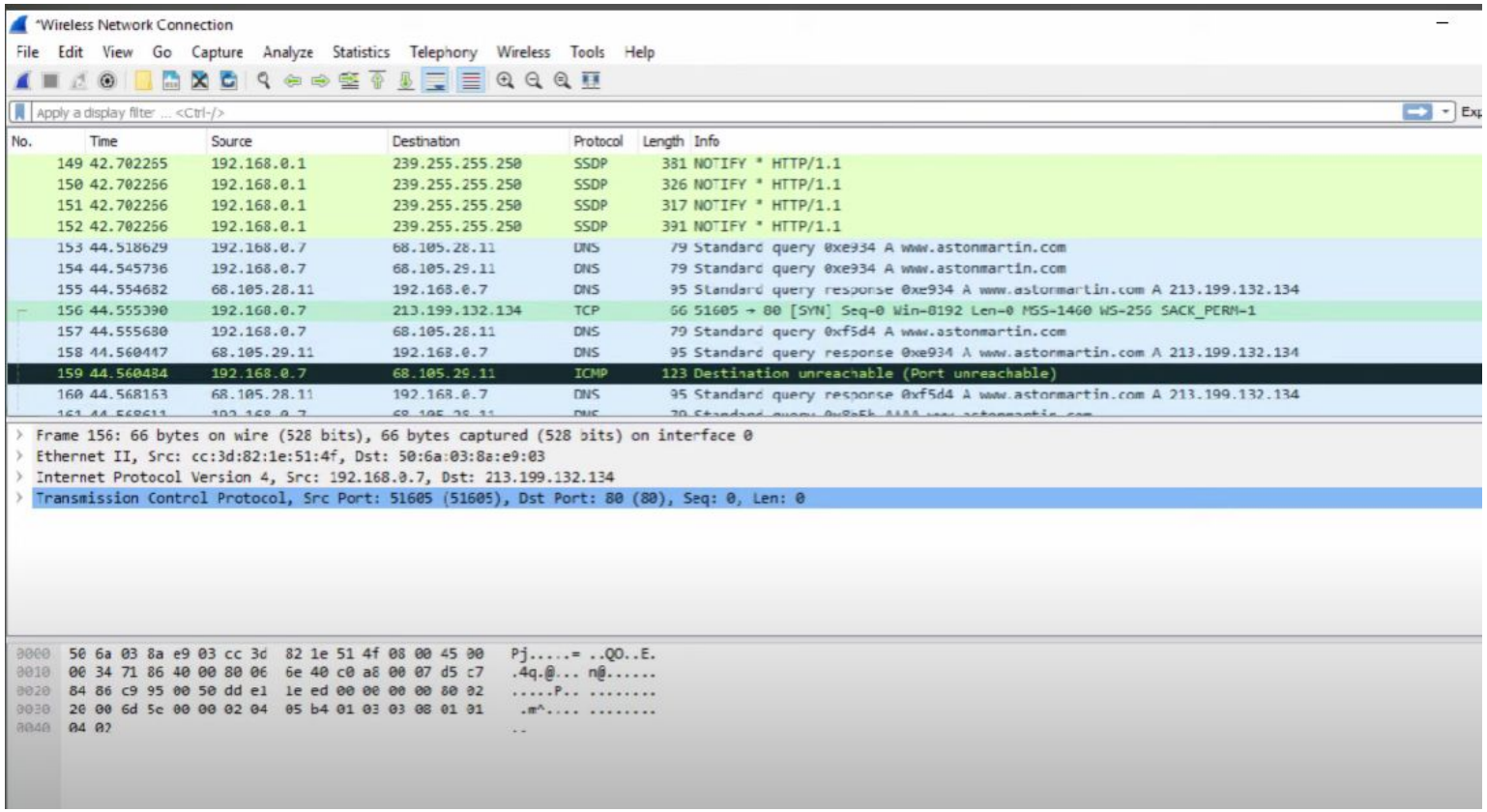
```
> Frame 1: 215 bytes on wire (1720 bits), 215 bytes captured (1720 bits) on interface 0
> Ethernet II, Src: 50:6a:03:8a:e9:04, Dst: ff:ff:ff:ff:ff:ff
> Internet Protocol Version 4, Src: 192.168.0.254, Dst: 255.255.255.255
> User Datagram Protocol, Src Port: 32822 (32822), Dst Port: 7423 (7423)
> Data (173 bytes)
```

```
0000  ff ff ff ff ff ff 50 6a  03 8a e9 04 08 00 45 00   ......Pj ......E.
0010  00 c9 00 00 40 00 40 11  78 7e c0 a8 00 fe ff ff   ....@.@. x~......
0020  ff ff 80 36 1c ff 00 b5  b4 96 4b 41 4e 4e 4f 55   ...6.... ..KANNOU
0030  25 4e 00 00 00 00 00 50  6a 03 8a e9 04 43 47 33   %N.....P j....CG3
```

SOMAIYA VIDYAVIHAR UNIVERSITY

Somaiya TRUST

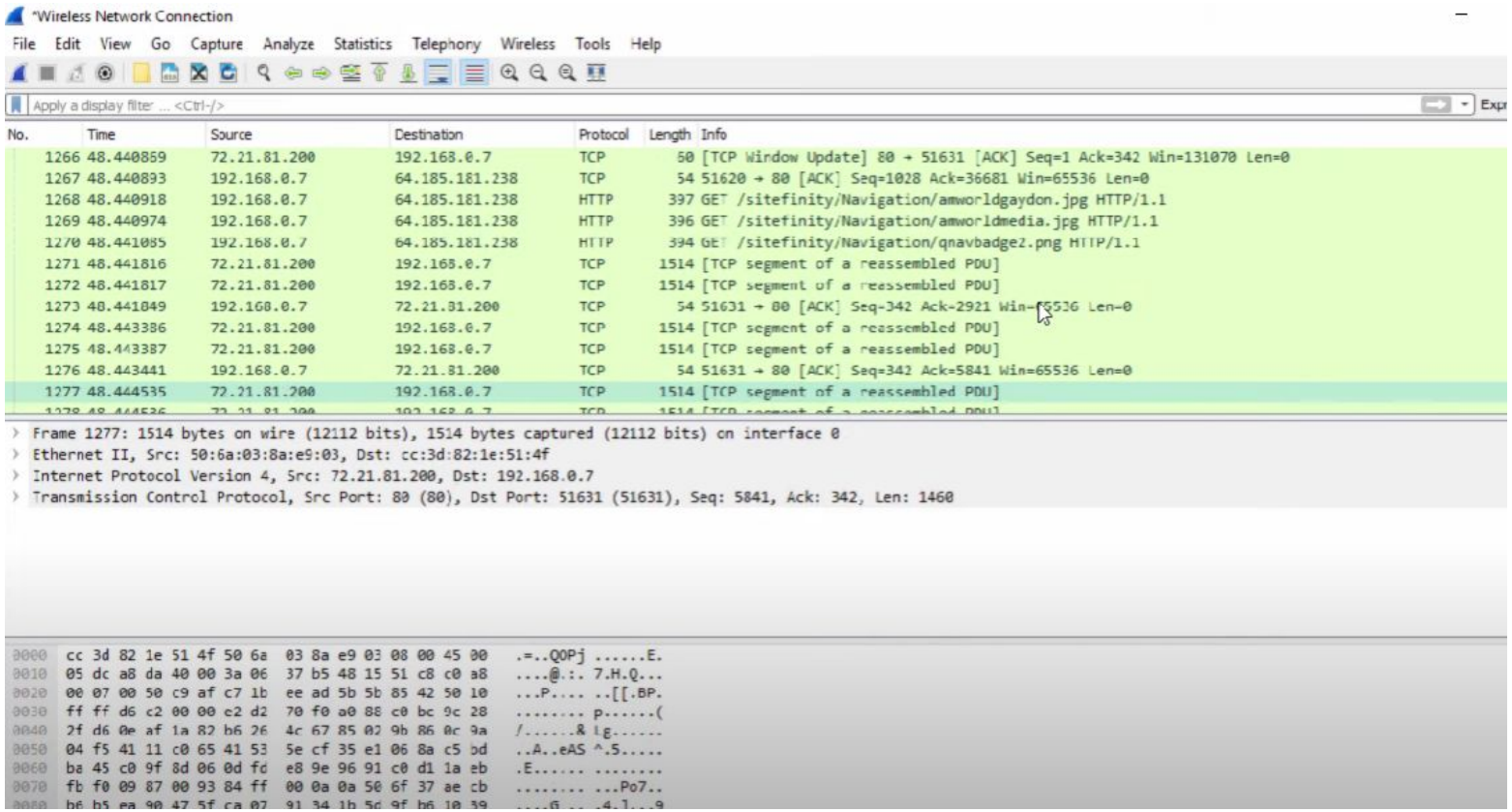
Protocol Analyser



SOMAIYA VIDYAVIHAR UNIVERSITY

Somaiya TRUST

Protocol Analyser



Conclusion

We have successfully carried out a port scan, analysed traffic on a network and sniffed packets using Wireshark.