Bachelor's thesis (TUAS)

Information Technology

Network Technology

2016

Jeewan Bhusal

NETWORK ANALYSIS WITH OPEN SOURCE PACKET ANALYZERS

- CASE: WIRESHARK

TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology | Network Technology

May 2016 | 72

Instructor: Ossi Väänänen

Jeewan Bhusal

NETWORK ANALYSIS WITH OPEN SOURCE PACKET ANALYZERS CASE: WIRESHARK

Computer Network is a growing field every day. Networking has made life easy. This world-wide computer network is accessed by more than 3 billion people in the world. The growth rate is quick and this shows the complexity of internet world. The defense research of USA gave birth to ARPANET which later created protocols to link two different computers. This creation is called TCP/IP protocols and this is how internet was born. Computers are connected to different topologies in a network and they communicate because of networks protocols. Small home, office network, local area network (LAN) of computers then become linked with Wide area networks (WAN).

In this thesis, architecture of computer networks together with analysis of packet is studied. Packet analysis is carried out with the open packet analyzer software “Wireshark”. This thesis also focuses on security challenges for a network and also presents some solutions. Wireshark can play an important role to keep the network secure and fully operational. Wireshark helps analyzing network, protocols, troubleshooting network and preventing attacks.

KEYWORDS:

Wireshark, TCP/IP, Packet analyzer, computer network, network security, protocols, packet capture, network attacks, sniffing, tapping, network analyzer

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS (OR) SYMBOLS

LAN	Local Area Network
WAN	Wide Area Network
HAN	Home Area Network
CAN	Campus Area Network
NIC	Network Interface Card
PAN	Personal Area Network
MAN	Metropolitan Area Network
OSI	Open System Interconnection
ISO	International Organization of Standardization
TCP	Transmission Control Protocol
IP	Internet Protocol
RIP	Routing Information Protocol
OSPF	Open Shortest Path First
ICMP	Internet Control Message Protocol
IPsec	Internet Protocol Security
ARP	Address Resolution Protocol
UDP	User Datagram Protocol
HTTP	Hyper Text Transfer Protocol
FTP	File Transfer Protocol
URL	Unique Resource Locator
DNS	Domain Name System
MAC	Media Access Control
SSL	Secure Sockets Layer
SSH	Secure Socket Shell
TLS	Transport Layer Security
GUI	Graphical User Interface

CAM	Content Addressable Memory
RR	Resource Records
HTTPS	HTTP Secure
MITM	Man-in-the-Middle
BPF	Berkeley Packet Filter
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
DoS	Denial-of-Service
IPv4	Internet Protocol version 4
TTL	Time to Live
IPv6	Internet Protocol version 6
SMTP	Simple Mail Transfer Protocol
POP	Post Office Protocol
CSV	Comma-Separated Values
DDoS	Distributed Denial-of-Service
SNMP	Simple Network Management Protocol
TAP	Test Access Point
NAC	Network Access Control
No.	Number

1 INTRODUCTION

In the world of information technology, networking has become an essential part of our daily lives. Computer network is the interconnection of different computers by a single technology. Protocols, hubs, cabling, switch, router, network management software all play a vital role to construct a network. Topology, protocol and architecture are the key characteristics of network. Local area networks (LAN), home-area networks(HAN), Campus-area networks(CANs) etc. are linked together in geometric arrangement or topology, with common set of rules or protocols, using network architecture such as peer-to-peer or client/server to construct a network. In fact, these networks or millions of computers connected together in global scenario is called the Internet, network of networks [1].

Computer networks always have a big risk of security problems, such as spyware injection, malware, configuration errors, and other different network attacks. However, to better understand real problems in a network and to solve them, it's important to go to the packet level. It is believed that all network problems rise from the packet level. This is where packet analysis plays a big role in computer networks. Network analysis, protocol analysis or simply sniffing are the other names of packet analysis. In general, it is defined as the process of capturing live data flow in the network and analyzing the result to see what is happening on that network. Network Interface Card (NIC) is switched to promiscuous mode to listen all traffic. This mode helps to collect raw binary data from the wire. This collected raw data is converted into readable form and this finally ends up with analysis. Packet-sniffing software or programs are used to analyze the network and also known as packet analyzers, network analyzers or packet sniffers. Some examples of such software are tcpdump [2], Omni Peek [3] and Wireshark [4]. Packet analyzers are quite useful to detect bugs, errors in a network and help efficiently to monitor the network. Wireshark is chosen for this thesis because it is user-friendly, free and considered one of the leading programs in the market.

The main purpose of thesis is to understand how computer networks operate, security issues in the network and to see them through the eye of network analyzers with some discussion in how to improve network security. Similarly, this thesis also includes how installation and initiation with Wireshark, along with its customization to get maximum benefit from this program. This thesis also explains how to capture data, how to find out network leakage and connectivity issues. Furthermore, this thesis investigates

some of the widely used protocols such as TCP/IP, HTTP and their communication strategy.

This thesis consists of 7 chapters. Chapter 1 is an introduction to the topic and explanation of this thesis's target. Chapter 2 discusses about the data network and its features, three different layers architecture of computer communication and sums up with some network connection principles. Chapter 3 defines and describes network analyzers, their uses and characteristics. It includes a brief comparison between some well-known packet analyzers. Chapter 4 points out the security issues in computer network. It discusses the importance of protecting every bit of information and explains about the few possible attacks in a network. Chapter 5 mainly focuses on Wireshark. It describes its beginning procedure, some useful rules in the program, customizing as per needs and possible benefits from built-in features such as expert info and I/O graphs. Chapter 6 is the continuity of Chapter 5 about Wireshark. However, this chapter describes where to place Wireshark in a network and some detail investigation of important protocols and their operation. This chapter ends by discussing network troubleshooting, and some practical examples of attacks captured in the network. Finally, Chapter 7 concludes this thesis.

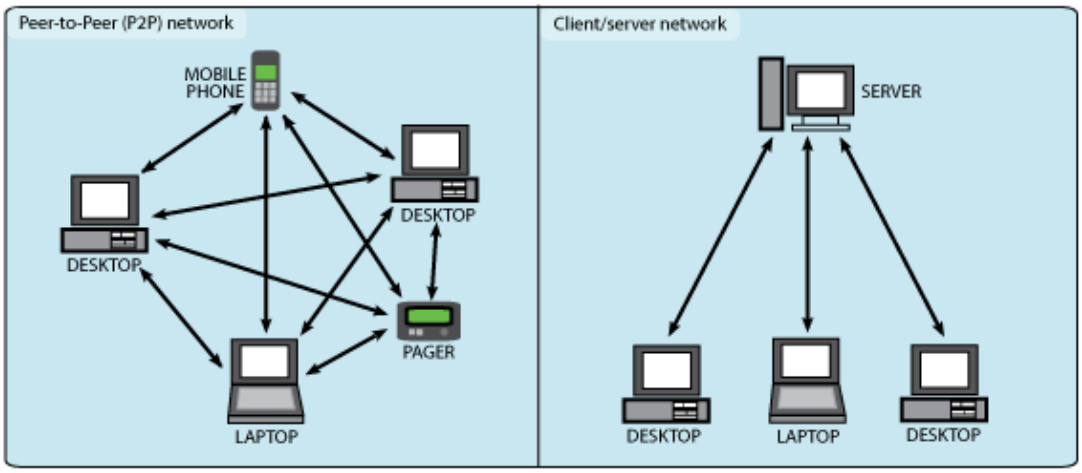

Figure 1. Example of computer network [5]

2 DATA NETWORK

Data network is also known as computer network. A set of connected computers in a network via cables or radio waves to share information or data is called data network. Computers on a network are called nodes. Computer network provides many advantages such as resource sharing, exchanging emails, internet, IP phones, video conferences and many more. Examples of computer network are personal area network (PAN), local area network (LAN), metropolitan area network (MAN), wide area network (WAN). In order to connect two computers, we depend on topologies. In general, topology is a way to connect two computers in some structure, design or arrangement. Examples of topology are Point-to-Point, bus topology, star topology, mesh topology and more. This thesis is about traffic analysis, so our main focus is to see how computers communicate rather than connect [6].

Networking is quite complex due to software, cables, physical devices, electric pulses. As a result, network communication is divided into layers. Layers are built on top of each other. These multiple layers exchange data, each layer performs a certain duty but they are independent of each other. Each layer contains a set of protocols or rules for communication. This mix of layers and protocols for communication builds network architecture. In order to clearly understand computer communication, three different network architectures are introduced in this thesis. However, a detailed explanation of all its layers is only given for the hybrid model. The whole networking process in layered network architecture is divided into small tasks. Each layer performs only one task. If one layer at the bottom starts the process, then it is passed onto another layer above it and vice versa. Generally, the task is started by topmost layer or lower layer and it is passed to next layer and continues to all other layers present in that model.

2.1 OSI Network Architecture

The OSI model (Open System Interconnection) was developed by International Organization of Standardization (ISO) and also known as the ISO-OSI model. This model can be used for any open connection. However, the OSI model is no more considered as a recommended standard and developers are not required to follow it exactly. This model has seven layers. Each layer performs a specific function. However, this model does not tell the exact protocols to be used. The OSI model is criticized because it was planned theoretically giving certain function to each layers before protocols were even

invented. The top three layers are difficult to clearly understand and hardly give precise meaning.

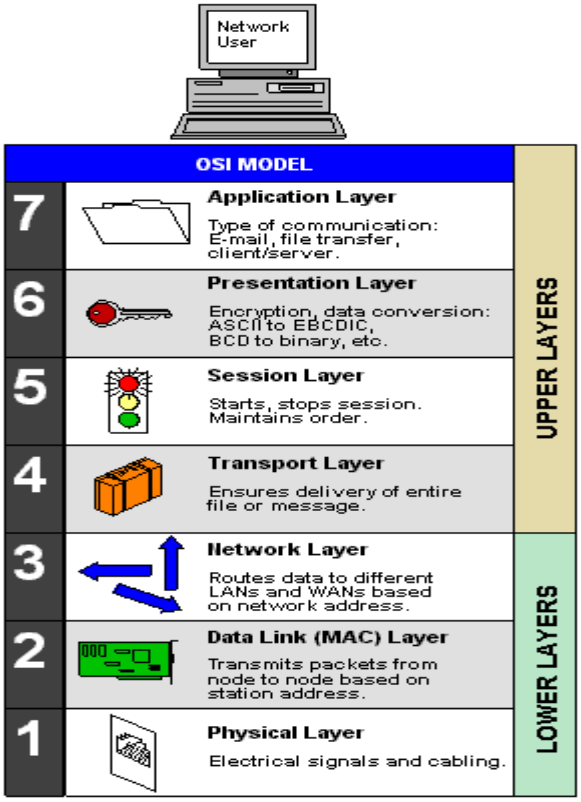

Figure 2. Layers of OSI Model [7]

2.2 TCP/IP Network Architecture

TCP/IP (Transmission Control Protocol/ Internet Protocol) is also known as Internet protocol suite. The name of this model comes from its 2nd and 3rd layer i.e. Transmission Control Protocol (TCP) from transport layer and Internet Protocol (IP) from Internet Layer. This model does not fulfill the three concepts in networking such as services, interfaces and protocols. It is also becoming harder for this model to describe new networks using new technologies. In general, it is suitable for the TCP/IP protocol stack only. It was the invention of ARPANET. This reference model was made after the invention of TCP/IP protocols.

Table 1. Layers of TCP/IP network model

Layer 4	Application layer
Layer 3	Transport layer
Layer 2	Internet layer
Layer 1	Host-to-network layer

2.3 Hybrid model

The OSI model and TCP/IP model have been criticized and also possess some problems. In today's scenario, the hybrid form of OSI and TCP/IP model is used and known as the hybrid reference model or the hybrid TCP/IP-OSI reference model. In this model, host-to-network layer of TCP/IP model is replaced with the Datalink and the physical layer of the OSI model. And all other three layers of TCP/IP are exactly used, making it a five-layered model. The addition of the physical layer in this model is responsible for bit transmission through wired or wireless sources. For instance, the Bluetooth communication cannot be described using the TCP/IP model. However, this model can also explain the wireless transmission. Application programs on different computers cannot communicate directly. In order to communicate, they use the layered communication principle or layer network model through the encoding and decoding process. For example, if an HTTP request is made by a host. Then this request is encapsulated into TCP segment, IP packet, Frame and this information travels through physical transmission media which is received by another end. The de-encapsulation process occurs in the receiving end or web server in this scenario. The information is passed from the lower layers to the next-higher layer. The five layers of this model are described below [8].

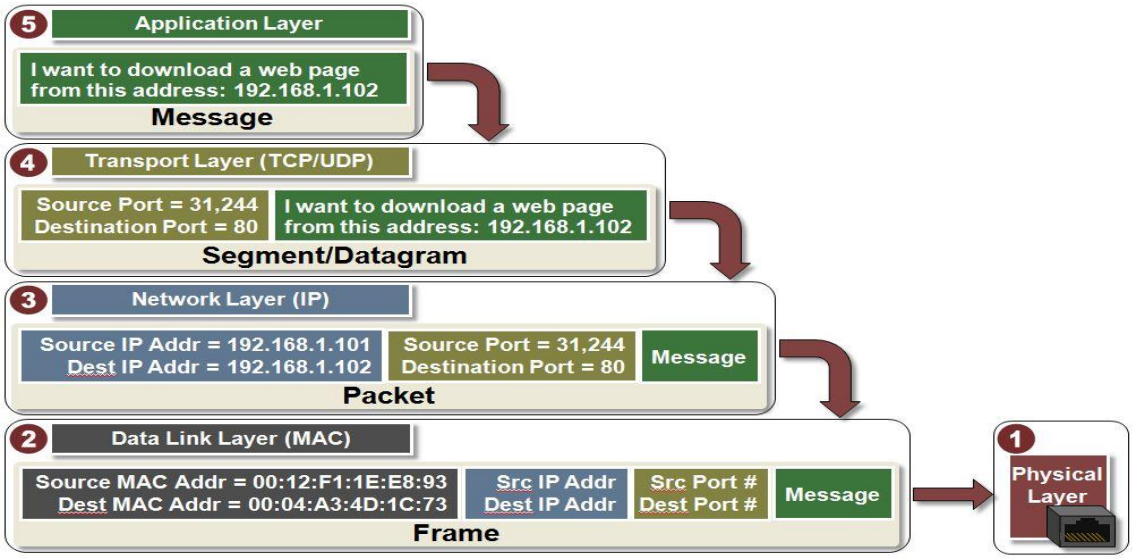

Figure 3. Hybrid network model [8]

2.3.1 Physical layer

The physical layer is the base of networking, networks and their layers. The main duty of this layer is to transmit a raw bit stream over a communication channel. Two devices can communicate because of this layer which acts as a media for information transmission. The physical layer has different roles such as establishing and breaking any connection, moving bits between devices, placing signal on the cable and many more.

2.3.2 Link layer/Data Link layer

The data link layer is the 2nd layer of hybrid model. This layer generates and transmits frames from one end to the other. At the receiving end, this layer receives data from the physical layer in the form of electrical signals, converts them into frame format and passes to the layer above it. In sending end, this layer receives IP packets from the network layer and encapsulates them into frames. Framing uses physical address or the MAC address of the host, to make it unique. Sometimes bits of information can be lost in physical transmission, therefore, this layer also performs the task of error detection and recovering lost bits. It also helps to control speed during data exchange between machines with fluctuating speed, also called flow control.

2.3.3 Network layer

Layer 3 in hybrid model is called the network layer. This layer plays a vital role in network addressing, internetworking, and managing sub-networks. The network layer delivers information in packets from source to destination, maps addresses and protocols. This process of delivering packets to the destination in different networks, subnets is called routing and routers are used to connect these networks. At the sending end, it forwards data in packets containing source and destination ports, addresses to the link layer. Similarly, at the receiving end, this layer checks the host address and forwards the packet to Transport layer.

In a network, every machine has to a unique address. This unique address is called Internet Protocol (IP) address. Currently, two versions of IP exists, i.e., IPv4, IPv6. When a host receives the IP address of its destination host, it forwards all its packets through a gateway. A gateway is simply a router which contains a routing table and data reaches the destination with the help of this table. This gateway router sends the packet to the next router which also follows the routing table to reach a destination within any subnet. Routing is a complex world in itself. This layer has routing protocols such as RIP (Routing Information Protocol), OSPF (Open Shortest Path First) and several other protocols for security and better control such as ICMP (Internet Control Message Protocol), IPsec (Internet Protocol Security), and ARP (Address Resolution Protocol).

2.3.4 Transport layer

Layer 4 of the hybrid model is called the transport layer. This layer provides peer-to-peer and end-to-end connection and exchanges data as segments and datagrams. The main task of this layer is to establish communication between application programs installed in two different computers in the network. In addition, it ensures the error-free transmission of data in a proper order marked with sequence numbers. Similarly, to uniquely process the data, every application program is marked with port numbers. In transport layer, every unit of data must contain the sending and receiving port numbers. For instance, port number 53 is used for communication between client and DNS server, port 80 is used to communicate with web server. Thus, this layer tries to provide error free transmission, flow control and also maintains the quality of service.

The transport layer has two main protocols: TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). TCP provides reliable connection. This protocol is reli-

able, connection-oriented, checks error, performs re-transmission, and maintains the order of the data sent. TCP uses three-way handshake method for connection and the actual data is sent only after this handshake succeeds between two hosts. However, UDP is an unreliable transport protocol. It is connectionless and does not maintain any order of data, but it is easy to process. UDP is really useful for streaming voice traffic, video traffic where even the loss of few packets is unnoticed. TCP and UDP communication happens through sockets, the unique end point in the network. Hosts use the combination of IP addresses and port numbers to transmit data segments in the correct network socket. The port numbers between 0 to 1023 contain some of the well-known ports and are reserved as system ports. Ports 1024 to 49151 are reserved as user ports. A user device randomly generates unique port number from user ports during TCP communication.

2.3.5 Application layer

The application layer is the top layer of this model. A user fires a query directly or indirectly in the application programs and this layer with the help of layers below it transfers encapsulated data to the remote host. However, every user application cannot be considered as application layer programs. For instance, a text-editor does not interact with the communication system, so it is not included as an application of application layer. However, the whole idea is the interaction between an application on one host with another application on another host through communication channel.

The application programs in a user device use the protocols of the application layer to communicate. For example, a web browser is an application program which uses HTTP (Hyper Text Transfer Protocol) for connecting web server. Similarly, software like FileZilla is used to upload files which use File Transfer Protocol (FTP) of application layer. The World Wide Web and email are also useful applications of the internet. The World Wide Web is the store house for files of different formats like video, images, games etc. This web store can be accessed by entering the URL (Uniform Resource Locator) or the domain names in a web browser, which are converted into IP addresses by Domain Name System (DNS) protocol of application layer to access the resource. Numeric IP addresses are confusing and hard to remember for humans. Therefore, domain names are used to address a particular resource in the network such as www.turkuamk.fi.

2.4 End-to-end principle

In computer networks, one host can communicate with another host, only if it knows the physical address of that host. ARP (Address Resolution Protocol) is the layer 2 protocol used to discover the physical address, commonly known as MAC address. A MAC (Media Access Control) address is a hardware address of network devices. When a packet arrives at layer 2 with source and destination IP addresses, the link layer needs to forward the packet in frame for which it needs source and destination MAC address. At first, ARP checks its ARP cache for the destination MAC address. ARP cache is the table which maps the IP address to the MAC address. If it is not found in the cache, the host sends the ARP request to every host connected in its local network as a broadcast. The broadcast traffic is sent with the source IP, the destination IP, the source MAC address and the destination MAC address as FF:FF:FF:FF:FF:FF, which means that the destination hardware address is unknown and this traffic is sent to every host in that network. The ARP request and reply process is shown in Figure 4 [9].

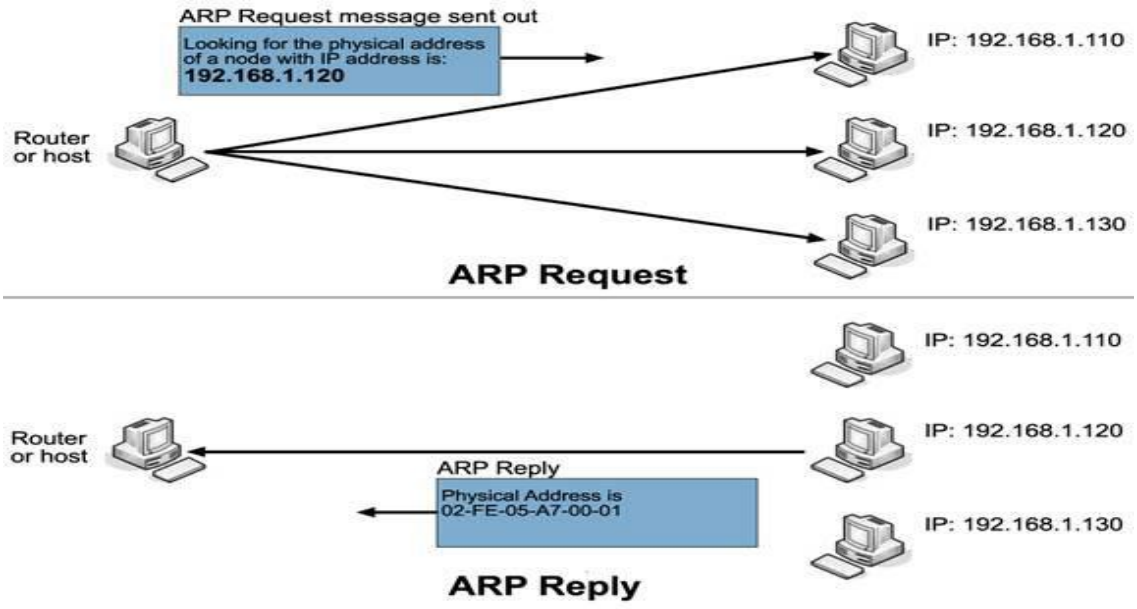

Figure 4. ARP request and ARP reply [9]

All the hosts in the network receive the ARP request. However, the host which resolves the same destination address replies to the ARP request and other host drop the broadcast frame. In the context of remote host, ARP request and reply process starts

from the default gateway or router of the sender host which continues through all the routers in between source and destination hosts. After the ARP process, hosts update their ARP table/cache which makes communication easy for the future. In the case of packet analyzer such as Wireshark, NIC (Network Interface Card) can be set to promiscuous mode which can capture traffic not destined to the host running this software. In an Ethernet network, packets would be dropped if MAC address of two ends are not exchanged. Basically, MAC address is always needed to forward packets in frame from layer 2. In order to achieve MAC address, ARP is an unforgettable protocol of networking. This is how end- to-end communication between hosts start. In Wireshark when capture starts, it begins from ARP request and reply. Hence, layered architecture of network, protocols, and end-to-end principle become an important part of this thesis. In the protocol tree of Wireshark, the highest layer protocol is shown at the bottom and lowest at the top which can be seen during packet analysis. The protocol analysis using Wireshark is explained in Chapter 6.

3 SECURITY PERSPECTIVES IN NETWORKING

This chapter discusses security issues in network from the aspect of network analyzers. The section 3.1 describes three key principles of protecting network. Section 3.2 points out different possible attacks in a network in which network analyzers can also be utilized.

3.1 AIC Triad

The concept of secure system stands on three principles of security model. If these three principles are fulfilled, any system is considered secure. However, if any one of them is compromised any system can face serious outcomes. These three crucial components of security are availability, integrity and confidentiality and known as AIC triad or CIA triangle [10].

3.1.1 Availability

The concept of availability can be described as a guarantee of easy access of information by authorized users whenever they want. In general, authorized users must be able to access information without any disturbance, all the times, and in the desired format. In order to make this happen, the system, accessibility channels, and authentication mechanisms should always be working. It is always necessary to upgrade and update system, maintain backups to protect data loss, use of firewalls and proxy servers should be considered in a system to keep data available.

The denial of service (DoS) attack and distributed denial-of-service (DDoS) attack are most common cyber-attacks against availability. In DoS attack, the intruder crashes system by sending floods of requests which eventually makes system unavailable. This results in great loss of time and money. Many websites stop their operation temporarily and recover system after negotiating great loss, which directly impacts authorized users and negotiates the availability issue. However, in DDoS attack the attacker controls many computers and uses them to flow false traffic requests which directly denies service to real users. These types of attack can be hard to stop, but maintaining updated system and being pro-active by performing hardware repairs, using security software always helps.

3.1.2 Integrity

Integrity is another important principle for maintaining a secure system. Integrity is defined as the way of protecting stored data and preventing its modification or destruction from any unauthorized users. In order to maintain integrity, data should not be easily accessible to anyone and only authorized persons or network administrators should be permitted to modify and monitor stored data. Therefore, integrity plays a great role to provide accurate data with consistency and maintain its trustworthiness.

However, there are many cyber-attacks which can force systems to negotiate integrity aspect of network security. One of the main threats to integrity is code injection. The SQL injection, cookie poisoning etc. can control and modify data. There are many viruses and worms in the internet which are intentionally designed by hackers to corrupt and leak data. In order to maintain data integrity, the physical access should only be granted to network administrators and system administrators. It can also be secured by preventing tapping, documenting administration procedures, using encryption and being prepared with recovery plans for virus attacks, server failures etc.

3.1.3 Confidentiality

Confidentiality concept in AIC triad refers to privacy. In other words, the private information of an individual should not reach to wrong hands. For instance, credit card numbers, personal information should only be in the hands intended user. This kind of personal information is highly sensitive and information on wrong hands can result in identity theft to big loss.

Network analyzers are also used by hackers to compromise confidentiality by sniffing into the network and stealing unencrypted data. There are many ways to protect confidentiality of a user. Data encryption can be used to encrypt information and protect user IDs, passwords, credit card information and other personal information. Users can also be pre-informed and trained to protect from many social engineering thefts.

3.2 Network Attacks

When we connect a computer to a network, we should be aware of the fact that we are not just using resources from the internet but we are open and prone to network attacks. Although, internet helps to make our daily life easy. But the risk of digital theft is always high and the attackers, sniffers, hackers can always compromise our three basic principle of network connectivity i.e. confidentiality, integrity, and availability.

Network analyzers are also used in many network attacks. Many network applications and protocols send packets in clear text. Therefore, a network sniffer can easily steal the sensitive information, such as user names, passwords, credit card information from any database. Although, encryption is helping to make these important information secure. However, some applications still deal in plain text which can easily cause huge loss of sensitive information by simply capturing packets in local area network. And there are some packet sniffers which are built for cracking encrypted passwords, decrypting hash values and breaking many more strong security features. Some common network attacks which utilize packet sniffer for the attacking process are described below [11].

3.2.1 ARP spoofing

ARP spoofing is the process of poisoning the Address Resolution Protocol (ARP) and also known as ARP cache poisoning or ARP poison routing. Address Resolution Protocol provides mapping between layer 3 IP addresses and layer 2 MAC addresses. When a host sends IP packets to another host, it needs to know the MAC address of its next node. If the host and receiver are in the same subnet, then next node is the destination host. If not, gateway or router becomes the next node. The IP-to-MAC mapping is stored in ARP cache. At first when two devices have to communicate, the MAC address of destination is searched in ARP table. However, if the physical address of destination is not found there, then ARP sends broadcast traffic to every host in the local network. The ARP responds with ARP reply. ARP request is a broadcast traffic and ARP reply is unicast traffic. During ARP request the IP address of next node is sent, and the machine matching this IP address should respond with ARP reply. The ARP reply contains MAC address of that host. After this process, the IP-to-MAC mapping is stored in ARP table for future.

ARP is considered a stateless protocol, which means machine simply update their table if they receive ARP reply and it does not matter if they have not sent any ARP request before. During ARP spoofing, attacker sends unreliable ARP request and reply packets to the victim. In this attack, attacker updates victim's ARP cache table with its own MAC address and convinces the host to send packets to attacker machine, which now sits as a destination node. This is how ARP cache is poisoned.

In switched network, a packet analyzer can be utilized during ARP spoofing attack. Some sniffers such as Cain & Abel, Ufasoft Sniff have built-in features to perform this

kind of poisoning. Network sniffers can also be tapped in a certain part of local area network, to gather required information of the host before attacking them. Sniffers can collect IP and MAC address of the host, which supports the attacker to plan the attack. It is believed that ARP spoofing opens door for man-in-the-middle attacks, DOS attacks, and session hijacking, which is strong enough to steal any sort of information from the network. The figure below shows the ARP poisoning attack and how network traffic is redirected because of this attack.

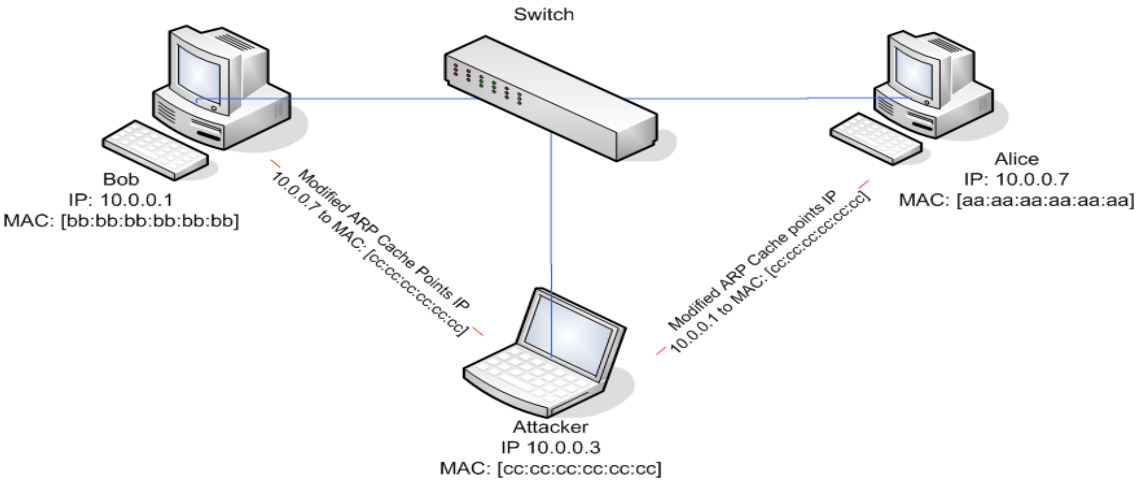

Figure 5. ARP Spoofing [12]

3.2.2 PORT Flooding

PORT flooding is also known as MAC flooding. In computer networks, switches map various MAC addresses to the physical ports on the switch which are stored in CAM (Content Addressable Memory) table. Switches have limited memory, therefore they can't store many MAC addresses. The benefit of CAM table is to send data only to the port for the destined computer. In this attack, an attacker connected to a switch port floods the switch interface by sending frames with large number of fake MAC addresses. As a result, the switch reaches a state where it cannot store any more MAC address. This is because of its limited memory, which results CAM table overflow. This overflow pushes switch into fail open mode, which makes switch behave like a hub. Thus, switch does not behave normally and starts sending traffic as a broadcast to all the ports, instead of sending to a correct port.

PORT flooding can be performed from some sniffer tools. Macof is a Linux tool which can easily perform MAC address flooding. It can send random MAC addresses and IP address to the switch, which is good enough to fill switch's CAM table with fake MAC address. Yersinia is another popular tool for performing PORT flooding. Similarly, network sniffers play an important role to attackers after PORT flooding attack becomes successful. This is because after PORT flooding, the traffic is broadcasted to all the ports. As a result, an attacker can capture all the traffic in that network and can easily steal sensitive information from other machines. Some protocols such as HTTP, TELNET, POP, SMTP, FTP etc. are vulnerable to network sniffers. Hence, an attacker can steal unencrypted passwords, e-mail, and instant messages from the network [13].

3.2.3 DNS spoofing

The Domain Name System (DNS) protocol is one of the hugely important protocols in the internet. This application layer protocol is defined as a zonal, hierarchical collection of name servers which resolves domain names to IP addresses. IP addresses are hard to be remembered by humans. In order to make it easier to remember and use, domain names are introduced such as www.turkuamk.fi. However, computers only understand IP addresses like 80.86.90.220. Therefore, Domain Name System protocol was introduced to map domain names and IP addresses.

A DNS client just needs to enter domain names into web browsers or email services to access a particular website or other web resources, and all other task is done by DNS server to fetch IP address for that particular query. At first, to resolve this query search starts from looking at the local cache of client's computer. A DNS server stores database of entries as resource records (RR) of IP address to DNS name mappings. If the website was not visited before, it cannot be found in the local cache of user's database. Now, this DNS query is send as a recursive query to nearest DNS server. This DNS server also checks its cache, but if the match is not found then this query is forwarded to root DNS server. DNS servers are separated into .com DNS server, .org DNS server etc. Therefore, if this root DNS server cannot resolve the query, it replies the local DNS server to send iterative query to other zonal name servers like .com, .org and eventually the domain name is resolved through this recursive and iterative queries. Finally, local DNS server receives the reply, sends it to the client and the client receives connection to the IP address. The local cache would easily solve this query in the future, which maintains this mapping in records.

There are various ways of conducting DNS spoofing attacks. Some of them are DNS cache poisoning, DNS ID spoofing, Birthday attack. The first two attacks are widely used to misuse DNS servers. In DNS cache poisoning, attacker sits in between the host and DNS server. The attacker captures the traffic, plans and performs this attack. DNS protocol is an UDP based, which makes it unreliable. An attacker sends unknown domain query to the local DNS server, redirects traffic from local DNS server to the DNS server higher in the hierarchy as an iterative query. After that, an attacker becomes successful to capture the superior server and can easily send response to the local DNS servers. This happens because DNS server replies to the attacker without authentication, an attacker zone transfer of a query between DNS servers, poisons the cache of the superior DNS server. Thus, an attacker can direct any user to a fake websites, or login pages. In this kind of attack, the attackers create their own fake pages for banking websites, social-networking websites, online shopping etc. In the case of a user, he/she receives connection to a fake page and if they enter their personal details such as passwords, usernames they are saved in attacker's database. As a result, attacker can easily misuse their identity, property etc. The cache poisoning remains, until the cache is updated, or deleted.

Each query send to the DNS servers always contain randomly generated unique ID numbers. In DNS ID spoofing attack, the attacker can steal this ID number through ARP spoofing attack, and sends DNS response back to the user pretending to be a DNS server. It acts as a DNS server to the user and performs many different kinds of social engineering attacks like phishing to steal sensitive information from the user. The attacker can easily redirect traffic from the user's machine to the fake webpages and so on. The packet analyzers play an important role to assist the attacker for performing DNS spoofing attacks. Network analyzers help the attacker to view traffic flow in between user and DNS server, guessing query ID numbers. The attacker mainly sits in the same network, studies traffic flow through tapping the network and quickly replies to any query to fully control the conversation between the user and DNS servers. Similarly, there are some packet sniffers which have built-in features to perform DNS spoofing attacks. Some of them are dsniff suite for Linux, Ettercap, Cain & Abel. The DNS cache poisoning attack using ID spoofing method is shown in Figure 6.

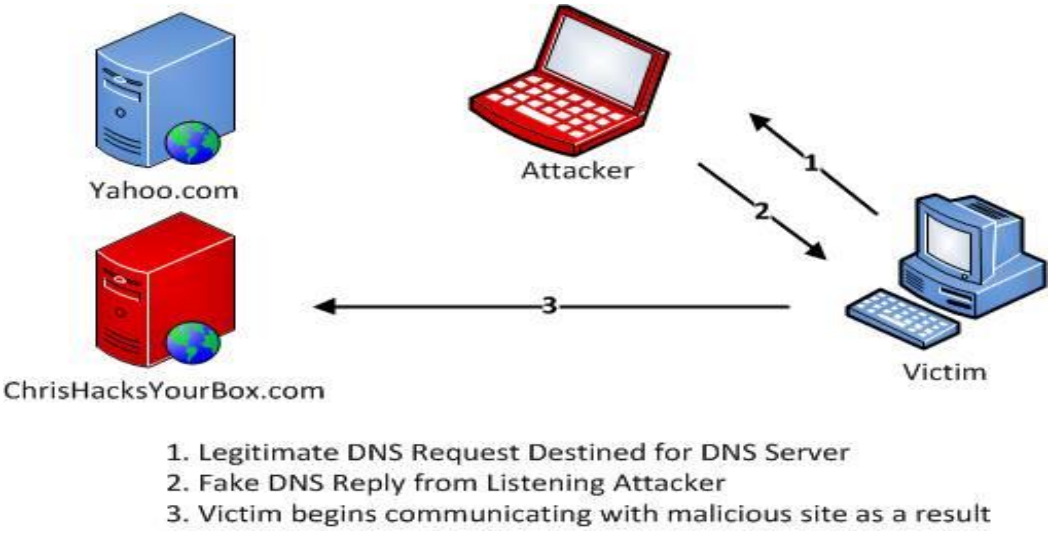

Figure 6. DNS cache poisoning using ID spoofing method [14]

3.2.4 Session hijacking

In this attack, the attacker tries to misuse existing connection or session between two network devices. This attack is also known as cookie hijacking, HTTP cookie theft. When a device has to access web pages or web resources, it needs to connect through HTTP protocol. The Hypertext Transfer Protocol (HTTP) in the web server and web applications or browsers in user side establish a session, when user receives connection to the server. However, HTTP is not a secure protocol. It communicates in plain text. Therefore, the sensitive information of the user such as online banking, shopping, and login process are handled in a secure way with session cookies. Cookies are stored and exchanged by web browser and web server, which tracks the user's activity. However, session cookies are temporary and usually deleted after the web browser is closed. These secured session cookies are transmitted via encrypted connection such as HTTPS, SSL. The browser and web server store the important information of the user in the form of secure cookies, with the help of unique identifier key called session ID. However, the encryption takes place mostly during authentication. After authentication by HTTPS, the server and web browser communicate mostly in HTTP. Therefore, the attacker can see the user's traffic easily and can steal any sensitive information.

There are four different ways of performing this attack, in which packet sniffers play a big role. The attacker uses network analyzers to understand traffic flow, steal session key and cookies. Once the session key is obtained by the attacker, it can be used to exploit webmail, online banking by simply modifying the cookie request to the server. The main idea behind this attack is that, the communication between browser and the server is encrypted only during authentication but not the whole session is secure. There is some software available which can easily perform session hijacking. For instance, Hamster and Ferret is a tool used by Backtrack (Linux distribution) to perform session hijacking. Another great example is Firesheep, which is an extension developed for Mozilla Firefox browsers. This extension can easily access the unencrypted session after the login process. DroidSheep, CookieCadger are some other examples of tools and apps developed to perform session hijacking network attack.

3.2.5 Man-in-the-Middle attack

Man-in-the-Middle (MITM) attack is one of the strong hacking techniques, and whose presence is hard to identify in the computer network. In general, the attacker tries to control traffic, modifies it and plays the role of communication controller. In this attack, the attacker sits in between two hosts such host A and hosts B, eavesdrops the traffic, edits messages and forwards them in real time.

For instance, if host A sends email to host B. The attacker sits in the middle of host A and B, and can easily encapsulate and de-encapsulate their conversation. Thus, the attacker can collect useful information. This kind of attack redirects traffic to a different location and acts as a legitimate host, proxy server. The HTTP connection between a client and a web server is considered unsecure because HTTP is a stateless protocol and communicates in plain text. However, protocols like Secure Socket Layer (SSL), HTTPS (HTTP Secure) layer up the HTTP protocol which provides security during authentication. But, it does not stop here. The hackers can easily steal public key of web server and client‘s communication through eavesdropping their traffic. Eventually, the attacker can create fake signature certificate and the client accepts this signature, believing it as a secure communication channel between it and the server. Hence, the attacker can capture, decrypt, and edit any message between the client and the server. The attacker can steal banking credentials, and misuse the client’s personal information, usernames, passwords, email. This attack becomes hard to detect in the network because the client feels normal communication, but the attacker controls the traffic all the time by sitting in the middle. This type of attack is mostly seen in public Wi-Fi

networks. Spoofing attacks and some forms of session hijacking such as side jacking, evil Twin, sniffing are considered different forms of man-in-the-middle attack. Packet analyzers play an important role in this attack because this attack is also known as eavesdropping attack. Eavesdropping also means listening to the traffic, this can be easily performed through packet sniffers. In Figure 7, the MITM attack is shown.

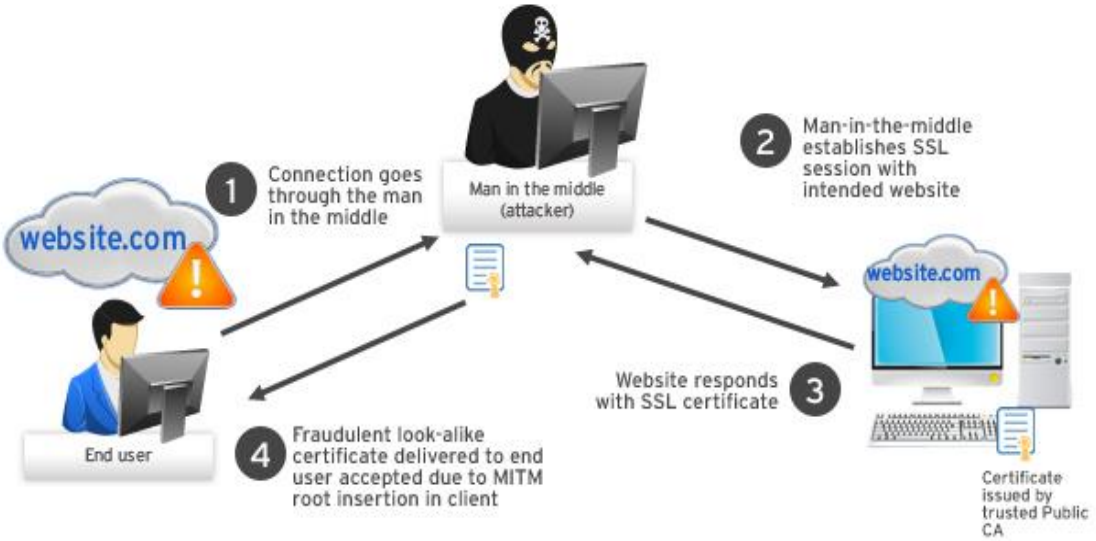

Figure 7. Man-in-the-middle attack [15]

4 NETWORK ANALYZERS

4.1 Meaning and Features

Network analyzers are also known as packet sniffer, protocol analyzers or packet analyzers. Network analyzers can be defined as computer programs or sometimes a hardware device which can listen to all the traffic flowing inside that network. In computer networks, information flows as raw binary data. However, these network analyzers can convert these raw binary data into human-readable format which helps to analyze the network. The legal use of network analyzer is to manage, troubleshoot and maintain network security by network administrators. However, network analyzers are used illegally too. Generally, the illegal use can be by a hacker who wants to gain unauthorized access and gather sensitive information and data from that network. Network analyzers can be tapped into many parts of the network, without the knowledge of IT administrator [16].

In Ethernet networks, Ethernet adapters are built with the feature called “filter”, which ignores any traffic not meant for it. However, network sniffing program puts the adapter into “promiscuous mode” and thus network adapters accept all frames even if the MAC address doesn’t match to its own. Hardware, capture filter, buffers, decoder etc. are the components of network sniffers. The network sniffing procedure is described below:

- Collecting - It is the first task of network analyzers. In general, analyzers put the network interface card (NIC) into promiscuous mode. Thus, the NIC of that computer can listen to all the traffic in its network segment and captures all the raw binary data.
- Converting - This process is carried out by decoder component of packet sniffers. In this second step, captured binary data from process one is converted into human readable form.
- Analyzing - This is the last step of sniffing process. It is the step to perform protocol analysis. The protocols used in the network traffic can be viewed from the information gathered from second process. All the packets can be analyzed and explained from the viewpoint of protocols.

4.2 Common Uses

Network analyzers can be used for both legal and illegal tasks. If used legally, it can provide lots of advantages to the network. Network administrators use network analyzers to give fruitful benefits to the network. Network administrators and people who work to secure network sees this tool as an efficient medium to protect and troubleshoot network problems. These tools are really useful to find out network latency or slowness in the network. It can always help to study the effectiveness of firewalls, access control lists, and protocol functionality. It can even help to gather network statistics. It helps to clearly see network speed, leakage by converting these statistics into graphs and reports. Therefore, network analyzers are really useful to smoothly run networks if used in a good manner.

On the other hand, there are people who use these tools to perform malicious activities which is against the law. Black hat hackers, crackers use these programs to eavesdrop in other's network and steal sensitive information. Networks are exploited to obtain private information of users such as bank details, credit card numbers, passwords, usernames etc. Anyone who performs spying and illegal activities is punished by the law. Nowadays cryptographic protocols are used to encrypt and protect the network. Protocols such as SSL, SSH, TLS etc. are used for end-to-end security. But network can still be attacked through different attacks such as MITM attacks, brute force attack etc.

4.3 Famous Network Analyzers

There are lots of network analyzer programs available in the market. However, I have included some of the well-known widely used programs below [17]:

4.3.1 Wireshark

Wireshark is a famous network packet analyzer. It is free, open source which is used for protocol analysis, monitoring and troubleshooting network. The original name of Wireshark is Ethereal and written in C, C++ programming language. And the current stable release is 2.0.2, released on February 26, 2016. Wireshark can capture live data flowing in network interface. Some of the well-known features are capturing and filtering live traffic, coloring packets on the basis of protocols, creating I/0 graphs and other statistics, and it can also export packets data in different file formats. The graphical

user interface (GUI) is easy to use and makes packet analysis easier, which is another reason for the popularity of this sniffer. It can also be used in command-line interface as Tshark. Another reason for choosing this program for protocol analysis is due to its support for more than thousand protocols. The figure below shows the graphical user interface of Wireshark.

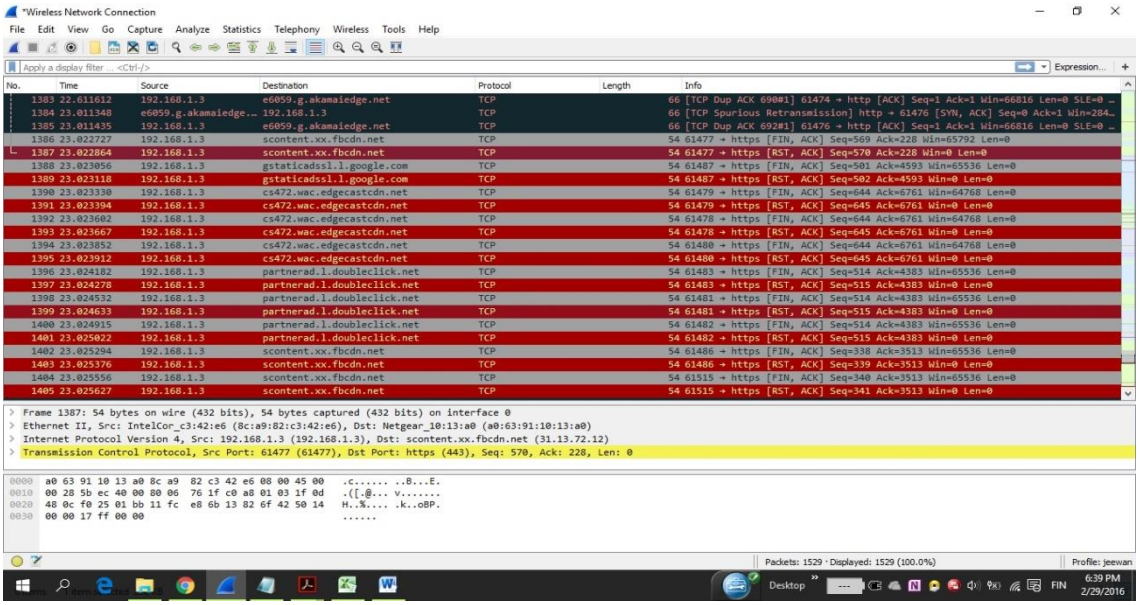

Figure 8. Wireshark GUI Screenshot

4.3.2 tcpdump

tcpdump is also a well-known network analyzer which runs in command-line interface only. It can analyze network behavior, view network login IDs, passwords, websites and its contents visited by a user. It is also a multi-platform analyzer like Wireshark. The latest stable release of tcpdump is 4.7.4 in April 22, 2015. It is also free to use. In order to use tcpdump, user should be logged in as root. However, tcpdump is not as easy to use and understand as Wireshark. It understands very limited number of protocols and can also be difficult for a normal user to read and understand printed output from command line.

4.3.3 Cain & Abel

Cain & Abel is free software for Microsoft Windows platforms, which can recover passwords. It can break security signatures by different attacks such as dictionary attacks, brute force attacks, cryptanalysis attacks to crack passwords. It can also recover passwords by sniffing packets in the network. The stable release of this software was

in April 7, 2014 as 4.9.56. This network sniffer can also crack wireless security like WEP cracking, and can also record VOIP conversations. This software can be used as a network tester, to find out its strength to stand out against attacks and improve network performance.

4.3.4 Colasoft Capsa

Colasoft Capsa is another advance network analyzer. However, its strong features are only available after paying certain price. It can also perform real-time packet capturing, network monitoring, protocol analysis and resolving application problems. It has quite strong in built features, well-designed GUI and can also capture both wired and wireless traffic. Other features of Colasoft Capsa are VOIP analysis, alarm notification by emails and audio for network problems, and it also provides automatic packet capture for pre-defined time. Similarly, another advance feature of Capsa is visual graphs and matrix feature to pinpoint network communication and protocol analysis. However, it is available for windows platform only [18].

4.3.5 General comparison

The following table, Table 2 points out the general properties of above mentioned network analyzers. It compares different features such as the operating system they support, disk space used, cost, and protocol support [19].

Table 2. Characteristic comparison of Wireshark, tcpdump, Cain & Abel, Colasoft Capsa

s.no.	Property	Wireshark	tcpdump	Cain & Abel	Colasoft Capsa
1	Os supported	Windows, Unix	Unix	Windows	Windows
2	Disk usage	81mb(windows), 448mb(unix)	448 kb	10 mb	32mb
3	Cost	Free	Free	Free	starts at 995 euros
4	User interface	GUI and CLI	CLI	GUI	GUI
5	Open source	Yes	Yes	No	No
6	No. Of protocols	more than 1000	Tcp/ip		300
7	UDP traffic	Yes	No	Yes	Yes

4.3.6 More description on analyzers

Although, only some examples of network analyzers are shown in this chapter there are other many kinds of network analyzers present in the market. Other widely used network sniffers are Ettercap [20], Kismet [21], Dsniff [22], NetStumbler [23], Ngrep, Ntop, Nmap and more. Ettercap is traditional terminal-based sniffer which supports lots of protocols and even ciphered protocols. Kismet and NetStumbler sniffs wireless traffic. Nmap is another useful tool for network scanning. There are so many tools and programs available as network analyzers, sniffers, packet analyzers in the networking world. And they can be of different shape, size and price. However, the proper use of these analyzers can provide strong and worry-free network.

The most basic functionality of these packet analyzers is to capture and filter network packets in real traffic flow. Some provide command line feature only and some of them provide GUI. Graphical user interface can be user-friendly for normal users and can always increase efficiency. Some of the programs can perform real attacks such as Ettercap, Cain & Abel and the main reason behind it is to point out network weakness. These real attacks can help to solve network vulnerabilities. Network analyzing software can perform penetration testing and injection attacks in the network. But the main reason for choosing Wireshark for further network analysis in this thesis, is because of its easy availability, and is considered one of the famous network protocol analyzer with strong built-in features. Wireshark can perform protocol analysis for more than 1000 network protocols, and supports almost all operating systems in the market. And its GUI is so well designed and easy to use. It is free and fulfills most of the requirements for network analysis.

If these programs are used for good reasons, it can provide strong network security. And there are people and groups who use this software in illegal activities which can steal any kind of information from other people. The misuse of these sniffers can always raise one question in normal human mind, Is my identity secure in internet?

5 WIRESHARK

This section of thesis describes how to install Wireshark in a machine, and the minimum requirements to install this protocol analyzer. And it further describes how to use its graphical user interface (GUI), which makes it easy and efficient to use. Similarly, the other benefits such as coloring rules, customizing its toolbars to make the analyzer more personal and helpful are also discussed. The Wireshark is also known for capture filters, display filters and huge benefits of expert Info function, I/O graphs which are also included in this section.

5.1 Installation process and getting started

Wireshark is already introduced in the section 3.3, as a famous network analyzer. In order to start using this software, at first the user needs to install it. Wireshark is always known for its strong features and benefits it provides in packet analysis. Some of its benefits are it supports more than 1000 protocols, well-known for its user-friendly GUI, and also supports all major operating systems in the market. However, Wireshark can only be installed in a machine which fulfills the following conditions [24]:

- at least 400 MHZ processor
- minimum 128 MB RAM
- At least 75 MB of available hard disk.
- Promiscuous mode supported NIC
- WinPcap capture driver.

Wireshark can be downloaded from the download section of its official webpage, http://www.wireshark.org. It supports Windows operating system, Mac OS X, and Linux based platforms. The download section contains its latest stable release installer for windows, DMG package for Mac OS and source code for Linux based platform. In windows, we can simply download **.exe** file and double click it to start the installation process. Wireshark needs WinPcap capture driver to run in windows platform. However, WinPcap comes together with the Wireshark installation package which can be installed during the same process. The official website for Wireshark provides all the necessary steps to install Wireshark for most of the operating systems. In Linux systems, we can first download the source code from the webpage. However, in debian-

based Linux distributions Wireshark can be installed from the system resource by simply entering the command **apt-get install Wireshark** in the console. And for other distributions it can be compiled from the source code. The steps to compile the source code are shown below:

- Download the source code from the official website of Wireshark.
- Extract the archive file as – **tar –jxvf downloaded_filename-version.tar.bz2**. However, the extracting command is different for different distributions.
- Create new directory and install it there.
- Use configure command, based on the distribution such as **./configure**.
- Enter make command to convert the source into binary format and complete final installation by using make install command.

Similarly, the installation process for Mac OS X begins after downloading the disk images (.dmg) package from the official Wireshark web page. In order to install Wireshark, first open the disk image and run the installer file that comes with the download package. The installer package also contains all required command line utilities, and a launch daemon. And further details for installation can also be found from the official Wireshark page.

After the installation process, the next step is packet capture and network analysis. First, the Wireshark might not look interesting at all. It amazes the user when the Wireshark performs the first capture of network activities. The steps to initiate and capture some data are shown below:

- Open Wireshark (can be started from shell or window manager)
- In the latest version of Wireshark, user can simply start capturing packets from the start window. The start-up window has capture option where it shows all the available interfaces in the network. The active interface is shown with the sparks. So, the capturing process can be started by just double-clicking the active interface and the capture begins.
- User can also start capture by going to Capture menu and selecting the Options from the Capture drop-down menu, which opens the capture interfaces window.

This window also shows the available interfaces and the user can start his/her first capture, by simply clicking the interface in which he/she wants to perform the capture.

- Wait for some time until Wireshark captures some amount of data. When the user is ready, he/she can click the Stop button from the Capture drop-down menu.

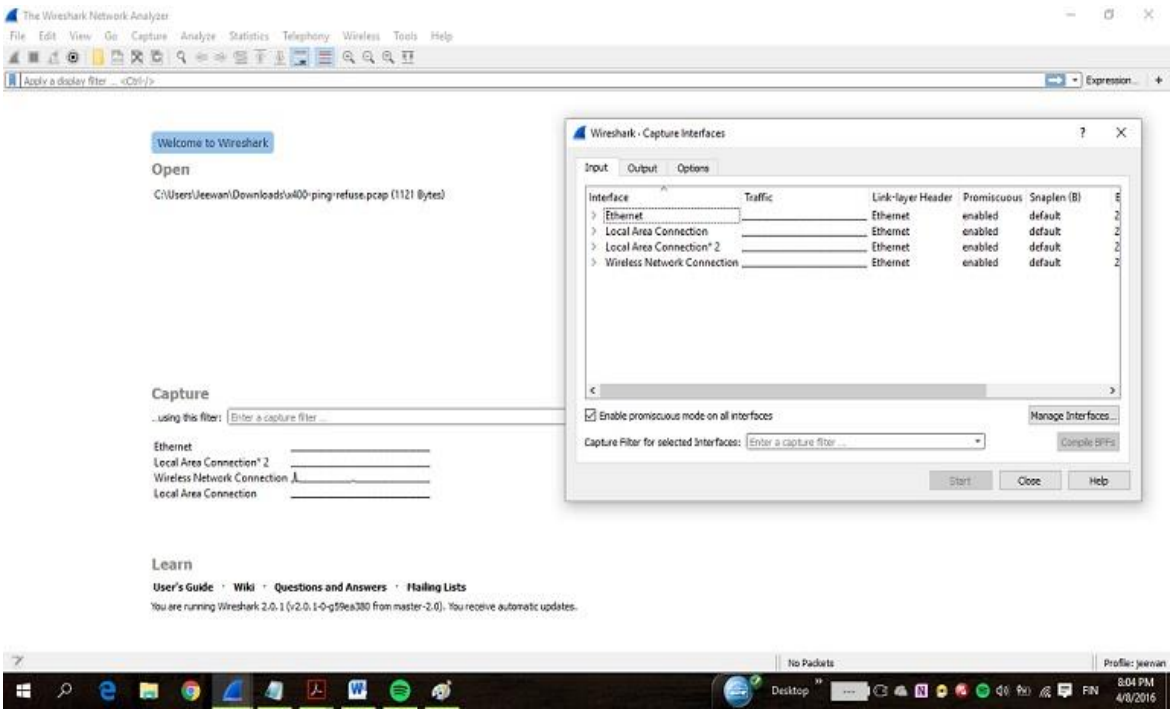

Figure 9. Selecting an interface to start packet capture

5.2 Graphical User Interface (GUI) of Wireshark

The Graphical User Interface (GUI) of Wireshark has different sections with equally important function to make task easy and user-friendly. In Figure 10, we can see different sections of user interface after some packets are captured.

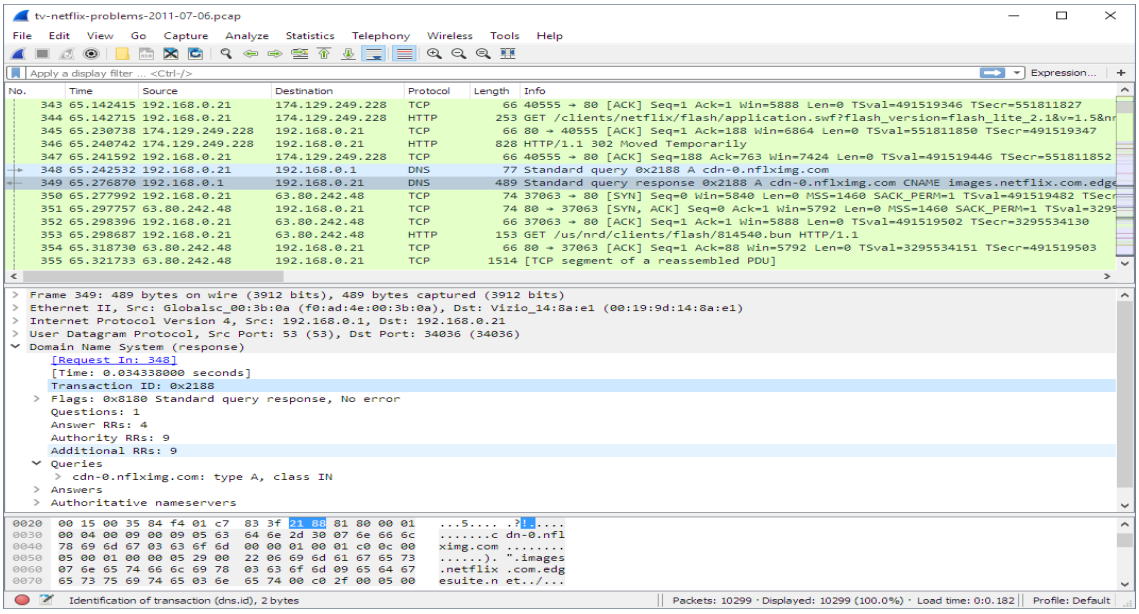

Figure 10. The Main Window [24]

The main window of Wireshark consists of following parts:

- The menu

The main menu of Wireshark is located at the top of Wireshark main window. The main menu items are shown in the Figure 11 below:

Figure 11. The Menu [24]

- File

This menu contains drop-down items. These items can open and merge capture files, save, print, or export capture files and also contains quit function to quit the Wireshark application.

- Edit

This menu contains drop-down menu items. The items help to find a packet, move from one packet to another by find next and find previous. It also has items to mark and un-mark packet. Similarly, the edit menu also contains item to create configuration profile.

- View

The view menu has items to control toolbar display. It also controls the colorization of packets and their rules.

- Go

This menu contains items to reach to one specific packet and also contains items which help to move from one packet to another.

- Capture

This menu controls the capture function. The items in this menu can start and stop a capture. It also contains settings for capture interfaces and capture filters.

- Analyze

This menu includes items to modify display filters, follow stream like TCP, UDP traffic, dissector and plugins function and also has item which provides expert information.

- Statistics

This menu provides I/O graph and other different statistics for HTTP protocols, DNS protocols and many more. Similarly, it also provides information through Flow graph and many more.

- Telephony

This menu contains items to display various statistic windows related to telephony. It provides media analysis. It includes information on VOIP calls, GSM packet analysis and more information through graphs for telephony streams.

- Wireless

This menu has items to describe Bluetooth and Wireless LAN traffic statistics.

- Tools

This menu has items to write dissectors for Wireshark through Lua programming. It also has items for creating different Rules such as Firewall access control lists.

- Help

This menu contains items to fully help the user benefits from this network analyzer. It provides help through manual pages, online access to documents and wiki in the web page. The user can remove any sort of confusion about Wireshark through the help of this menu.

- The main toolbar

The main toolbar is a shortcut to the frequently used items from the menu. In general, the most used items from the menu are put with symbols to make work easy for the user. The figure below shows the main toolbar.

Figure 12. The main toolbar

The function of each item in the main toolbar can be obtained by just hovering the mouse cursor above those items.

- The filter toolbar

The filter toolbar helps to edit and apply the display filters. This toolbar contains area to enter and edit a display filter string. The packets related to the specified display string and protocols are only shown because of the display filter. In Figure 13, we can see the example of filter toolbar.

Figure 13. The filter toolbar [24]

- The packet list pane

The packet list pane is an important section of Wireshark main window. This packet list pane displays all the packets captured in a specific capture session. Each line in the packet list represents a single packet. The more details about a packet can be obtained in “Packet Details” pane and “Packet Bytes” pane by simply selecting a line from packet list pane. This list pane contains different columns to provide details about packets. The default columns are **No.** (Number of packets captured), **Time** (time spend), **Source** (source address of packet), **Destination** (destination address of packet), **Protocol** (name of used protocol), **Length** (length of each packet) and **Info** (additional information on packet content). In Figure-14, the packet list pane is shown.

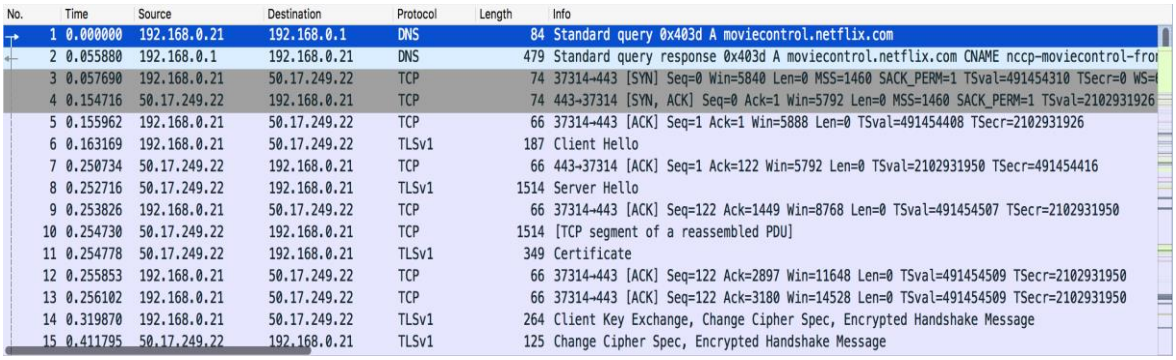

Figure 14. The packet list pane [24]

- The packet details pane

If the packet is selected in packet list pane, the more detail information can be obtained from the packet details pane. This pane provides information about the protocols and protocols fields of the selected packet. The protocols and fields of the packet are shown in the tree hierarchy which can be expanded and collapsed. In Figure 15, example of packet details pane is shown.

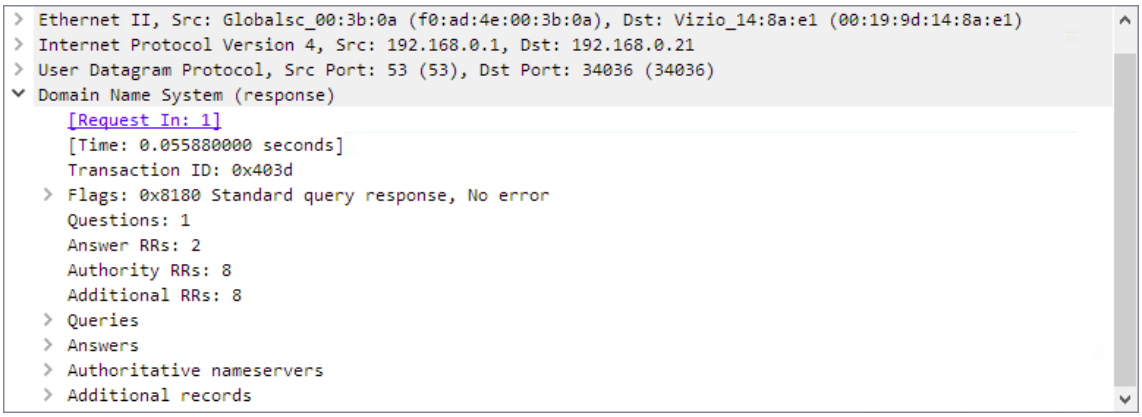

Figure 15. The packet details pane [24]

- The packet bytes pane

When a packet is selected in the packet list pane of Wireshark, the packet byte pane shows the information of the selected packet in a hexdump style. An example of the packet byte pane is shown in the Figure 16 below:

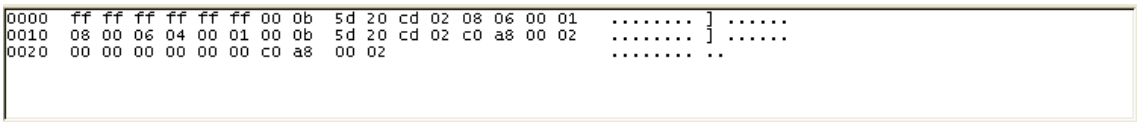

Figure 16. The packet bytes pane [24]

The left side shows the data offset in the packet, the middle part represents data in hexadecimal form and the right part shows packet data in ASCII characters in corresponding to middle data. Sometimes it can show packet information in different tabs, after Packet assembling done by Wireshark.

- The Status bar

The status bar is a place to show the information about the number of captured packets, number of displayed packets in packet list panel, and configuration profile used to capture the data. In Figure 17, an example of the status bar is shown.

Figure 17. The initial empty Status bar

When Wireshark is just started or no capture file is loaded an empty status bar is shown. However, when a capture file is loaded or any capture is performed the status bar shows the following information as shown in the Figure 18.

Figure 18. The Status bar after a capture is loaded

In this loaded status bar, the left bottom corner shows the highest expert info, next to it is the name of the captured file. Similarly, on the right side number of captured and displayed packets are shown, and at the right bottom corner is the place for configuration profile used to capture data.

5.3 Wireshark customization

Yes, Wireshark also has an amazing feature of changing settings to fit our needs. We can always customize Wireshark to make work faster, easier and to perform detail analysis. Wireshark can be customized in command line also. There are different commands available in command mode to perform different functions such as path setting, capture filter, capture interface setting, time stamp format and many more.

Another important feature which can be customized in Wireshark is protocol dissection. Each protocol is dealt by its own dissector. Dissector performs the observation and disassembling of a protocol to study its functions. In Analyze menu, we can find an item called Enabled protocols. The enable protocol dialogue box helps a user to enable or disable some protocols. However, all protocols supported by Wireshark are enabled by default. And there are other features which help to temporarily divert some protocol route and to change its ports. Similarly, different user profiles can be created by right clicking in the profile tab in the right bottom corner of the Wireshark window. Configuration profiles can store different sets of preferences and configurations. In these profiles, recent changes in the settings are saved. It also saves pane size in the main window, number of columns and its width in packet list. Similarly, there are more features that can be customized such as display filter macros, database paths, user table, protocol table, color customization etc.

Packet colorization is another significant feature of Wireshark. This feature helps to color a packet according to the display filters and protocols. There are two ways of coloring the packet. First one is temporarily saved and other one is permanent rule which is saved in the preference file and available next time. Inside **View menu**, there is an item called **coloring rules**. The color settings for different protocols can be performed from coloring rules dialog box. The figure below shows the default coloring rules dialog box for default configuration profile.

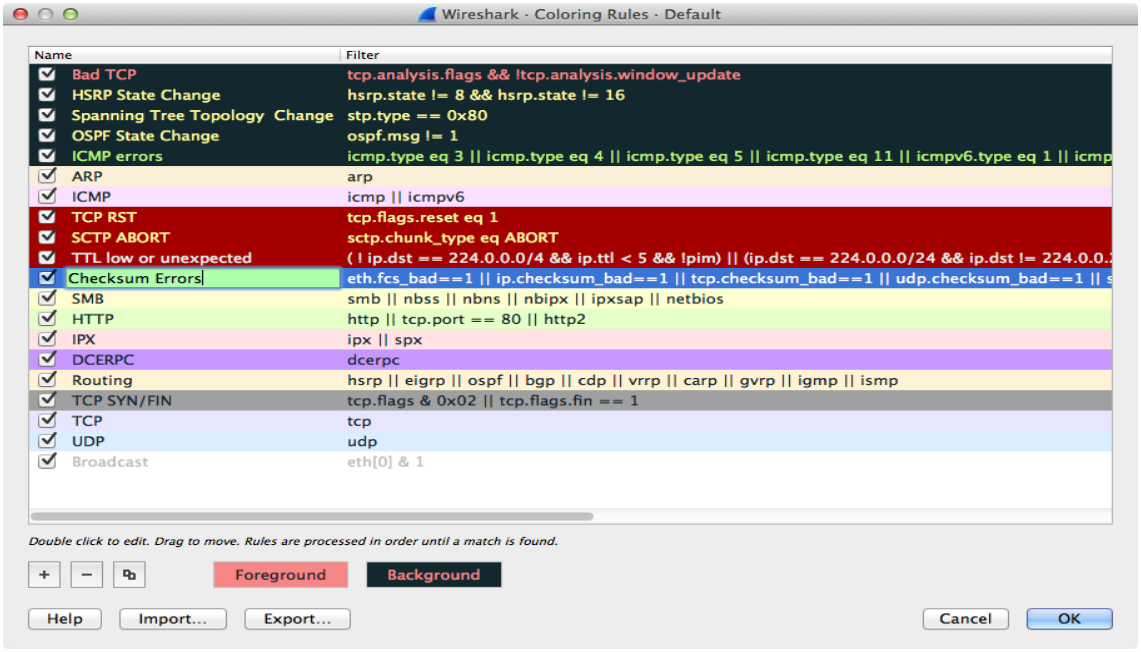

Figure 19. The coloring rules dialog box [24]

New coloring rules can be created by simply clicking on the **+** button and some existing coloring rules can be deleted by clicking the **–** button. Similarly, we can duplicate a rule with the help of **copy** button which is next to - button. When a rule is selected, the options for foreground and background color become active. The foreground and background buttons open a color chooser dialog box which helps to select different colors for background and foreground for a certain rule. The coloring rule higher in the order replaces the color for another similar protocol coming later. For instance, if there is a coloring rule for UDP traffic before the rule for DNS. The color for UDP traffic replaces the color for DNS traffic. This is because DNS is also considered as UDP traffic. The

figure below shows the color for bad TCP rule applied in the packets 176,177,287 and 1471 along with the tcp filter.

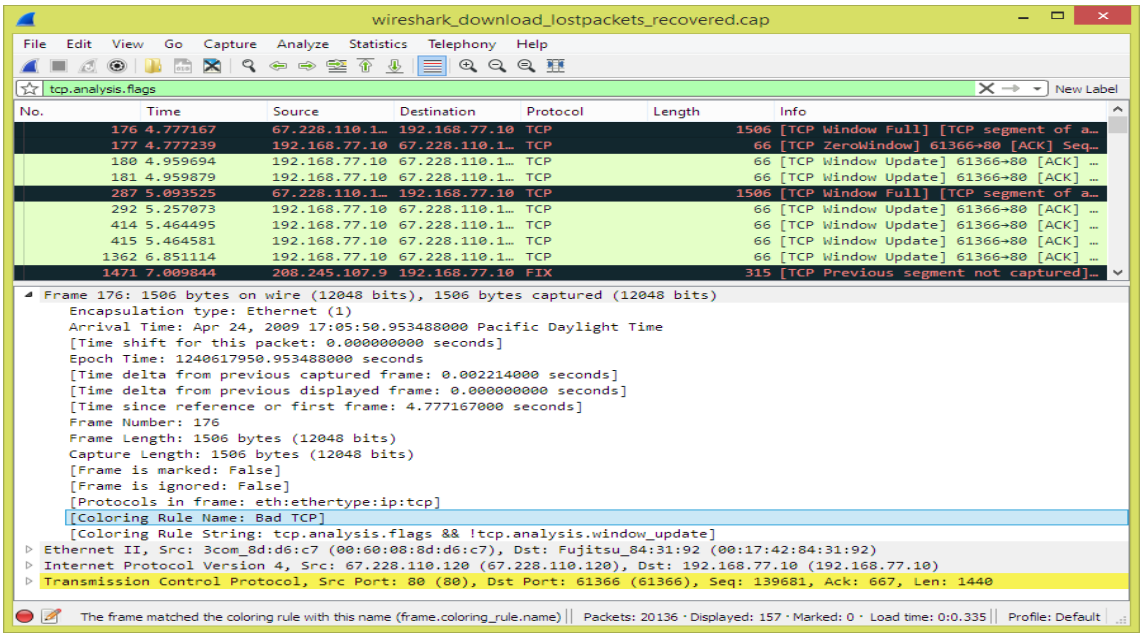

Figure 20. Colors used in Wireshark

5.4 Controlling Capture with Filter

Capturing data in a live network is considered one of the main duties of Wireshark. Most of the network analysis starts after capturing packets in the network. Wireshark can capture packets in Ethernet network or wireless network. Packet capture cannot be performed in any network. It is always important to make sure that we have permissions to capture packets in the network we are working on. Packet capturing in someone's private network without permission is considered illegal. Capture privilege, proper capture driver, and right network interface are some basic requirements to start packet capture. Packets can be captured in local network or remote network. Therefore, it's important to find out the place where the protocol analyzer can be tapped in to capture traffic. In section 6.1, the more description of where to capture data in a network will be done. The captured data can be saved in .**pcap** Wireshark's default file format. Similarly, it is also possible to export these capture files in different formats such as plaintext, comma-separated values (CSV). Hence, several captures can be made at once, saved

and can be analyzed all together afterwards. In addition, it is important to fix correct time display during packet capture. The correct time display format is a great help during packet analysis. It gives the absolute timestamp about the exact moment when a packet was captured and it's time relation with other captured packets.

Filter is a language expression which can be used when capturing packets and when displaying packets to include or exclude packets in the capture. In other words, filters help to select and show packets depending on the user's choice. There are two types of filters used in Wireshark and they are described below:

- Capture Filters

Capture filters are applied when packets are being captured in a network. After applying capture filters, the Wireshark will capture only those packets which are asked to be included or excluded by applied capture filter expression. By applying capture filters during actual packet-capturing process saves processing power, time and improves performance.

In Wireshark, choose **Capture – options** this opens the **Capture interfaces** dialog box. First, the desired interface where packets are to be captured should be selected and capture filter can be entered in the capture filter field in the left bottom corner. When a correct filter expression is entered in the text field the background changes to light-green color. WinPcap driver controls the implementation of capture filters. And in WinPcap libraries, the Berkeley Packet Filter (BPF) syntax is used to create capture filters. An expression is defined as the filter created using BPF syntax. This expression is also called primitive. Primitive can contain qualifier and ID. Capture filter can contain one or more primitives. However, to connect one or more expressions logical operators are used. In Figure 21, the dialog box to enter capture filter expression is shown.

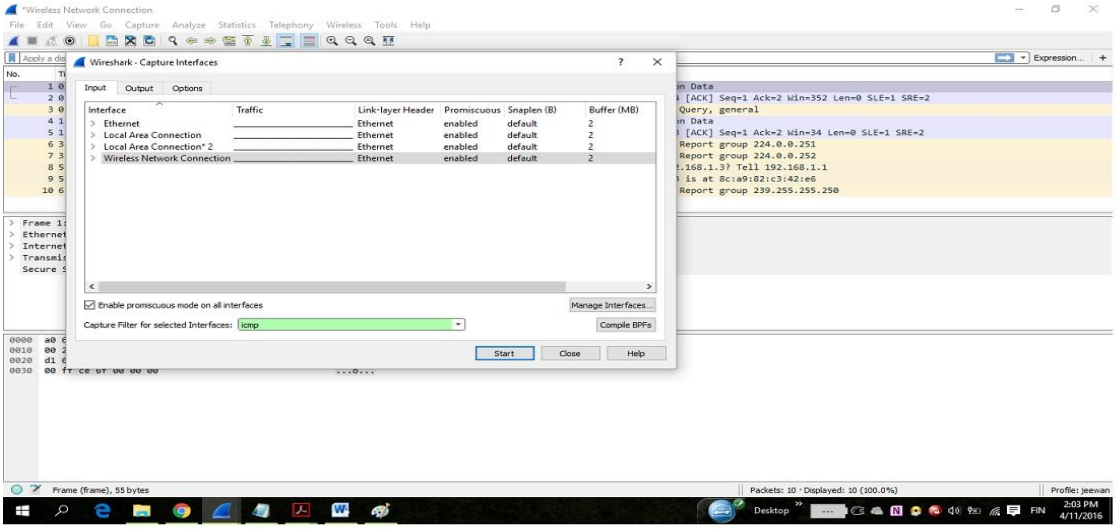

Figure 21. An example of ICMP capture filter used in Capture interfaces dialog box

Capture filters can be used in different forms to achieve different targets. In general, there are BPF syntax, hostname and addressing filters, port and protocol filters, and protocol field filters. In Table 3, some examples of these types are presented.

Table 3. Examples of Capture Filters

S.no.	Types	Example	Description
1	BPF Syntax	src 192.168.2.20 && port 80	src 192.168.2.20 = primitive. port 80 = primitive. && = operator.
			captures traffic with a source Ip address of 192.168.2.20 and a source or destination port of 80
2	Hostname and Addressing Filters	host 192.168.2.20	Only captures traffic for this specific IPV4 host address
		ether host 00-1a-a0-52-e2-a0	Using MAC address of a host as capture filter. Traffic to or from this MAC address is captured.
3	Port and Protocol Filters	port 80	Captures traffic only on port 80
		!port 80	excludes traffic from port 80
		icmp	ICMP traffic only
		ip6	IPv6 traffic only
		ip	IPv4 traffic only
		udp	UDP traffic only

- Display Filters

Display filters are applied to the capture file. When applied to the captured data display filters tell the Wireshark to only show the packets that match the applied filter. The dis-

play filter text box is shown above in Figure 13. Display filters are used more than capture filters. It is because packet analysis can be done for a specific data without actually removing all captured packets. It helps to simply see some form of traffic, work with that and also gives the opportunity to go back to the original capture. Display filters use comparison operators to compare values. For instance, ip.addr == 192.168.1.20 this expression filter shows all packets with the IP address of 192.168.1.20. Here, the equal- to (==) comparison operator is used. The Table 4 below shows some examples of comparison operators, logical operators and commonly used display filters.

Table 4. Wireshark Filter expression operators and some display filters

Comparison Operators	Description	Logical Operators	Description	Display Filters	Description
==	equal to	and	both conditions must be true	http	all HTTP traffic
>	greater than	or	either one condition must be true	!arp	clear ARP traffic
<=	less than or equal to	not	neither one of the conditions is true	tcp	all TCP traffic

5.5 Other features of Wireshark

Wireshark is a useful tool for a beginner to practice packet analysis and also a strong analyzer for experts in packet analysis field. Wireshark is a free packet analyzer but still it provides huge number of benefits in everyday use. It supports large number of protocols, built with user-friendly GUI, and supports most of the operating systems.

Wireshark is not just limited to packet capture. It provides other powerful features such as stream following, IO graphing, expert info, name resolution. When someone fully understands and uses these advance features, he/she can master the art of packet analysis with ease. The Name resolution feature helps to convert numeric address into easy to read DNS name. Similarly, graphing is another strong feature of Wireshark. Graphs can help to find out performance problems in protocols, to compare data streams and to find out latency issues. There are different types of graph available in Wireshark. They are IO Graphs, Round-Trip Time Graphing, and Flow Graphing. The IO graph is in form of spikes and lulls. It can be built from **Statistics – IO Graphs**. Another one is Round-Trip Time Graphing (RTTs), which visualizes round trip times for a given capture file in the form of graph. The main task of this RTTs graph is to find out the latency in the network. It finds out the slow points or slow connection in the net-

work. Flow Graphing is another feature of graphs available. This graph shows the flow of data over time in the form of column-based view.

Another strong feature of Wireshark is expert information. Expert information provides information in the form of Chat, Note, Warning and Error. The expert info provides information about any uncommon or notable events occurring in the network. It provides notes, warnings and errors occurred in any protocol connection. The main idea behind this is to help the user discover issues related to protocol and connection in the network. To summarize, all the features and characteristics described above about Wireshark makes this packet analyzer one of the top most desired software in packet analysis field.

6 TRAFFIC ANALYSIS WITH WIRESHARK

6.1 Where to capture data

Packet analysis solves lots of problems related to the network. In computer networks, when network performance degrades it is very important to find out the reason. Some of the networks have installed alarm system like Intrusion Detection System (IDS) to inform about network attacks and malfunction. Network issues and attacks can arrive from the internet, internal network, software bugs, hardware failures etc. However, these issues can be monitored through packet analysis and Wireshark. But analyzing traffic is not as easy as just plugging a laptop into a network port and start capturing traffic. It is always considered a difficult task to find the right spot to measure essential traffic flows with analyzers. Network architecture consists of complex cabling, hub network, switched and routed environment. In order to appropriately capture the traffic, it is important to have knowledge about networking hardware, cabling system and the way these devices work in networking world. Therefore, this process of placing a packet sniffer in the current physical location in the network is called as tapping the network, tapping into the wire or sniffing the wire. Some techniques to understand where to place packet sniffer like Wireshark in a network are discussed below. These techniques are also used by hackers to eavesdrop in a network [25].

6.1.1 Using a Hub

In computer networks, hub is a device which contains multiple ports and acts as a common connection point for other devices. When a packet arrives in one port of the hub, it sends this packet to all other ports as a broadcast traffic. Hence, to analyze traffic for any device connected to a hub the packet sniffer has to be connected to any empty port on the hub. An example for this is illustrated in Figure 22. However, nowadays hubs are hardly used in modern network designs because of the collision issues between two devices transmitting at the same time.

6.1.2 Switched and Routed environment

Most of the methods used in switched and routed environment to capture traffic are quite common. Packet analysis is performed after the data is captured. The right placement of packet sniffer gives correct data of the network. And the correct data can help troubleshoot the network and analyze the network process. Therefore, the correct placement of sniffer is directly related to the successful traffic analysis. Some of the

common methods used in switched and routed environment to place a sniffer are discussed below. These methods are used by network administrators and network troubleshooters to secure and analyze the network. However, these same methods are used illegally by hackers to benefit from other's information.

- Port Mirroring

In switched network, port mirroring is the process of duplicating all traffic between one or two switch ports and mirror it to one single port we desire. For example, to capture traffic in the second port of a switch. Port mirroring can be performed by plugging in the packet analyzer into port 3 and mirroring port 2 to port 3 which is also shown in Figure 21. Although port mirroring is considered an easy way to capture traffic, there are few things to consider. The switch should have port mirroring function and an empty port available. After mirroring some number of ports to one single port, it is important to understand physical capacity and bandwidth of a single port for traffic flow to avoid packet loss or network slowdowns.

- Bridge Mode

This process of traffic capture is performed when the switch device is not physically accessible. For instance, if the traffic between the switch and the server has to be captured this method can be used. A machine having two network cards is positioned between the switch and the server. The machine can be configured in bridge mode by installing bridge packet utilities which are mostly available for Linux. After the pc configuration, the Wireshark can be started to capture traffic. In Figure 22, modo is the Spanish word for mode.

- Using a Tap

A Tap is a network device which is placed in between two points in computer networks to capture traffic. There are two types of tapping devices. They are aggregated taps which contain three ports and non-aggregated taps containing four ports. The main way of using this hardware based device is to place it, in a network cabling system and capture traffic. For instance, an aggregated tap can be connected in between a computer and a switch by simply plugging the computer and switch cable to the tap ports. And connecting the sniffer machine with Wireshark installed in the third port of the tap device. After this setup, the sniffer machine can capture all the traffic flowing in and out

of the computer. However, non-aggregated taps have two ports to connect two different networks and other two ports as monitoring ports to connect monitoring program such as Wireshark. The two monitoring ports such as M1 and M2 receive traffic for two network ports N1 and N2 respectively and packets can be captured through the monitoring ports.

- ARP Cache Poisoning

ARP cache poisoning also known as ARP spoofing is the process of sending ARP reply messages to the router or switch with fake layer 2 or MAC addresses. In computer networks, ARP table plays an important role for the traffic flow to start. When fake MAC addresses are stored in the ARP cache table of switches and routers, the traffic flow is diverted to the fake position. This process can cause Man-in-the-Middle attacks and other denial-of-service (DOS) attacks in the network. However, ARP cache poisoning is sometimes used as a legitimate way of traffic capturing by network administrators of a target machine to solve different issues in the network. When the traffic is diverted to another location, a computer device having Wireshark installed is placed to capture traffic. This same process is also used by hackers to steal sensitive information from the network.

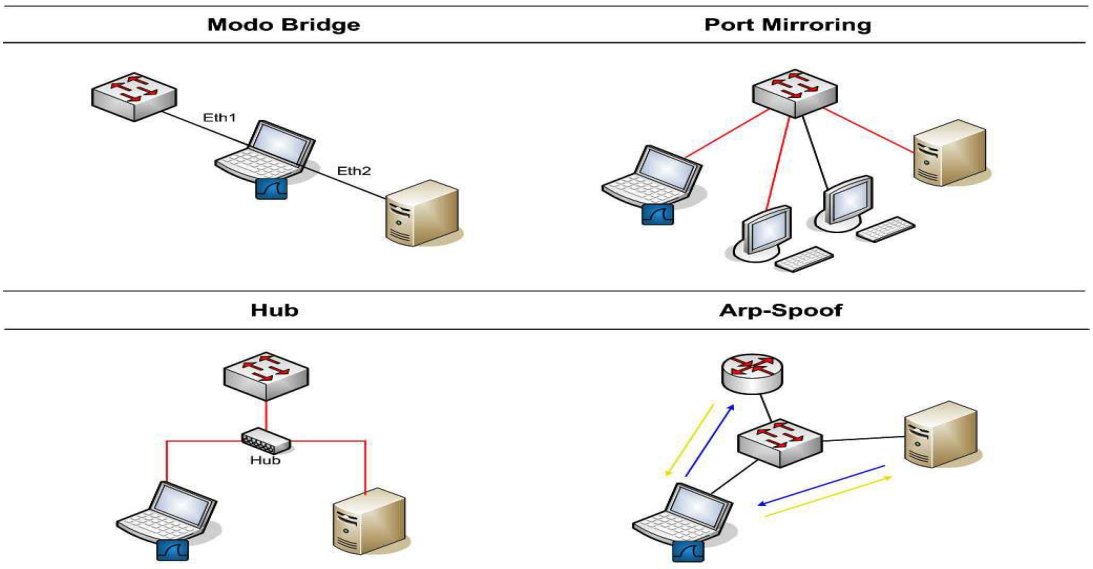

Figure 22. Capture Modes [25]

6.2 Investigating Protocols

In Computer networks, network threats and security issues are always difficult to monitor. However, a good knowledge of network architecture and network traffic can solve many issues in the network related to traffic congestion, bandwidth and security threats. In this section, some commonly used protocols in network communication are analyzed. The well-known network protocol analyzer, Wireshark, is used to perform detailed monitoring of these protocols. Network traffic of my own local area network (LAN) is captured to investigate and understand the characteristics of these protocols [26].

6.2.1 IP

The Internet Protocol (IP) is the layer 3 protocol of the OSI model, which allows internetwork communication. There are two version of IP and they are IPv4 and IPv6. The most commonly used is IPv4 at this point of time. This protocol can carry data between devices regardless of location, and complexity in the network. Each device in the internet is identified by 32-bit unique addresses called IP addresses. For instance, 192.168.1.7 is an example of IP address. In layer 3, internet protocol binds information data in the form of packets which is delivered to the destination by using transport protocols at layer 4. In Wireshark, we can see all the IP details of a packet in the packet details pane. The IPv4 header has the following fields related to a packet:

- Version: This field shows the version of IP being used such as IPv4, IPv6
- Header Length: This field shows the length of the IP header. For instance, Header Length = 20 bytes.
- Type of Service: This field sets the priority level of the traffic on the basis of service flags. This value is considered by routers to deal with the traffic.
- Total Length: This field is the sum of the length of the IP header and the data present in the packet.
- Identification: This is the unique identification number given to the packet for its identification or sequence number of fragmented packets. For example, 0x8ecc (36556).
- Flags: This field identifies if the packet is the part of the fragmented packets.

- Fragment Offset: This field provides value for reassembly of a packet if it is the fragmented one.
- Time to Live: This field provides the life span of a packet which is measured as hops/seconds through routers. The Time to Live (TTL) decreases by 1 every time the packet is forwarded by a router.
- Protocol: This field tells about the type of packet coming next.
- Header Checksum: This field finds out errors and also verifies the contents of the packet are not damaged.
- Source IP Address: This field provides the IP address of the host which sent the packet.
- Destination IP Address: This field provides IP address of the packet's destination.

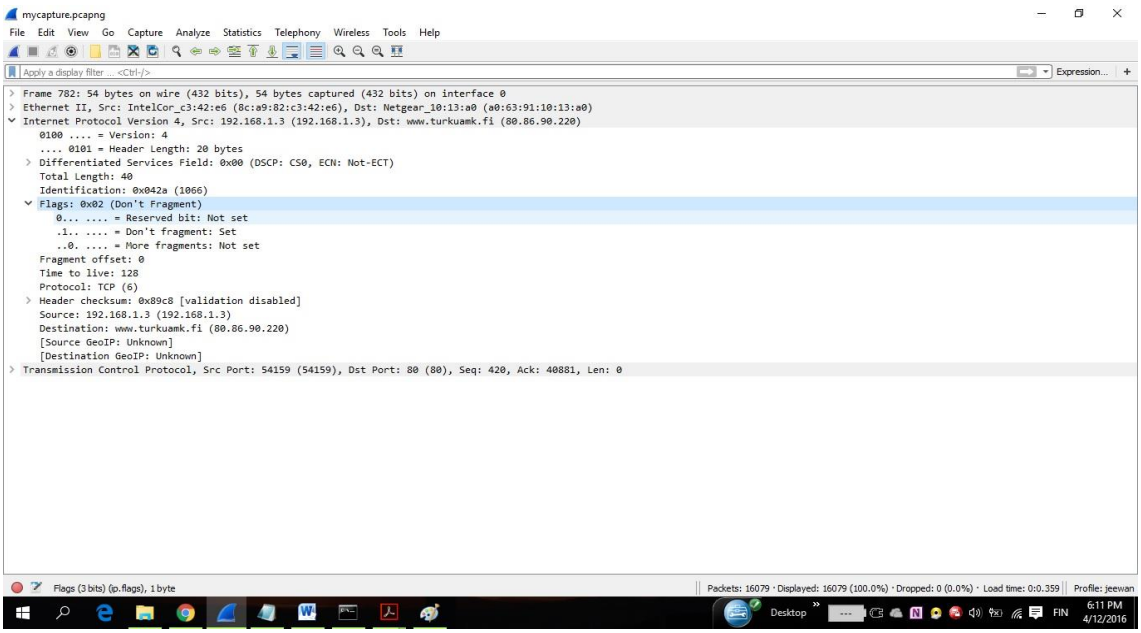

Figure 23. The IP header of the source packet

6.2.2 ARP

In computer networks, IP address of the remote host is resolved through Domain Name System (DNS) protocol which initiates layer 3 to layer 7 communications in OSI

model. However, physical address or MAC address is required for directly connected devices to communicate. And this resolution process of IP to MAC address is done by Address Resolution Protocol (ARP). This process has two parts ARP request and ARP reply. The ARP header can be seen in Wireshark in the packet details pane, which contains the following fields:

- Hardware Type: This field represents layer 2 hardware type. For example, Ethernet (1).
- Protocol Type: This field shows the higher layer protocol used for ARP request.
- Hardware Address Length: The total size of hardware address is shown. For instance, hardware size is 6 for Ethernet hardware type.
- Protocol Address Length: The total size of the protocol used is shown. This length is represented in octets/bytes form, which is 4 for IPv4 protocol.
- Operation/Opcode: This value is 1 for a request and 2 for a reply.
- Sender MAC address: The physical address of the sender.
- Sender Protocol address: This field provides the sender's protocol address.
- Target MAC address: This field provides the intended receiver's physical address, which is unknown for ARP request.
- Target Protocol Address: The protocol address of the intended receiver is shown.

Figure 24 below shows the ARP request captured in Wireshark

.

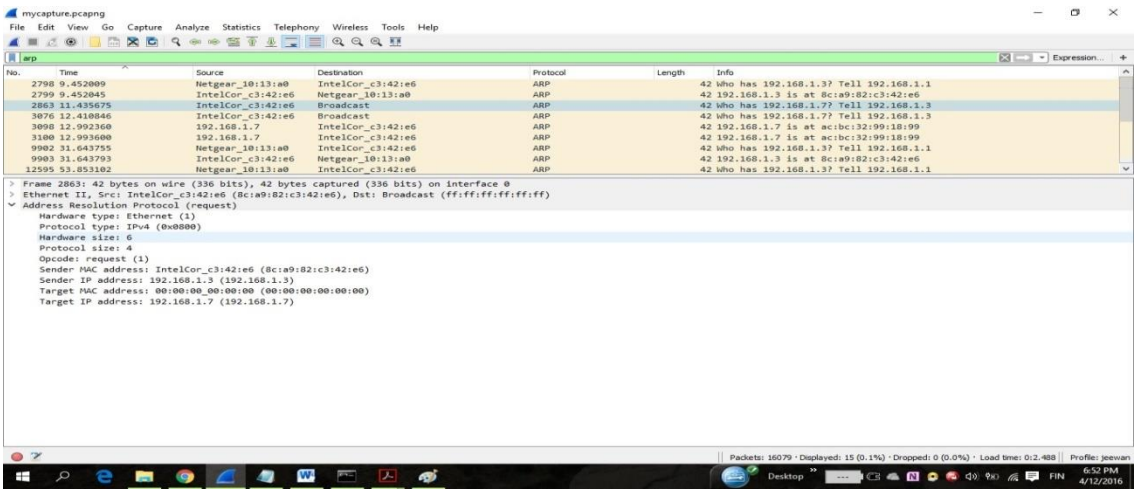

Figure 24. ARP request captured in Wireshark

In the figure above, the packet 2863 is sending an ARP request which can also be referred as ARP broadcast traffic. The Network Interface Card (NIC) of my computer is sending broadcast request to all other host in my local area network. My computer is asking the MAC address of 192.168.1.7 host to communicate and to update the ARP table.

Similarly, in Figure 25 ARP reply captured in Wireshark is shown.

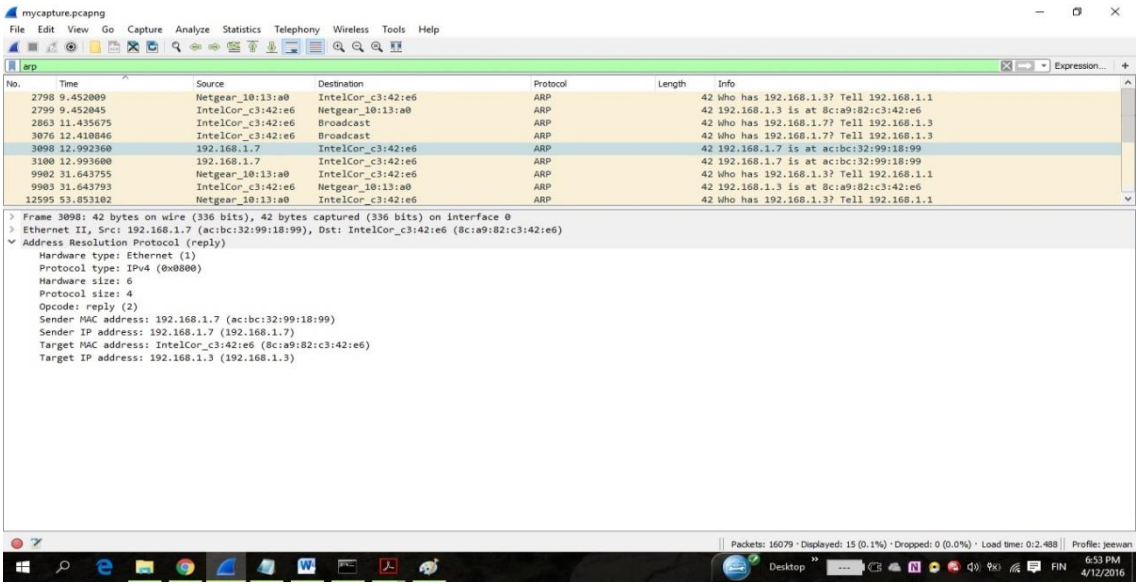

Figure 25. ARP reply captured in Wireshark

In the figure above, the packet number 3098 is sending an ARP reply back to my network interface card. This packet provides details about the MAC address of 192.168.1.7 client. Hence, my computer can update its ARP table for future communication with 192.168.1.7 client. And in the above shown packet details pane of Wireshark, all the values of ARP header can be obtained. For example, Hardware Type: Ethernet (1), Protocol type: IPv4 (0x0800), Hardware size: 6, protocol size: 4 and so on.

6.2.3 TCP

Transmission Control Protocol (TCP) provides reliable communication between two ends and ensures reliable data delivery. This transport protocol operates at layer 4 of the OSI model which supports sequencing and error control of the data. This protocol delivers data to the destination without dropping or misplacing the packet information. The TCP communication is established using source and destination ports. There are standard ports from 1 through 1023 and ephemeral ports from 1024 to 65535 for the operating systems to select any random port for unique communication. TCP communication starts with three-way handshake. For instance, if host A wants to communicate with host B. At first, host A sends packet without data as SYN flag set which contains initial sequence number and maximum segment size. Host B replies to this packet with SYN and ACK flags set. And at last host A sends the last packet containing ACK flag set. If this process is successful, these two hosts start to communicate. However, the end of TCP communication session is completed by sending four packets using FIN flags. For example, Host A informs Host B about the end of communication by sending FIN and ACK flags set. Host B responds with ACK packet and further sends its FIN/ACK packet. Finally, Host A sends an ACK packet and ends the communication session. However, when a connection is ended quickly without any notice or for any connection refusal indication the TCP resets flag is used. So, the RST and ACK flags also indicate the end of communication.

Some of the packets in TCP communication go under retransmission. The TCP provides reliable communication. Therefore, this protocol sends packet under retransmission to avoid losing or damaging the data. The TCP header has following fields, which can also be seen in the packet details pane of Wireshark.

- Source port: The port used by sender's device to transmit the packet.

- Destination port: The port where the packet is transmitted.
- Sequence number: TCP uses sequence numbers to keep track of each segment during a session. This field is important for a reliable TCP session. This value keeps an eye on each segment and ensures that any data is not missing.
- Acknowledgment Number: This field provides value to be expected as the sequence number in the next packet from the other device taking part in the communication.
- Flags: This field provides value as flags for identifying the type of TCP packet. Examples of the flags used are URG, ACK, PSH, RST, SYN, and FIN.
- Window Size: This field provides value for the size of TCP receiver buffer.
- Checksum: This field ensures that the content of the TCP header and data are complete upon arrival.
- Urgent Pointer: This field is active when URG flag is set and provides information for the CPU about data reading point in a packet.

In Figure 26, the TCP header captured in Wireshark is shown.

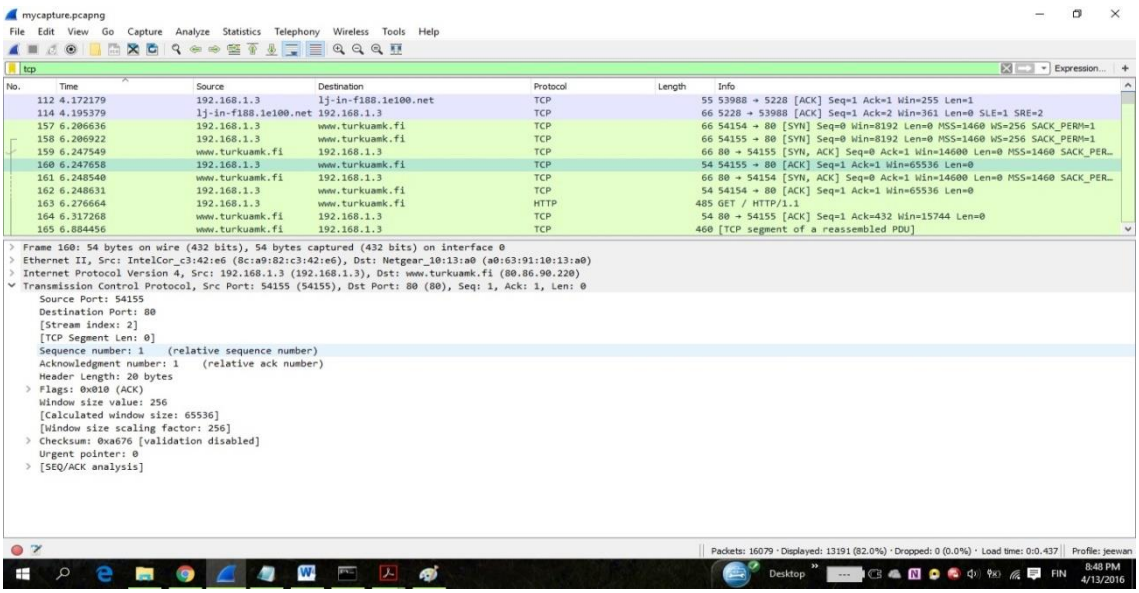

Figure 26. TCP connection and header captured in Wireshark

6.2.4 HTTP

The Hypertext Transfer Protocol (HTTP) is a server/client based protocol which delivers web pages on the network. This protocol uses transport layer protocol to establish reliable connection and delivers the web page. The Figure 27 shows HTTP communication in Wireshark and the Figure 28 shows some packets captured.

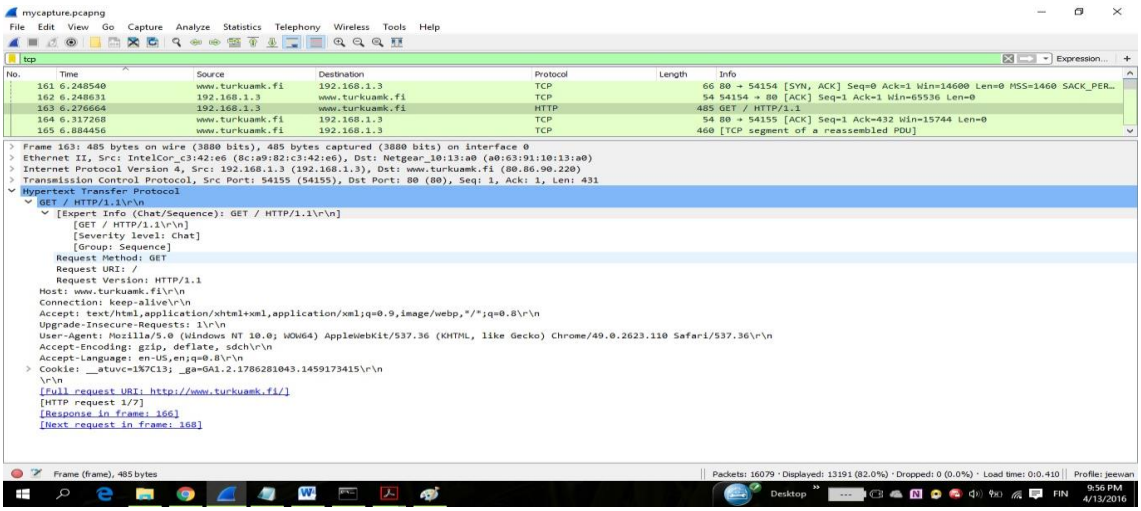

Figure 27. HTTP Get request packet

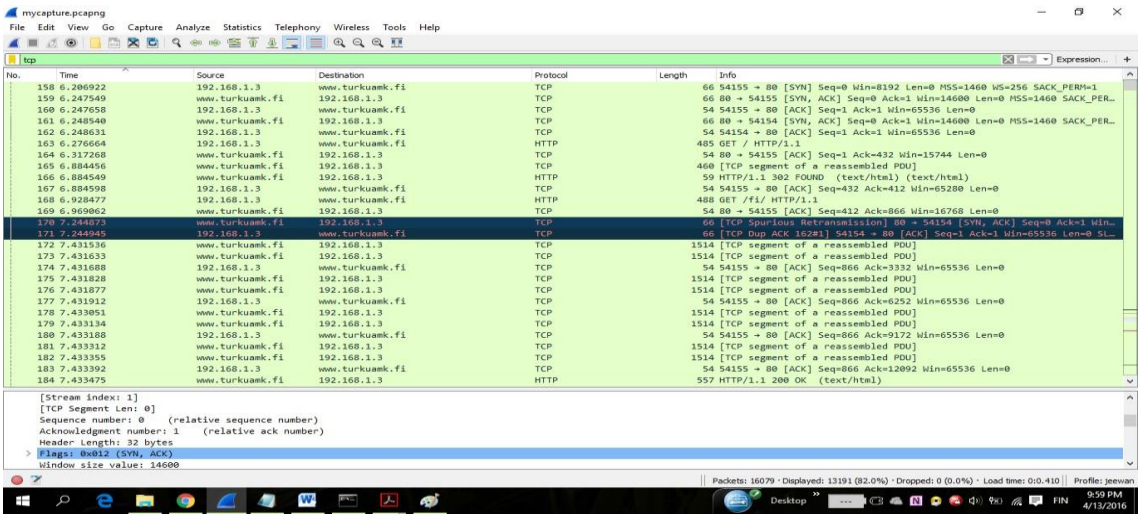

Figure 28. HTTP packets captured in Wireshark

In Figure 27, the browser requests for the webpage http://www.turkuamk.fi, as seen in the HTTP field in packet details pane. But the HTTP packets are delivered by TCP to and from server's port 80. The TCP handshake is being established at packet 158 from my computer's random source port 54155 to port 80 by sending SYN flag. Similarly, at

packet 159 the web server replies with SYN/ACK flag. And at packet 160, the host replies to the server with ACK flag and TCP connection is established. The packet at 163 sends GET request method of HTTP to the web server. As shown in the figure, this method tells the server that the client is requesting to download the web page from the web server's root directory by using version 1.1 of HTTP. Similarly, the host also sends information about its browser, languages accepted by it and some cookies information too.

In Figure 28, when the server receives HTTP GET request at packet 163 it responds with TCP ACK flag, acknowledging the packet, and begins transmitting the data in packet number 165. However, in packet 166 the web server sends HTTP 302 status code message to the host. This means the host is redirected to another location and the location is http://www.turkuamk.fi/fi. This is because the contents for the initially requested address were found in another directory /fi/ by the web server. After this information of redirection was received by the client from the web server, it sends another ACK flag to the web server at packet 167. And again GET request is received by the web server at packet number 168. Finally, the data is being transmitted from the web server starting at packet number 172. Data is sent in packet 173 from the server and an acknowledgment is sent by the client in packet 174. Similarly, more data received by the client in 175 and 176 followed by acknowledgment in packet 177. And more data in packet 178,179 followed by another acknowledgment packet from client in 180. More data is received from the server in 181,182 and again the client acknowledges server in 183. Finally, the web server replies with status code 200 and response phrase OK. This means the request made by the client is fulfilled by the server and the page is downloaded. The HTTP packets were downloaded through nine reassembled TCP segments, although HTTP is responsible for the transmission.

6.2.5 ICMP

Internet Control Message Protocol (ICMP) provides information about the availability of devices and routes in the network. This protocol is an important feature of networking to troubleshoot a network. Ping is one of the features of this protocol, which tests the connectivity of a device. In general, **ping** command sends one packet at a time and waits for the reply packet to determine the connection between two devices. The basic two steps are ping request and ping response. This protocol also solves issues related to unreachable destinations and ports. In Figure 29, ICMP packets captured in Wireshark are shown.

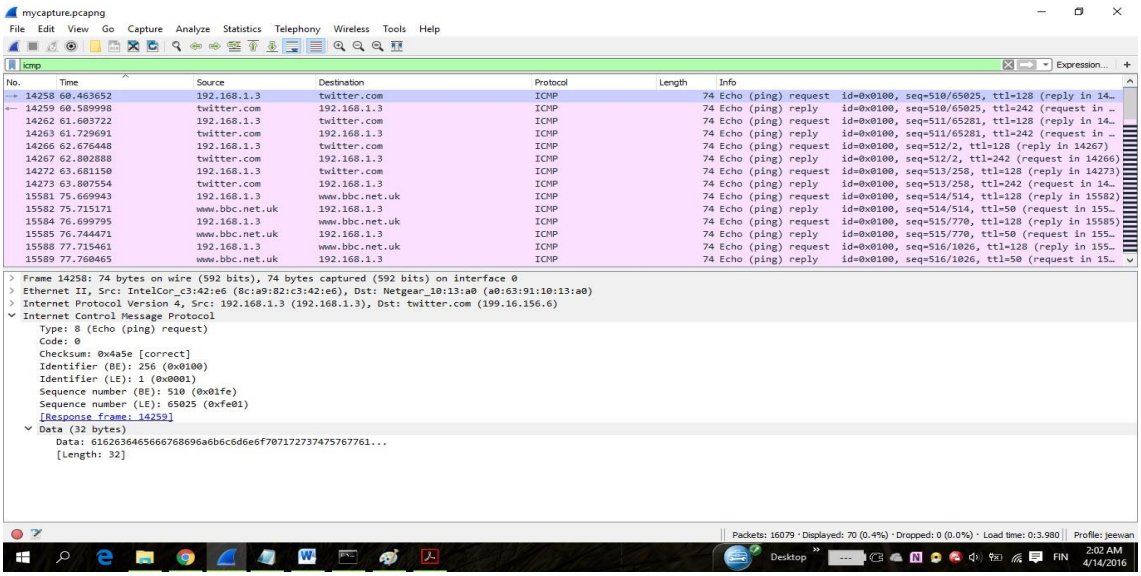

Figure 29. ICMP packets captured in Wireshark

In the figure above, the host device uses ping command to check the connectivity to www.twitter.com. In the packet number 14258 echo ping request is sent to the server and in the next packet 14259 echo ping reply is received from the server within 0.1263 seconds which proves the active connectivity. The ICMP header as shown in the packet details pane of the Wireshark contains the following fields:

- Type: This field gives the value for the type of ICMP message. For example, this value is 8 for ping request and 0 for ping reply.
- Code: This field is the sub classification of ICMP message.
- Checksum: This field checks the contents of the ICMP header and data, if they are broken upon arrival.

6.3 Network security with network analyzers

Network security is one of the major concerns of today's technical world. There are lots of tools available to increase network attacks, vulnerabilities and to penetrate the secure system. Network analyzers are one of those tools, which initially were designed to detect issues and to troubleshoot them in a network. But, hackers use network analyzers such as Wireshark to constantly investigate a network to plan and succeed in a network attack. This does not mean that network analyzers cannot help administrators

to stop attacks and troubleshoot network problems. Wireshark also helps to differentiate between normal traffic flow in the network and the unusual traffic pattern. Thus, network analyzers can be a handy tool to provide secure and smooth running system.

6.3.1 Network troubleshooting

Network troubleshooting is a way of detecting and diagnosing problems and issues inside a computer network. And network analyzers can become a good friend of administrators to keep an eye on the network, to solve issues and provide a smooth system. Some of the issues like unreachable destinations, unreachable ports, fragmented packets, network connectivity, TCP connection, and malware problems can be solved using network analyzers.

ICMP provides the facility of ping to check the connectivity between two devices. When the target device does not respond to ping packets, the destination unreachable message is received to the host side. In Wireshark, the packet details pane provides ICMP header and codes to check the status of the packet. If the packet has type 0, then it is an echo ping reply. However, for an unreachable destination a type 3 status is shown. This value helps to point out the issue of unreachable destination. Similarly, another benefit provided by network analyzers to troubleshoot issues within a network is IP fragmentation. In the packet details pane of the Wireshark, inside Internet Protocol section we can see Flags to verify if the packets are fragmented. When the amount of data is large, it is divided into small fragments and also called IP fragmentation. Flags with value 0 means that the packet is not fragmented. Network analyzers provide help to keep track of every bit of information, so no packets are lost. Similarly, TCP retransmission feature helps to keep every bit of information safe. For instance, when the receiving host receives packet with unexpected sequence number in TCP session it assumes packet is lost on the way. This data can be recovered through TCP, in which the receiving host sends three duplicate ACKs to the server. Then, the server assumes packet was lost in the transit and starts the TCP fast retransmission process. This helps to recover the lost data and keep the system smooth and safe. Another way of locating the network's performance is by solving the latency issue. Network analyzers help to calculate time delta to find out if the packets arrived late and some factors for them can be wire latency, server latency and client latency. Therefore, Wireshark can be a helpful tool to discover poor network performance. It helps to calculate time delays, packet loss and configuration faults [27].

6.3.2 Discovering malicious traffic patterns

Malicious traffic is sometimes called suspect traffic or unusual traffic pattern in the network. In other words, if some packets travel showing unusual behavior, using unknown protocols or applications, and reaches to suspicious targets gives a hint about malicious traffic. However, if someone understands how all protocols communicate and strong understanding of TCP/IP architecture can help identify network breaches. Some of the methods to understand unusual traffic are shown below [28]:

- Unknown addresses

It is important to know the range of IP addresses used in the network. This helps to identify suspicious traffic if it is travels to and from unknown address. Traffic generated from known addresses and ranges can be referred as normal traffic.

- Unknown port numbers and applications

In TCP communication, applications and protocols use well-known TCP port numbers. For instance, some of the standard port numbers are port 80 for HTTP, port number 20/21 for FTP, port 25,110 for mail services, port 53 for DNS and more. Therefore, it's important to know the standard TCP port numbers used by applications. If communication between a client and a server is running in unusual and unknown port number, that session can be suspected and more action can be taken.

- Unusual TCP traffic

During TCP communication, it starts with three way handshake and sends SYN, SYN-ACK, ACK flags to establish connection. Similarly, RST flag is send to indicate connection stoppage. And FIN, FIN-ACK flags are send to close any TCP session. They all represent a normal TCP connection. However, sometimes large number of SYN packets travel in the network which indicates someone is scanning the network and preparing for an attack. Similarly, different unusual flags used, URG flag pointed to non-existent data also hints unusual TCP traffic.

- Broadcast traffic

Some of the protocols send traffic as broadcast to all the hosts in the same subnet and the ratio should be one packet in several seconds. However, if there are hundreds of

packets travelling within a second indicate some suspicious activity in the network. Protocols such as ARP, DHCP send broadcast traffic to identify unknown hosts. In Figure 30, an example of suspicious traffic is shown where it uses nonstandard TCP port 18067 for destination.

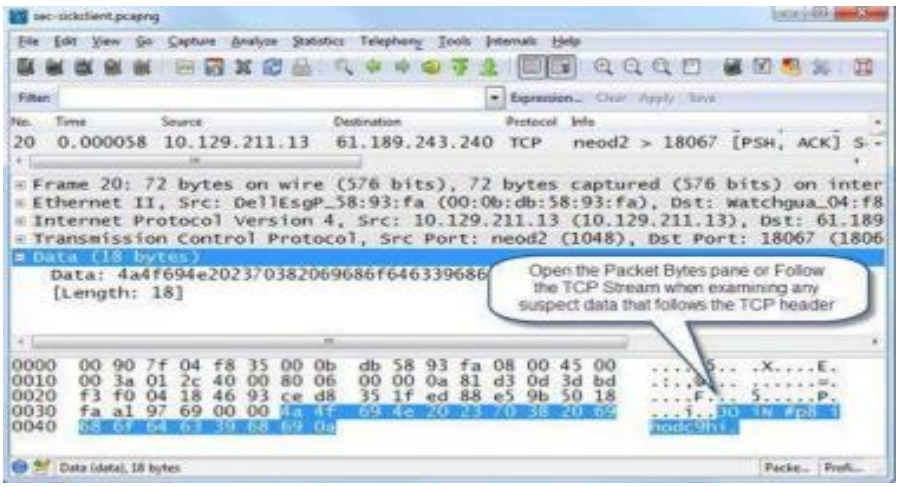

Figure 30. Traffic using nonstandard port [28]

6.3.3 Network Forensics

Network Forensics is defined as the method of studying network traffic to discover attacks and unusual behavior in the network. This process is significant to understand traffic flow and identify crimes or illegal activities happening in the network. Network forensic method can easily locate network breaches. The study of unusual traffic pattern can significantly help to plan and make network secure for the future. Network forensic can help gather evidence behind any attacks. Viruses, worms, DoS, DDoS, MITM, scanning etc. can be discovered through network forensic process. Some of the methods of discovering attacks are described below:

- Discovering ARP Poisoning

ARP poisoning attacks can be caused by scanning software and some analyzers like Cain & Abel, Ettercap etc. These attacks can be discovered using Wireshark. ARP cache poisoning attack builds platform for man-in-the-middle attacks. In order to discover this attack, Wireshark should be connected to a port in the network and broadcast traffic should be studied. ARP scanning process can also be used to discover lo-

cal devices in the network. For example, Nmap is a program that can be used to perform ARP scan which helps to discover hidden local devices. In Figure 31, duplicate IP address used in the ARP poisoning process is identified. In packets 13 and 14, the host causing this poisoning sends false MAC address information. In packet 20, Wireshark discovers that duplicate IP addresses are configured. The attacker informs that both 192.168.1.103 and 192.168.1.1 have the same MAC address i.e. 00:d0:59:aa:af:80.

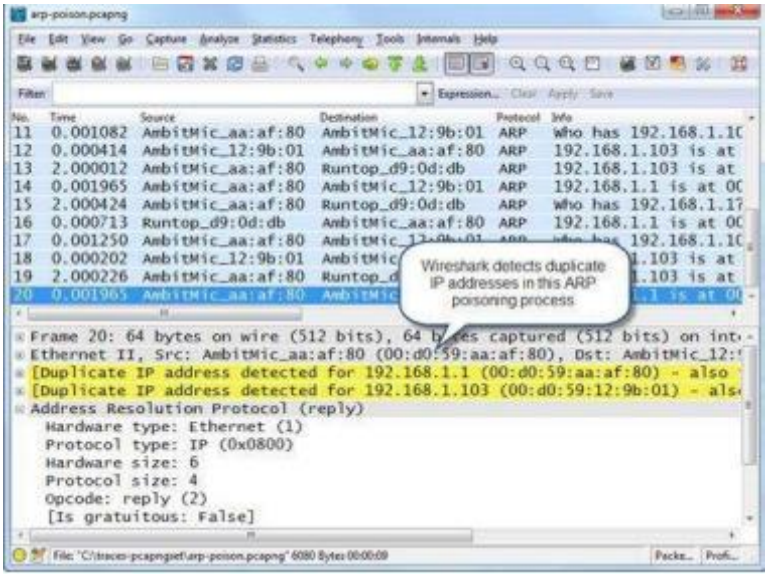

Figure 31. Duplicate IP addresses captured by Wireshark during ARP attack [28]

- ICMP scans and route redirection

Network scanning is a process of discovering active hosts in the network by sending packets, which can be for attack preparation or examining network security. ICMP ping sweeps is an example of ICMP scans. ICMP ping sweep uses ICMP Type 8 echo code and ICMP Type 0 echo reply. Similarly, route redirection is a method offered by ICMP protocol to provide best route for the packet. This redirection details can be used by attacker to cause man-in-the-middle attack. The ICMP Redirect (Type 5) packet is sent to the host by a gateway router offering a better path. For example, in Figure 32 ICMP redirect detail is shown which is sent by the gateway router 10.2.99.99 offering the best gateway to use as 10.2.99.98. After receiving this packet, the host updates its routing table with new gateway address. Therefore, when host 10.2.10.2 has to communicate

with 10.3.71.7 it should send data through new gateway address 10.2.99.98. This route redirection process can be used by an attacker to redirect traffic to suspicious destination. This can be easily detected using icmp.type==5, display filter. Similarly, when huge number of pings sweeps or ICMP packets are discovered in the network it can be caused due to a worm or some SNMP software discovering the network.

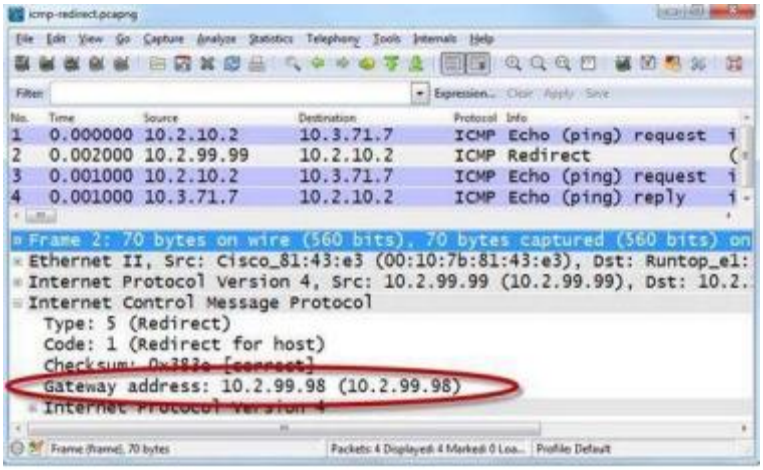

Figure 32. ICMP redirection packet showing better path [28]

- TCP scans and attacks

There are different types of TCP scanning methods used by attacker to break TCP connections and slow down the network function. Some of them are described below:

- ACK scanning

This Acknowledgment scanning can break any existing TCP connection by sending large number of ACKs in different TCP ports. This process can stop any ongoing TCP session. It is shown in the figure below [29]:

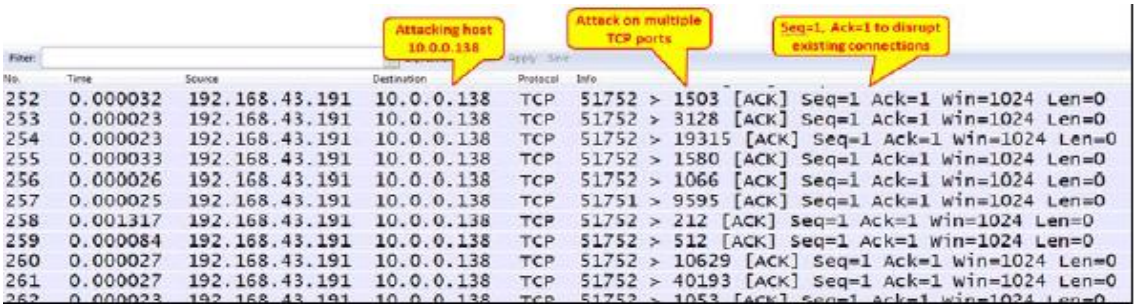

Figure 33. ACK scanning to attack TCP ports [29]

- Xmas scan

Xmas scan contains URG, FIN, and PUSH flags together. These flag sets are used to attack TCP ports and find out if they are open. This scanning is also a good preparation for an attacker to plan attack. In the figure below, these flags are seen together which indicates the ongoing Xmas scan being performed by host 192.168.43.191.

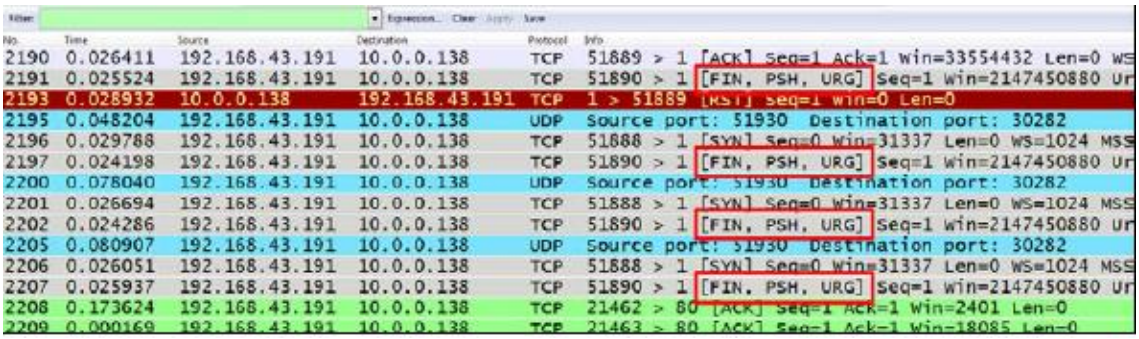

Figure 34. Xmas Scan [29]

- FIN-ACK scanning

This scanning is performed with the intention to flood the network or to close the TCP ports. When FIN and ACK flags with value 1 are sent to the TCP ports, this scanning does its job. In this scanning, packet with flag bit 1 doesn't go under TCP handshake process. And it can easily collect information on filtered, blocked, and opened ports. In the figure below, an example of FIN-ACK scanning is shown.

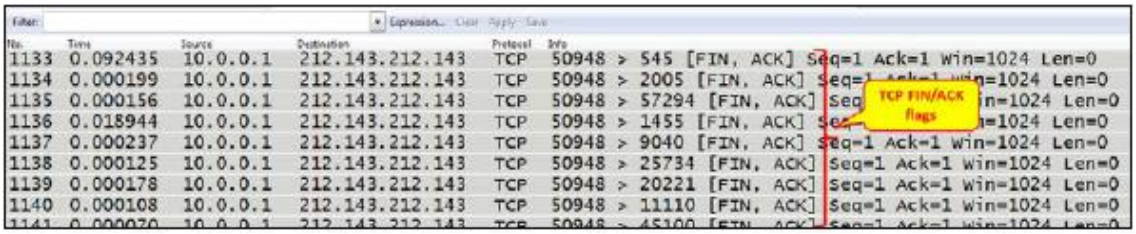

Figure 35. FIN-ACK scanning [29]

- TCP-SYN scan

In this scanning, an attacker sends large number of SYN packets and receives SYN-ACKs in response. This process can lead to half-open connections which can later lead to DoS/DDoS attacks. In the figure below, an example of TCP-SYN scan is shown.

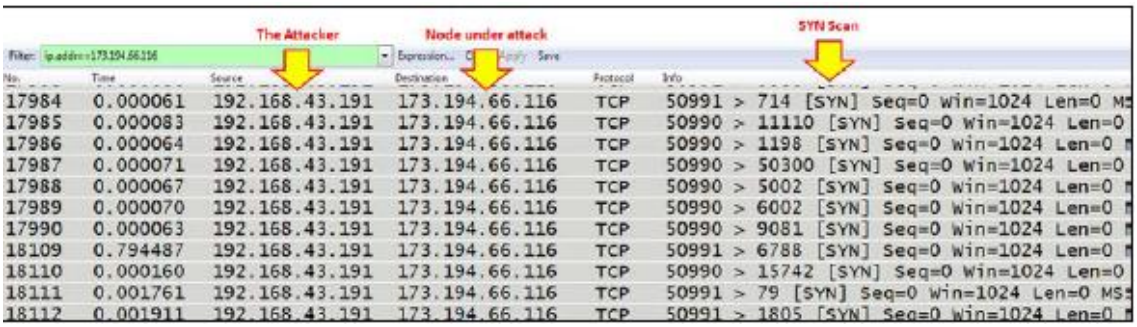

Figure 36. TCP-SYN scan [29]

In the figure above, the destination 173.194.66.116 is continuously under attack due to SYN flags. However, most of the ICMP scans, TCP SYN scans and other various types of scans can be controlled and stopped using intrusion prevention systems, intrusion detection system and firewalls in the network. This kind of protection strategy is good enough to make these scanning attacks disappear from the network.

- Password cracking attacks

There are lots of trial and error methods used in the network attacks. And these methods are used to crack passwords, to enter administrator account, and to obtain information from user directories. Some of them are brute force attack, dictionary attack,

and application attacks. Brute force attack is used to discover user passwords by using different characters, numbers, and keys. But the success doesn't come easily because it is performed on trial and error basis. Dictionary attack is also performed with the common words, names, and numbers to crack passwords. Scanning program such as Nmap is used to plan and cause application attack to break into application server. In the figure below, an example of unsuccessful attempt of an attacker to break into the account of administrator in FTP server is shown. The attacker has performed many trial and error attempts by putting passwords such as salt, aaa, abc, academia to enter the server. When this type of unusual traffic is generated in the network, coloring rule of Wireshark can be used to easily identify these attacks in the future.

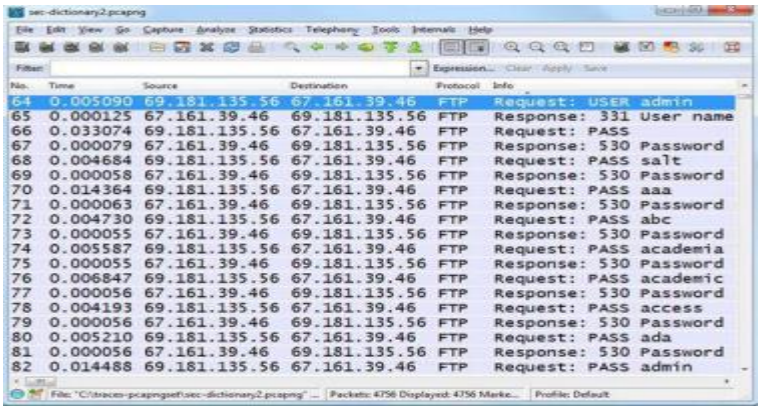

Figure 37. Unsuccessful password cracking attempt [29]

- Various systems to protect from attacks

It is really important to know the steps in cleaning an infected system. We went through the methods of differentiating between normal traffic and suspicious traffic in section 6.3.2. Similarly, we discovered some scanning attacks and other network attacks in the section above. After identifying the threats and attacks through network analyzer, we should also be aware of the methods of making network secure. The captured packets provide an idea about network activities and its behavior. The following systems can help to protect from various attacks:

- Firewalls

Firewalls restrict unauthorized traffic to enter in any specific area. Firewalls can be placed on the connection to the internet, before any servers or even personal firewalls in every personal computer.

- Network access control (NAC)

Network access control provides access only to the authorized users in the network. When unauthorized device tries to connect to the system, the link in the device goes on and off. And finally this device is restricted from Mac layer or layer 2.

- IDS/IPS

These systems can identify attack patterns and they block those devices from the network. Mostly, they are located in between the firewall and the internet. Sometimes additional software is installed in the firewall as intrusion detection and prevention system.

- Application Firewalls

These protection systems are located at layer 7. They keep an eye on the applications and if any application attack is discovered, they block those attacks.

- Web filters and mail filters

These devices keep an eye on the mail or web contents. After scanning those contents, they forward only the unrestricted traffic.

- Antivirus software

Small programs that attack and cause damage to computer systems are called viruses. However, these attacks can be controlled through antivirus software.

7 CONCLUSION

In this thesis, the basic architecture of computer networks and network analyzers were discussed. In the beginning, layered network architecture and the process of data flow in the network were studied. Next, the features of packet analyzers, their uses and some examples were described. Similarly, information security aspects of computer networks and some common network attacks performed to compromise network security of a system were introduced. Next, the installation process and useful benefits of famous network analyzer Wireshark were listed. In addition to this, protocol analysis was performed using Wireshark and at the end, methods of network troubleshooting as well as methods of discovering attacks through captured packets were presented.

In summary, a detailed analysis of network traffic can help in running a secure and fully operational network. Computer networks face a huge number of problems related to configuration issues, hardware failure, computer virus, malware, spyware and a wide range of network attacks. Network administrators have a challenging role to keep networks secure. Packet analysis can be one way of investigating and solving these problems. The better understanding of network problems and the method to solve them can be achieved through packet analysis. Packet analysis can help understand how a single bit of information travels through a network. Packet analyzers can analyze traffic, discover network attacks and also can provide methods to prevent those attacks. Therefore, a good understanding of network structure and features of packet analyzers can help solve issues related to slowness of network, improve network performance and provide a secure system.

In conclusion, this thesis quite clearly shows the basic principle behind computer network operation. The main objective of this thesis was to understand the theory behind end-to-end communication, attacks faced by end-to-end devices and to see these issues through packet analysis. It is believed that network problems can be better understood through packet level. The Internet provides huge benefits to humans but with the benefits it incurs vulnerabilities to a network. This thesis has attempted to focus on some basic concepts of packet analysis world through free and popular network analyzer, Wireshark. The ideas and concepts presented through this famous tool can do contribute greatly to discovering network crimes and to provide smooth, worry-free network. Hence, the combination of theory and practical examples shown in this thesis can be quite useful to help understand network and to make it strong and secured.

REFERENCES

[1] Coulter, C. (1997). Understanding Computer Networks. 2nd edition. [ebook] Hendersonville: Atrium Technical. Available at: http://www.netguru.net/book_ntc.shtml [Accessed: 7th April, 2016].

[2] Tcpdump.org, (2010). Tcpdump and Libpcap. [Online] Available at: http://www.tcpdump.org/ [Accessed: 3rd March, 2016].

[3] savvius.com, (2016). Omnipeek Network Analysis. [Online] Available at: https://www.savvius.com/products/overview/omnipeek_family/omnipeek_network_analysis [Accessed: 5th March, 2016].

[4] Wireshark.org, (2016). About Wireshark. [Online] Available at: https://www.wireshark.org/ [Accessed: 20th February, 2016]

[5] Peer-to-Peer. [Online] Available at: http://www.thinkuknow.org.au/site/peer-peer [Accessed: 20th April, 2016].

[6] Tanenbaum, A. and Wetherall, D. (2011). Computer Networks. 5th ed. Boston: Pearson Education.

[7] Seven-Layer OSI Model, (2016). [Online] Available at: http://encyclopedia2.thefreedictionary.com/Seven-layer+OSI+model [Accessed: 18th April, 2016].

[8] Microchip.com, (2016). TCP/IP Five Layer Software Model Overview. [Online] Available at: http://microchip.wikidot.com/tcpip:tcp-ip-five-layer-model [Accessed: 4th March, 2016].

[9] Louiewong.com, (2009). The Address Resolution Protocol. [Online] Available at: http://www.louiewong.com/archives/date/2009/05 [Accessed: 15th March, 2016].

[10] “Confidentiality, integrity, and availability (CIA triad) – Definition from Whatis.com”, [Online] Available at: http://whatis.techtarget.com/definition/Confidentiality-integrity-and-availability-CIA [Accessed: 10th March, 2016].

[11] Sanders, C (2010). Understanding Man-in-the-Middle Attacks. [Online] Windows Security. Available at: http://www.windowsecurity.com/articles-

tutorials/authentication_and_encryption/Understanding-Man-in-the-Middle-Attacks-ARP-Part1.html [Accessed: 15th April, 2016].

[12] Tountas, D (2011). ARP cache poisoning. [Online] Tournas Dimitrios. Available at: https://tournasdimitrios1.wordpress.com/2011/02/08/4426/ [Accessed: 23rd March, 2016].

[13] Popeskic, V (2011). Mac Address Flooding. (Online) How Does Internet Work. Available at:http://howdoesinternetwork.com/2011/mac-address-flooding [Accessed: 15th April, 2016].

[14] Sanders, C (2010). Understanding Man-In-The-Middle Attacks- Part2: DNS Spoofing. [Online] WindowsSecurity.com. Available at: http://www.windowsecurity.com/articlestutorials/authentication_and_encryption/Understanding-Man-in-the-Middle-Attacks-ARP-Part2.html [Accessed: 15th April, 2016].

[15] Bailey, C (2015). Warning Against MITM Attacks. [Blog] TrendLabs Security Intelligence Blog. Available at: http://blog.trendmicro.com/trendlabs-security-intelligence/extended-validation-certificates-warning-against-mitm-attacks/ [Accessed: 16th March, 2016].

[16] Sanders, C. (2011). Practical Packet Analysis. 2nd ed. [pdf] San Francisco: No Starch Press. Available at: http://repository.root-me.org/R%C3%A9seau/EN%20-%20Practical%20packet%20analysis%20-%20Wireshark.pdf. [Accessed: 10th March, 2016.

[17] Shaikh, A (2010). Top 10 Data/Packet Sniffing and Analyzer Tools for Hackers. [Online] Internet Geeks. Available at: http://www.internetgeeks.org/tech/hacking/top-10-data-packet-sniffing-analyzer-tools-hackers/ [Accessed: 25th February, 2016].

[18] Colasoft.com, (2001). Colasoft Capsa's Official Website. [Online] Available at: http://www.colasoft.com/capsa/ [Accessed: 2nd March, 2016].

[19] Gandhi, C., Suri, G., Golyan, R., Saxena, P. and Saxena, B. (2014). Packet Sniffer – A comparative Study. Journal of Computer Networks and Communications Security, [Online] Volume 2(5), pp. 179-187. Available at: http://www.ijcncs.org/published/volume2/issue5/p6_2-5.pdf [Accessed: 1st April, 2016].

[20] Eittercap.github.io, (2016). Welcome to the Ettercap project. [Online] Available at: http://ettercap.github.io/ettercap/index.html [Accessed: 15th March, 2016].

[21] KismetWireless.net, (2016). Kismet Wireless. [Online] Available at: https://www.kismetwireless.net/ [Accessed: 19th March, 2016].

[22] Monkey.org, (2000). Dsniff. [Online] Available at: https://www.monkey.org/~dugsong/dsniff/ [Accessed: 25th March, 2016].

[23] Stumbler.net, (2002). Stumbler dot net. [Online] Available at: http://www.stumbler.net/ [Accessed: 27th March, 2016].

[24] Lamping, U., Sharpe, R. and Warnicke E. (2014). Wireshark User's Guide. [Online] Available at: https://www.wireshark.org/docs/wsug_html_chunked/ [Accessed: 2nd March, 2016].

[25] Febrero, B. (2011). Traffic Analysis With Wireshark. [Online] Valencia: Inteco. Available at: http://www.csirtcv.gva.es/sites/all/files/downloads/cert_trafficwireshark.pdf [Accessed: 21st February, 2016].

[26] Kaur, A. and Saluja, M. (2014). Investigating TCP/IP, HTTP, ARP, ICMP Packets Using Wireshark. International Journal of Emerging Technology and Advanced Engineering, [Online] Volume 4(1), Available at: http://www.ijetae.com/files/Volume4Issue1/IJETAE_0114_35.pdf [Accessed: 5th April, 2016].

[27] El-Affendi, M (2010). Computer and Network Security. [Online] Available at: http://info.psu.edu.sa/psu/cis/affendi/cs391/n.html [Accessed: 2nd April, 2016].

[28] Chappell, L. (2012). Wireshark Network Analysis. 2nd ed. San Jose: Protocol Analysis Institute.

[29] Orzach, Y. (2013). Network Analysis Using Wireshark Cookbook. Birmingham: Packt Publishing.