DETECTING MAN IN THE MIDDLE ATTACK IMPLEMENTING TIME-BASED VERIFICATION METHOD

by

Syed Naziur Rahman Topu
16201109
Saif Ahmed
17101530

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
BRAC University
May 2022

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Syed Naziur Rahman Topu	Saif Ahmed
16201109	17101530

Approval

The thesis/project titled "Detecting Man in the Middle attack implementing time-based verification method" submitted by

1. Syed Naziur Rahman Topu (16201109)
2. Saif Ahmed (17101530)

Of Spring, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 26, 2022.

Examining Committee:

Supervisor:
(Member)

Mr. Arif Shakil
Lecturer
Department of Computer Science and Engineering
Brac University

Co-supervisor:
(Member)

Dr. Muhammad Iqbal Hossain
Assistant Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Robiul Alam, Ph.D
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Internet is one of the most important parts of our daily life. Network security is necessary to secure our internet network. For security purposes, we are developing different protocols, methods, and techniques. Still, those are not enough to secure the internet. Attackers are finding new loopholes to create different kinds of attacks. Man in the Middle attack is one of them. Attackers sit between a connected line and breach their privacy and steal their information. Many existing solutions have practical limitations, deployment costs as well as adversaries. To overcome this problem, we are suggesting a new program to detect the Man in the Middle attack at the beginning of the connection establishment and also alert the user about the attack with the attacker's information

Keywords: Man in the Middle, ARPspoofing, ARP poisoning, SSL stripping, MAC address, Ping time, SSL, TLS, PKA

Acknowledgement

First and foremost, we give thanks to Almighty Allah for letting us complete our thesis work on schedule throughout all the difficulties and without any serious seat-back. I would like to give my warmest thanks to my supervisor, Mr. Arif Shakil for the unwavering support and constant monitoring which made the thesis work possible. We also would like to express my sincere gratitude to my co-supervisor Dr. Muhammad Iqbal Hossain. His guidance and advice carried us through all the stages of our thesis project. We would like to give special thanks to our friends Sumya Kabir and Shaeera Islam Tasfia for their continuous support and help to complete our thesis project.

Table of Contents

List of Figures

List of Acronyms

MITM = Man in the Middle
SSL = Secure Socket Layer
TLS = Transport Layer Security
ARP = Address Resolution Protocol
HTTP = Hypertext Transfer Protocol
LAN = Local Area Network

Chapter 1

Introduction

1.1 Introduction

Security Socket Layer (SSL) is an encryption-based internet security protocol. SSL was developed by NETSCAPE in 1995 for ensuring privacy, authentication and data integrity in internet communication. SSL encrypts data to transmit across the web and users need to decrypt those data to see it. For example, if a customer visited a shopping website, placed an order and entered their credit/debit card information on the website, that information would travel across the internet openly. SSL was created to protect that credential information. SSL encrypts that information and ensured that information will be shown only who can decrypt them. SSL certificate is also known as Digital Certificate (DC) which operates based on public key infrastructure (PKI) with public-key cryptography (PKC). Nowadays, it becomes a problem that some cyber attackers can decrypt that information. A very common cyber security attack is Man in The Middle Attack. The attackers secretly relay and possibly alters the communication between two parties who believe that they are communicating with each other. In our previous example, the user and website are communicating with each other and an attacker is sitting between them to alter their information. When a website sends an encrypted message, the attacker can decrypt that message and send an altered encrypted message to the user according to his/her wish. In this case, the user has no idea that this message was sent by someone else. So, they will provide the required reply in an encrypted message which will be again decrypted by the attacker and later he will send another encrypted message to the website. There are different kinds of ways to operate MITM attacks like ARPspoofing, SSL poisoning, ARPpoisoning, DNS cache poisoning, session Hijacking, etc. MITM attackers eavesdrop on the communication between their targets and attack is difficult to detect. Some systems and solutions have been introduced to detect and prevent the attack. Several studies show that some solutions can prevent but due to several limitations and high costs sometimes they fail to prevent practically. For data encrypting and decrypting SSL protocol requires different algorithms like RSA, ECC, etc to generate keys for data encryption. Then they use Diffie Hellman key exchanger to send those keys to users for data decryption. In this paper, we propose a new time-based verification model with attackers' information for preventing MITM attacks. The rest of the paper is organized with selected related works and literature review in section 2 and a proposed model and its explanation in section 3. Finally, we provide the conclusion and future works in section 4.

1.2 Research Problem

As SSL protocol is very common internet security protocol, it is very obvious that attackers try to decrypt this protocol. Man in The Middle Attack (MITM) is the mostly common among them. Attackers enter the web when two parties are communicating with each other. According to Ankita R Chordita, Subhrajit Majumder, Ahmed Y Junaid, MITM is caused when attacker can redirect the traffic between the user and communication gate way. Most of the time attack is done by signaling out a WIFI network by using a combination of both ARP spoofing and SSL stripping. The attackers easily get access to the confidential information using SSL Stripe and SSL encryption cannot protect the confidentiality of the communication tunnel alone. In another research paper, researchers showed MITM attack based on SSL stripping using ARP spoofing. When someone use HTTP protocol, the data is transmitting as open book through the tunnel. So, it is easier for an attacker to get access to the information. To avoid the situation HTTPs is suggested to users to transmit data as it is more secure than HTTP. The problem is HTTPs is secure until it has "s". By ARP spoofing, attacker can downgrade HTTPs to HTTP and get access to their required information. During this attack browser does not show any error to the user as well.

Technically the most common methods to implement the Man in the Middle attack are ARP spoofing/ARP poisoning, SSL stripping.

ARP poisoning: ARP poisoning is a technique by which an attacker sends ARP message onto LAN to receives information from other users connected on that line. ARP table stores the ARP responses in the cache even if it does not know the sender or if it got the response for the message which it does not request. Now the attacker sends the ARP message all over the network to link its MAC address with IP address of the victim. After the MAC address of the attacker is connected to the Ip address of the legitimate user, the attacker can intercept modify or block the traffic.

SSL stripping: SSL stripping attacks are a type of cyber-attack in which hackers downgrade a web connection from the more secure HTTPS to the less secure HTTP. This makes all communications unencrypted and sets the stage for a man-in-the-middle attack, in which the hacker sits in the middle of a conversation listening or intercepting information. SSL stripping can lead to security risks like hackers eavesdropping on private information or even altering data or communications without any knowledge from legitimate users.

MITM attacks eavesdrop between two user communication line and the attack is hard to detect. Attackers can easily intercept in the communication line and gather important information. In recent times sharing valuable information through internet is very common. During pandemic, E-commerce market place has been increased at around 20 percentage. Global E-commerce sales jumped to 26.7 trillion USD just in 2019. To buy products a used need to share his confidential information like name, address, credit card details, etc. through internet. If a user or the market place is not aware enough to secure their communication line, they can be the victims of MITM attackers. To prevent this, we are suggesting an algorithm based on sending and receiving time. Our algorithm will be implemented on a server which will calculate the time sending and receiving messages. If the server finds any suspicious activity calculating time it will sends alert with the necessary information.

Chapter 2

Related Work

The paper [1] talks about the three attacks in SSL protocol and their solution. Both SSL (Secure Socket Layer) and TLS (Transport Layer Security) protocols are built for providing security to the users so that they can protect their privacy whenever they are sending data from a client-side to a web server. However, SSL and TLS both have some security holes which can cause successive attacks which is quite dangerous for both the user as well as the company. The researchers introduced three types of attacks in this paper: Cipher suite rollback attack, Version rollback attack and Password interception in both SSL and TLS channels. In a Cipher suite rollback attack, the attacker can delete the "change cipher spec" message and this will create a situation where the server and client will never upgrade their current cipher suite. One possible solution to prevent this attack is including the "change cipher spec" in the finished message's message authentication calculation but this will require a change to the SSL specification. However, this attack can be prevented by another solution which will generate a warning message for this kind of error caused by an attacker. The researcher proved that if the server supports non-RSA key exchange while working in SSL 2.0 mode, the version rollback attack will not have any effect and in this case, two sides that support both SSL 2.0 and SSL 3.0 will be forced to revert to version 2.0 protocol. Finally, one successful attack was performed by Martin Vuagnoux where he intercepted a password that was sent to an IMAP server when checking emails with MS Outlook Express 6.x using a secure connection. All these attacks can be prevented by changing the protocol itself and also other intrusion detection on the server-side is an effective way to prevent attacks from man-in-the-middle and repeated attacks. The research paper [2] is based on SSL (Secure Socket Layer) stripping attack using ARP (Address Resolution Protocol) spoofing. When someone uses HTTP communication protocol, the transmitted data has a high tendency to get sniffed by the attacker like an open book. So, it no longer stays safe and becomes completely useless at the time of using complicated websites, such as online banking. In order to avoid the above scenario, HTTPS is used to ensure that the data is being transmitted through a secure tunnel. It basically establishes a link between the web server and browser and sends encrypted information between them so that even if it gets sniffed by an attacker, it becomes useless to him. So, it can be said that HTTPS is a secure protocol as long as the 's' is stripped from HTTPS. This is known as an SSL strip attack and this SSL strip attack is downgrading the HTTPS site to an HTTP site by using many methods. The most surprising thing about this attack is the browser does not show the SSL errors to the user. The

researchers used ARP (Address Resolution Protocol) spoofing to strip HTTPS to HTTP. At the time of the SSL strip attack, the user can run ARP defenders on their PC in order to get protection against such types of attacks because the attacker must have to maintain static ARP tables to prevent an HTTPS website from stripping which is not possible for the attacker to do in all cases. This is why using static ARP tables is the best for smaller networks. However, large organizations like Google use HTTP Strict Transport Security (HSTS) protocol. So, we can say that even if this attack cannot be prevented, using ARP spoofing can keep us safe from this attack. Ankita R Chordita, Subhrajit Majumdar and Ahmed Y Javaid [3] focused on Man-in-the-Middle (MITM) attack-based hijacking of HTTP traffic using open source tools. A Man-in-the-Middle attack is caused when an attacker can redirect the traffic between the user and the communication gateway. The most common way of hijacking is done by signaling out a WIFI network by using a combination of both ARP spoofing and SSL stripping. The SSL header, as well as HTTP packets, are attached to the data being transferred for security. This is how the URL turns green and displays HTTPS. This paper gives a complete demonstration of the MITM attack over secure network connections and rerouting of all the traffic from the victim's machine towards the attacker's machine. The hackers can easily get access to confidential information by using an SSL stripe and the SSL encryption cannot protect the confidentiality of the communication tunnel alone. In order to perform this attack, the attacker and the victim need to be connected to the same local network. So, this article describes numerous weaknesses that can be exploited to achieve a MITM attack. However, this attack can be detected and prevented by SSL hijacking. The hijacking machine provides certificates to the web server and acts as a client thus, the data traffic was sent from a protected website and after that, it is provided to the user. In the research paper [4], the researchers used a trusted time-based verification model for the purpose of the automatic detection of the MITM attack in cyber-security. As with the gradual development of information technology, the risk of cyber-attacks is also increasing. Therefore, it is obviously required to protect the user's information as well as the networks and also ensure the basic security principles of confidentiality, integrity and availability. This confidentiality and integrity can be achieved by using SSL or TLS but using them contain the risk of MITM attacks. They proposed one trusted model for the automatic detection of the MITM attacks with the help of trusted time-base verification along with a novel learning-based interface scheme. One of the advantages of the proposed model is that it does not require any complex system configuration or expensive security implementations. Furthermore, their inference algorithm works in the TLS without any need to modify the current protocols which are being used. Also, the performance measures show the real-time deployment considerations for monitoring the effectiveness of the model. The main contribution of this research paper is the ability of the automatic detection of MITM attacks with the help of trusted verification of the transmission time using a learning-based interfacing algorithm and when it is used in conjunction with the existing systems which requires comprehensive configuration and network resource costs like IDS (Intrusion Detection Systems), it can provide a robust solution that addresses these practical limitations and also saves cost by providing assurance. Hitesh Mohapatra, Subhashree Rath, Subarna Panda and Ranjan Kumar [5] described handling MITM attacks in WSN through intrusion detection systems. The decentralized architecture of WSN creates loop-

holes that invite different kinds of attacks, such as MITM, black holes and so on. The researcher proposed a MITM-Intrusion Detection System (MITM-IDS) model for the detection of those attacks, isolation of the attacks and reconfiguration of attacked nodes. This proposed IDS system helps in training the nodes with the possible attacks. In the proposed model the attacker will be monitored on the basis of signature-ID templates. This model will be able to differentiate between legitimate users and fraudulent users with the help of MITM-IDS features for WSN. It is working based on a CDN (Centralized Database Network) that contains the entire IDS for WSN setup. The MITM-IDS examines the network traffic by using NIDS (Network Intrusion Detection System) as well as packet sniffer tools. The rules of NIDS can be updated on the basis of the situation and NIDS captures selective data based on the predefined rules instead of storing the total log file. With the help of the integration of both CDN and NIDS, a resource of signatures is created. The network is analyzed by the deep learning-based MITM-IDS by applying intelligent refined signature rules. The packet which is received from WSN passes through a CrowdStrike application to the WSN domain. Here, CDS controls all the switches of the network and the LSTM (Long Short-Term Memory) based deep learning approach increases the performance of CDN. Finally, we can say that the proposed model aims to develop an attack tolerant IDS. Robbi Rahim [6] described a system that will prevent man-in-the-middle attacks using the interlock protocol method. One person can easily intercept the communication medium which is used by the two individuals for communication. This process is known as a man-in-the-middle attack. In this process, all the data that is transmitted by people who are communicating with one another go through the person who tapped them in order to know all the information which is being transmitted. This kind of attack can be prevented by using interlock protocol. This protocol has a core algorithm that sends two parts of an encrypted message. Here, the first part occurs as a result of the one-way hash function of the message and the second part occurs as the encrypted message itself. So, as a result, the wiretapped person is unable to decrypt the first message with the help of using its private key. This protocol can only create a new message and send that message to the person who will receive the message. Using the interlock protocol method MITM attack can be prevented because even though the receiver and the sender's public key are acquired and replaced by the eavesdropper, he/she cannot run the MITM attack because the encrypted is divided into two parts and the delivery of messages is done gradually to make sure that the eavesdropper does not get any idea about the original message which was sent. In the research paper [7] the researchers focused on compromising password-based authentication over HTTPS. Server Authentication mode and Client Authentication mode are two types of authentication modes of SSL. In order to achieve mutual authentication in SSL communications, Server Authentication mode can be utilized with password-based authentication such as Basic Authentication or Digest Authentication implemented in HTTP. However, when password-based authentication mode is used with Server Authentication mode, both the password and its digest cannot be secured because an attacker can easily abuse the communications. However, one possible solution in the case of using SSL in Server Authentication mode is to use it in a closed system and it basically means that the clients and the servers are limited to access each other. In this case, the root certificates and the server certificates are distributed to the users and the servers directly by hand. Moreover, it can be said that the

Web browsers are removed. However, these attacks do not work when SSL is used in Client Authentication mode because when SSL is used in Client authentication mode, it can detect the attacks. Chik How, Tan and Joseph Chee Ming Teo [8] researched protection against web-based password phishing. Due to the gradual increase of phishing attacks in SSL, it becomes quite visible that SSL is not sufficient enough to protect users against fraudulent websites. This article discussed the ongoing phishing attacks as well as the limitations and the weaknesses of the current SSL protocol and it also proposed a solution to overcome those limitations and weaknesses which is an ID-based SSL protocol. The channel that is established with the help of SSL between the user and web server is used for the transmission of the authentication information. As SSL is not secure enough against web-based password theft, the attackers spoof a website by creating a fake website. Then they ask the user to click on the particular link on the website. When users click on the link, all of their personal information will directly go to the attacker's account. The current implementation of SSL depends on PKI where the CA signs a digital certificate that binds the public key of an entity to its identity and that public key of a server needs to be certified by the users for the authentication of the server. In the ID-based SSL protocol proposed in this paper, the identity information of the original web server such as its registered URL will be used for authentication and the private key of the original web server will be generated by a trusted third-party called Private Key Generator (PKG). Here, the PKG will be responsible for authenticating every web server and will not issue any fraudulent private key to any unauthorized web servers. However, the properties of this ID-based SSL protocol make it more advantageous over the traditional PKI which is used in the current SSL. Finally, it can be said that the proposed solution of this paper is able to provide secure infrastructure against phishing attacks for all types of applications. In this article [9], the main intention of the researchers is to show that a man-in-the-middle attack can be detected in SSL and TLS with the help of detecting the timing differences between a standard SSL session and an attack. In order to measure the difference between the normal SSL handshakes and the one that is conducted by the man-in-the-middle attacker, the researchers designed a program that will initiate an SSL handshake and after that, it will close the connection after it will get the certificate from a remote party. Then, it calculates the time requires to establish the TCP connection as well as the time required to receive a reply with a certificate from the remote party. Due to the presence of certain properties man-in-the-middle attacks generate certificates in real time after it is connected to the actual target, the gap between when establishing a connection and receiving a certificate has to be widened significantly under the attack condition. As the gap present between the hello of the handshake and the certificate of the server incorporates network transport time, the researchers subtract out an RTT to account for this. Then, they average the time that is required for the remote server to give a response to the several TCP SYN packets in order to estimate the RTT. In short, the researchers performed a timing analysis for determining whether the man-in-the-middle certificate generation phase can be detected in order to long response times after starting the SSL handshake or not. In the article, the authors [10] proposed a method that can detect and locate man-in-the-middle attacks between two nodes in a fixed wireless network with the help of analyzing the round-trip time as well as the measured received signal strength collected from the fixed access points. The model

is implemented to be like a client-side application and it establishes a baseline to measure the RTT (Round Trip Time) after that statistical measures are applied to RTT and RSS (Received Signal Strength) in order to detect and locate Man-in-the-middle attacks. Three machine learning algorithms: Gaussian Naive Bayes, K Nearest Neighbors and Support Vector Machine, are used to measure the RSS dataset in order to find the location of the attacker. The article [11] an outstanding process named MITTM Distributed Assessment System (MIDAS) that includes everything from pinning-in-the host to the pinning-in-the net with the help of an enabling mechanism that can be used for certificate validation as the certificates go through a provided network. All these certificates will be marked as trusted ones or untrusted ones as the outcome of cross-information that is obtained from various sources. This process will detect suspicious certificates as fast as it can and it will also initiate certain mechanisms to defend against these attacks. It will also try to cut down the intensity of the impact and collect information about the attackers. MIDAS depends on the existing network management and monitoring mechanisms in order to give a pinning-in-the-net approach that enables the host so that the host can successfully access the authenticity of the certificates that they encounter at the time of TLS interactions and it also relies on an existing group of network probes that are located in the different ingredients of the network, a distributed analyzing engine that works based on the Bayesian network as well as a reaction subsystem that uses SDN technologies. In this research paper [12], the authors proposed an efficient algorithm that can detect man-in-the-middle attacks with the help of ARP spoofing by analyzing the behavior characteristics of ARP spoofing on the access switch. The ARP protocol can be found in the data link layer and the main job of this protocol is to complete the mapping conversation between the IP address and MAC address. However, the ARP protocol does not have any mechanism for authentication and this result in giving easy access to an attacker in order to pretend like a gateway or server that the attacker wants to target for snatching the data packets of the user and successfully achieve their malicious purposes. With the help of the ARP spoofing the man-in-the-middle hides in the middle of the link established between the user as well as the destination IP address or server and to achieve this spoofing, the attacker has to do a couple of things. The first one is to mischievously make the users mistake the attacker as their designated gateway and the second one is to make the original gateway mistake the attacker as its desired user these two things can easily be done by sending the ARP response packets to both the user as well as the gateway at the same time. However, at the time when the attacker is trying to get his job done, some abnormal characteristics are found which include having two IP addresses under the interface and data passing twice between the user and the gateway port after the completion of the attacker's task. So, with the help of this man-in-the-middle attack can be determined based on ARP spoofing. For implementing SSL or TLS in the most modern web browser, CAs (Certificate Authorities) is required for ensuring trust. As per the laws of CS's government, it has to generate certificates and also sign those certificates so that with the help of these certificates intercepting as well as eavesdropping on the SSL or TLS sessions can be done. The researchers [13] designed a SignatureCheck protocol that provides the clients with all the necessary abilities in order or retrieves the certificate thumbprints of a website remotely. The entire process is done by a third-party server over a channel that is protected by the RSA signatures and these

RSA signatures are independent of the Public Key Infrastructure (PKI). After this, the user can easily compare the thumbprint which is retrieved remotely with those thumbprints that are received locally so that the user can detect the MITM attacks that have been conducted using authentic certificates or those certificates that are accepted by the browser as a result of either a bug or a hashing collision. The proposed SignatureCheck protocol is highly efficient to detect man-in-the-middle attack on SSL sessions making the protocol a lot more beneficial at the time when the client transmitting any information through the internet that is confidential.

Chapter 3

Methodology

3.1 Our Methodology

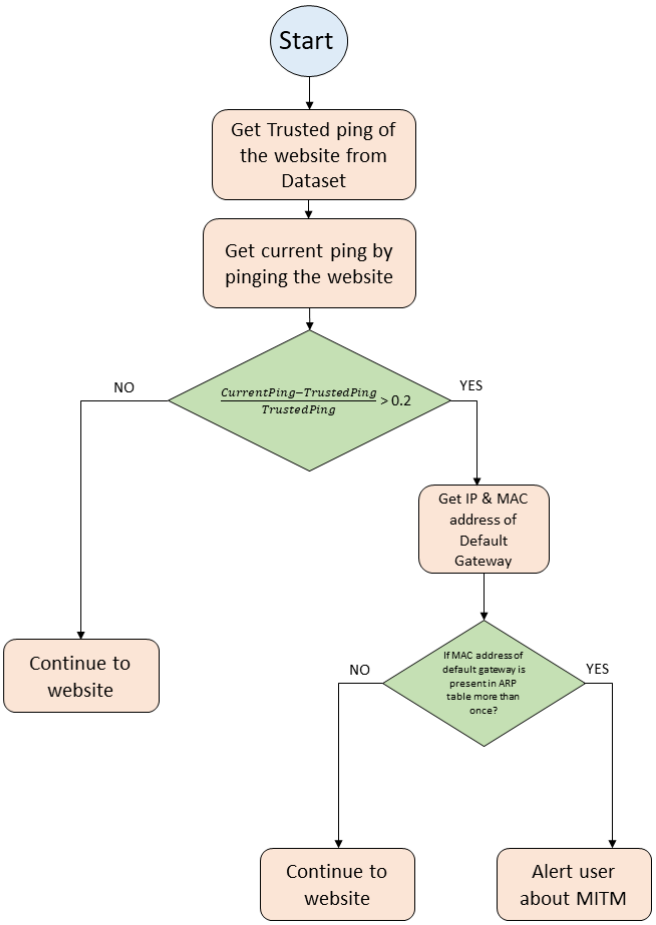

Figure 3.1: Methodology

Initially, we type the web address in the search bar of our browser. Now this searched website address will pass to our programs method where it will check the ping time of that website at that time. Then our program will compare the current ping time with our stored database trusted ping time. For our developed program, we are considering the ping time delay at most 20% because of the weather or other disturbance. If the compared time delay is less than 20% then our program will allow the user to continue the website otherwise our program will collect the IP address and MAC address of the default gateway and it will run the ARP table to verify whether the MAC address exists more than once in the ARP table. If it finds any duplicate MAC address in the ARP table, then it will send an alert message about a suspected MITM attack threat. Otherwise, it will allow the user to continue browsing the website.

In Fig 3.1, we showed the steps regarding the model. Let us explain the steps involved in our model.
Step 1: A user searches a URL in a browser.
Step 2: Browser will pass the URL to our method
Step 3: Our program ping that website and compare the ping time with our database. (If the ping time is higher than our expected value then follow step 4 else follow step 6)
Step 4: Program will check the ARP table to identify any duplicate MAC addresses. (If they find any duplicate MAC addresses then follow step 5 else follow step 6)
Step 5: Send an alert message to the user.
Step 6: Continue browsing that website.

3.2 Software

To implement MITM by ARP Poisoning we used two software. One is Ethercap and another is Wireshark.

Ethercap: Ethercap is a free and open-source tool for Man in the Middle attack on a LAN. It runs on various Unix-like operating systems including Linux, and MAC OS X. It is capable of intercepting traffic on a network segment and capturing passwords.

Wireshark: Wireshark is a free open-source packet analyzer. It is used for network troubleshooting, analysis, and software communication protocol development. It runs on Linux, MAC OS, and other Unix-like operating systems. Wireshark lets the user put network interface controllers into promiscuous mode, so they can see all the traffic visible on that interface. In our research implementation, we used Ethercap to implement a successful attack and we used Wireshark to collect data from the victim's pc

Chapter 4

Implementation

4.1 Dataset

4.1.1 Source

We used 50 websites to collect our dataset

4.1.2 Data sample

NO.	Address	Ping(ms)
1	google.com	56.00
2	daraz.com.bd	56.01
3	bux.bracu.ac.bd	254.32
4	chaldal.com	71.97
5	foodpanda.com.bd	57.84
6	aliexpress.com	182.91
7	facebook.com	64.98
8	youtube.com	49.87
9	accuweather.com	49.75
10	weather.com	61.84
11	secure.incometax.gov.bd	8.00
12	passport.gov.com	45.67
13	wikipedia.org	66.82
14	startech.com.bd	85.02
15	ucc-bd.com	16.12
16	prothomalo.com	61.99
17	jagonews24.com	49.01
18	dhakapost.com	52.01
19	kalerkantho.com	57.96
20	thedailystar.net	60.99
21	ittefaq.com.bd	72.8
22	samakal.com	53.02
23	somoynews.tv	56.04
24	ntvbd.com	53.87
25	bbc.com	71.91

Figure 4.1: Sample picture of the dataset

In figure 4.1, we can see some of the data samples of some website's ping timing

4.1.3 Data collection

For data collection, we used 50 websites. We ping those websites multiple times a day using 2 different IP addresses and store them. After collecting 500 sets of ping time data, we agreed to create this average ping time. We considered a standard environment for collecting data. There was no time delay for bad weather issues or any disturbance.

4.2 Implementation Environment

Normally when we are connected to the internet and we are browsing something on the internet the normal scenario is a device is connected with the router and the router is connected with the internet(Figure 4.2). But when the Man in the Middle attack occurring this scenario just changed. Initially the device was connected to the router but during the attack there is another device connected between the user device and the router(Figure 4.3)

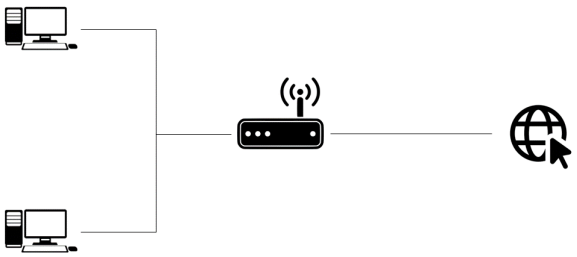

Figure 4.2: Normal connection

Figure 4.3: Man in the Middle Attack

For implementing our research work we used 2 devices. One was a Windows 10 installed PC and another was Kali Linux installed PC. We considered our Windows 10 installed pc as a victim pc and the Kali Linux installed pc as an attacker pc. Both devices were connected under the same network. For ARP poisoning, we need to establish a connection in the same network.

4.3 Attack Implementation

Initially, from the Command Prompt of our victim's pc we will get an ARP table which will show our victim's pc information.

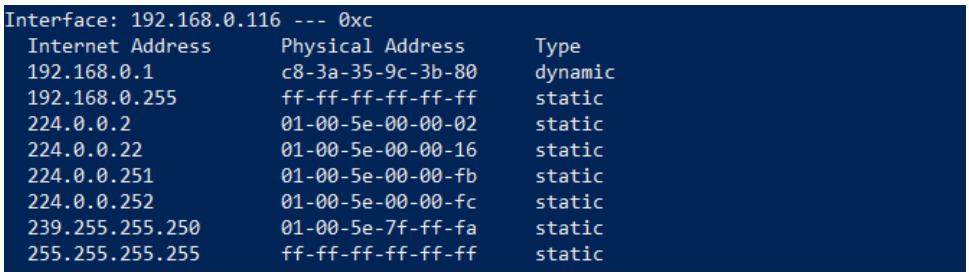

Figure 4.4: ARP Table of Victim's PC (Without Attack)

In figure 4.4, we can see the victim's pc IP address is 192.168.0.116, Default Gateway is 192.168.0.1 and Mac address of Default Gateway is c8-3a-35-9c-3b-80 in the ARP table. Then we will ping google.com from the victim's pc to see the ping time without any attack

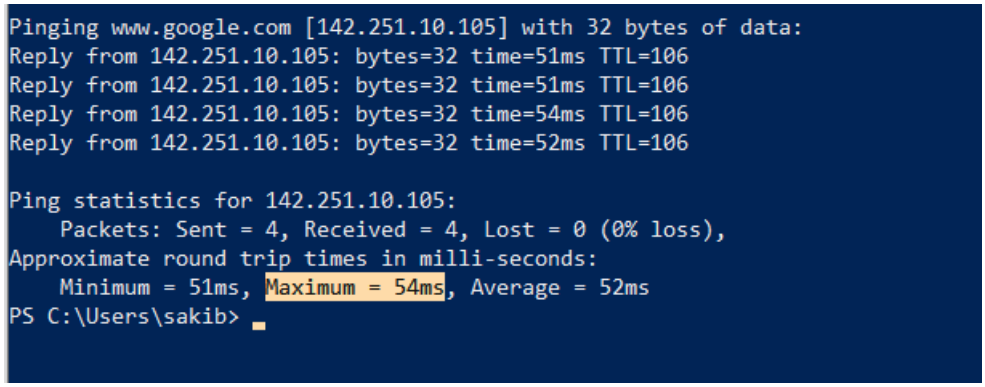

Figure 4.5: Ping time(Without Attack)

From the above figure 4.5, we can see the maximum ping time is 54ms and the average ping time is 52ms. Before initiating the attack, we will search the attacker's pc IP and mac address for further process. (Figure 4.6)

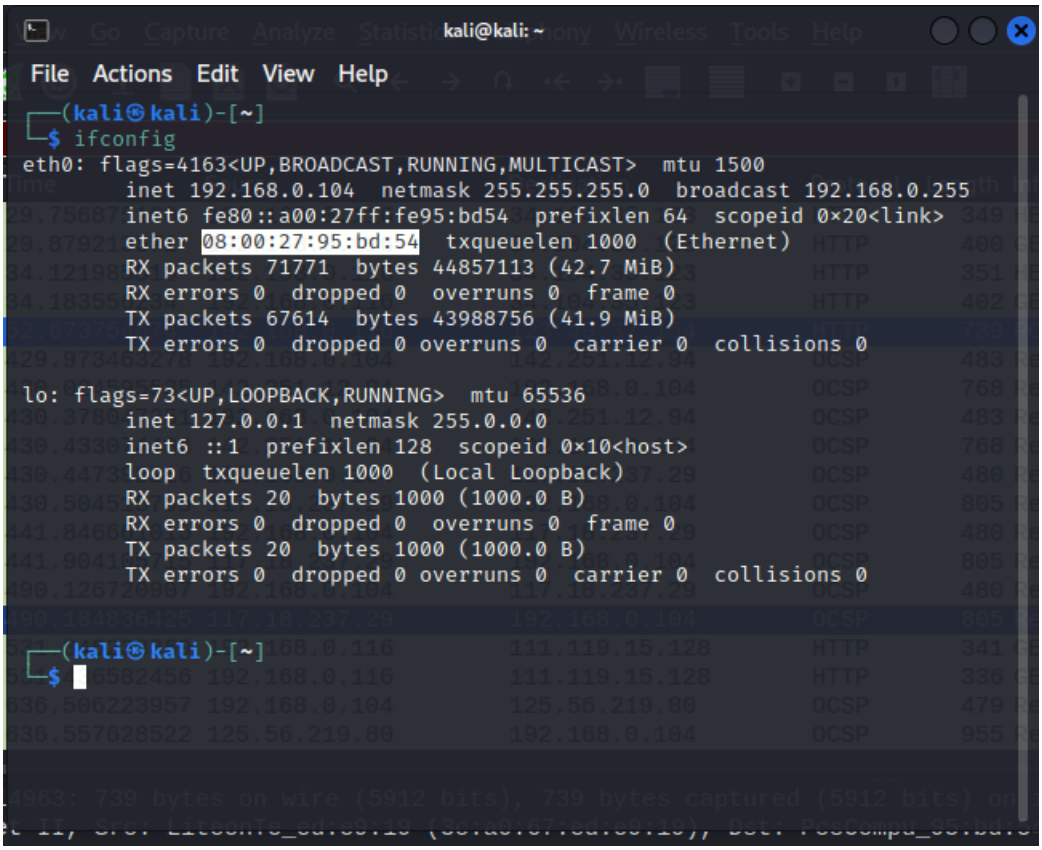

Figure 4.6: Attacker's PC information

In figure 4.6, we can see the attacker's pc ip address is 192.168.0.104 and MAC address is 08-00-27-95-bd-54.

For ARP spoofing both victim and attacker, devices have to be connected under the same network. Now, we initiate the attack using Ethercap. For the attack, unified sniffing is started in Ethercap. Then, Ethercap selects an interface in united sniffing, To select the target, it will scan all the hosts and from them, Ethercap will select the victim's pc as Target1 and then select the router as Target2. So, the victim's IP address 192.168.0.116 will be shown in Ethercap when it scans the hosts(Figure 4.7).

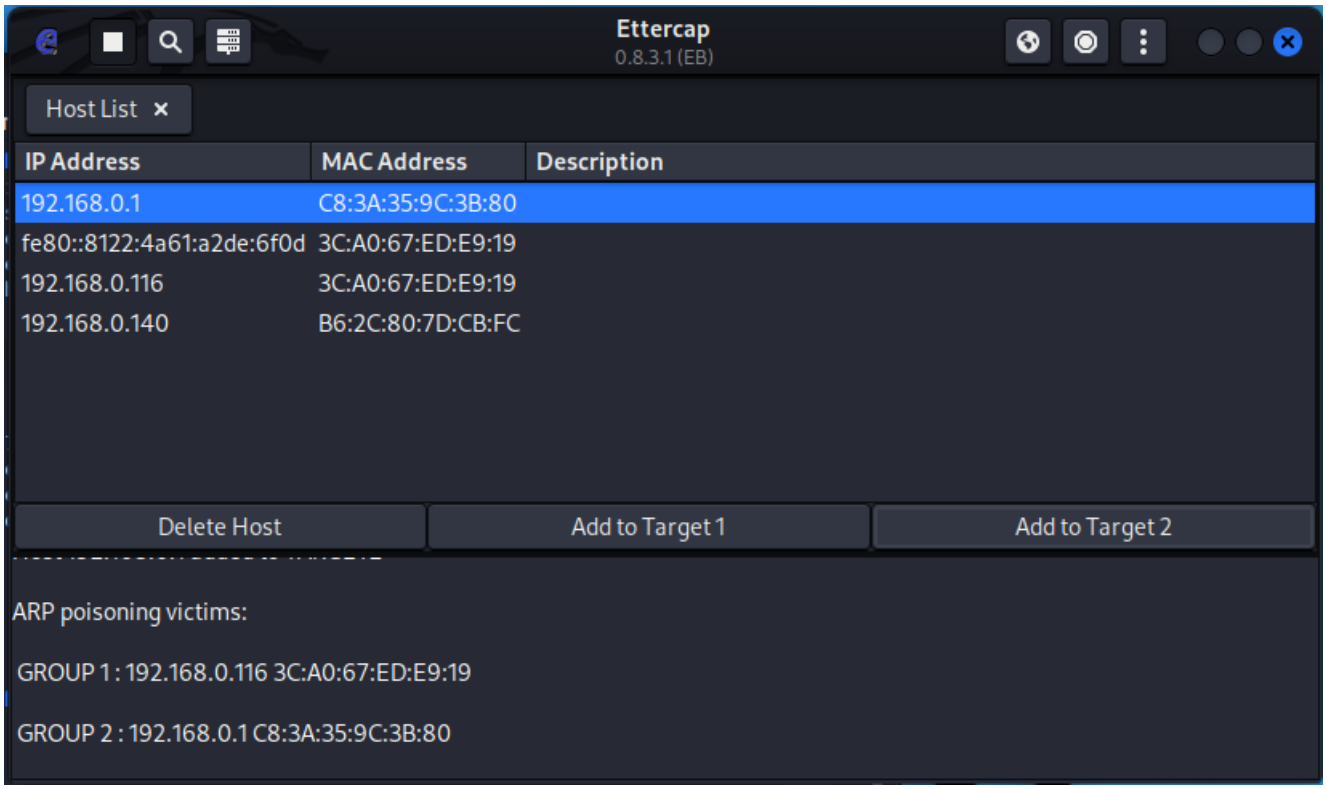

Figure 4.7: ARP poisoning the victim's PC and Router

Then MITM attack starts by selecting "ARP poisoning" in Ethercap. After initiating the attack, from the ARP table we can see the attacker's address is included in the ARP table of the victim's pc. (Figure 4.8).

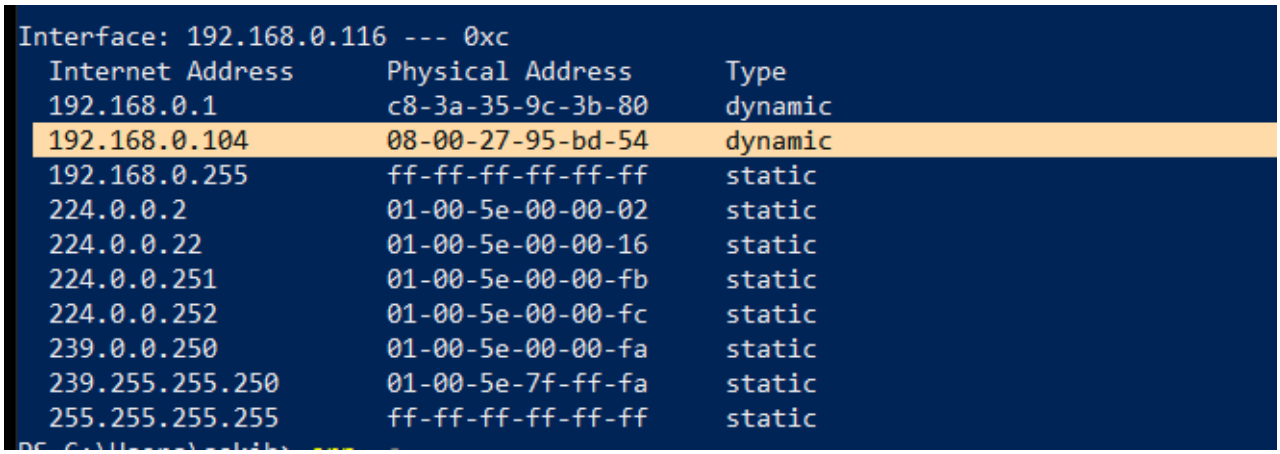

Figure 4.8: ARP table while initiating the attack

```
Interface: 192.168.0.116 --- 0xc
  Internet Address      Physical Address      Type
  192.168.0.1           08-00-27-95-bd-54     dynamic
  192.168.0.104         08-00-27-95-bd-54     dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.0.0.250           01-00-5e-00-00-fa     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Figure 4.9: ARP table after the attack

From the ARP table, the default gateway MAC address changed to the attacker's MAC address. (Figure 4.9). So, the MITM is implemented. Now if we ping google.com again, we find the maximum ping time is 88ms and the average ping time is 76ms which is much higher than our previously-stored ping times. Time Delay = After attack ping time – Before attack ping time = 76ms-52ms = 24ms For our research work, we considered a 20% time delay due to bad weather or other disturbance If the time delay is 20% higher than our expected average ping which will be compared with our dataset, then we considered it as a MITM attack.

```
PS C:\Users\sakib> ping google.com

Pinging google.com [172.217.194.138] with 32 bytes of data:
Reply from 172.217.194.138: bytes=32 time=69ms TTL=51
Reply from 172.217.194.138: bytes=32 time=88ms TTL=51
Reply from 172.217.194.138: bytes=32 time=85ms TTL=51
Reply from 172.217.194.138: bytes=32 time=62ms TTL=51

Ping statistics for 172.217.194.138:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 62ms, Maximum = 88ms, Average = 76ms
```

Figure 4.10: Ping after the attack

Now we will try to access a non-HTTPS site. For our research purpose we selected 'www.passport.gov.bd' to implement our attack. Now in the applicant id and password section we provided our Applicant id- ' 12345678 ' and password ' examplepassword '. After submitting those credentials, we will try to collect that information using Wireshark from the website (Figure 4.12)

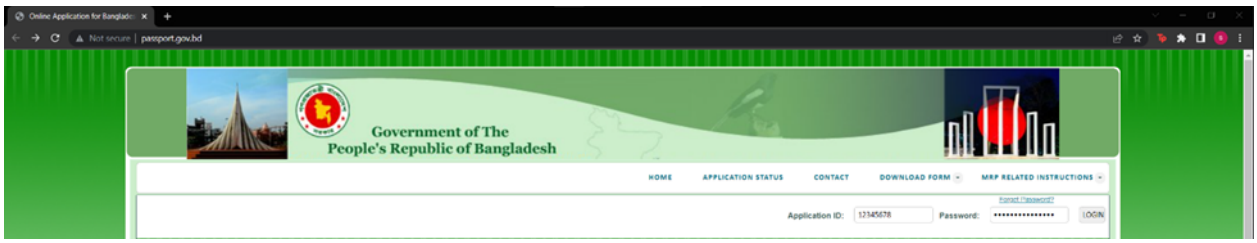

Figure 4.11: Accessing non-HTTPS site

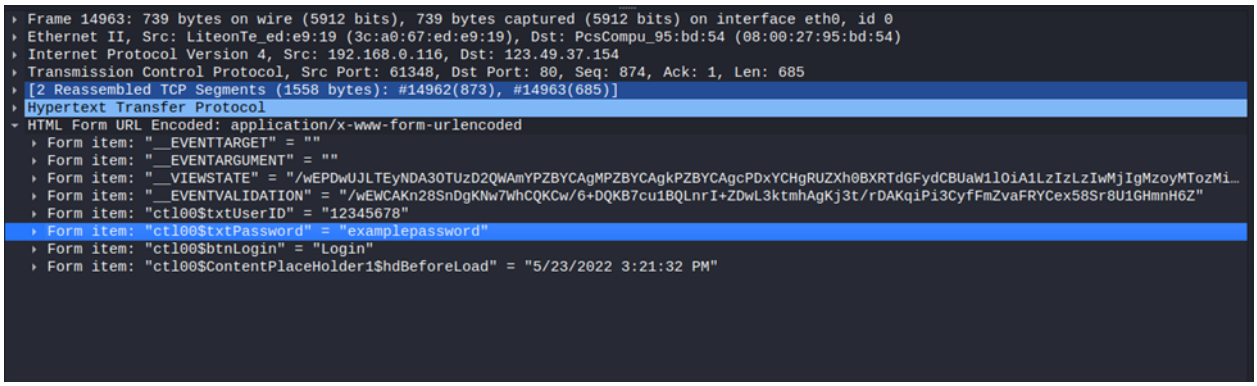

Figure 4.12: Victim's Credential being stolen using Wireshark

In Figure 4.12, we can see the used id and password have been passed to Wireshark from the website. Now we can agree that using Ethercap and Wireshark, we were able to implement the Man in the Middle attack by ARP spoofing successfully.

4.4 Detection

This Entire Man in the Middle attack was implemented on a victim's pc and the user could not understand that his privacy had been breached. To prevent this kind of situation we developed a program that will help us detect a Man in the Middle attack. We used python 3.9 and pycharm community edition 2021.3.2 for our developed program. We have used some python library such as os,pandas,netifaces and re.

When we searched the website address in a browser, that website address will pass to our program method first. Our program will ping the website to get the real-time ping time which will be compared with our database's trusted ping time. Our considered ping delay is 20%. If our program finds more than a 20% ping delay, then it will search the ARP table to find any duplicate Default Gateway MAC address. If there is any duplicate MAC address then our program will send an alert message about the suspected MITM attack. Otherwise, it will allow the user to continue on the website. Our developed program's pseudo code is given below

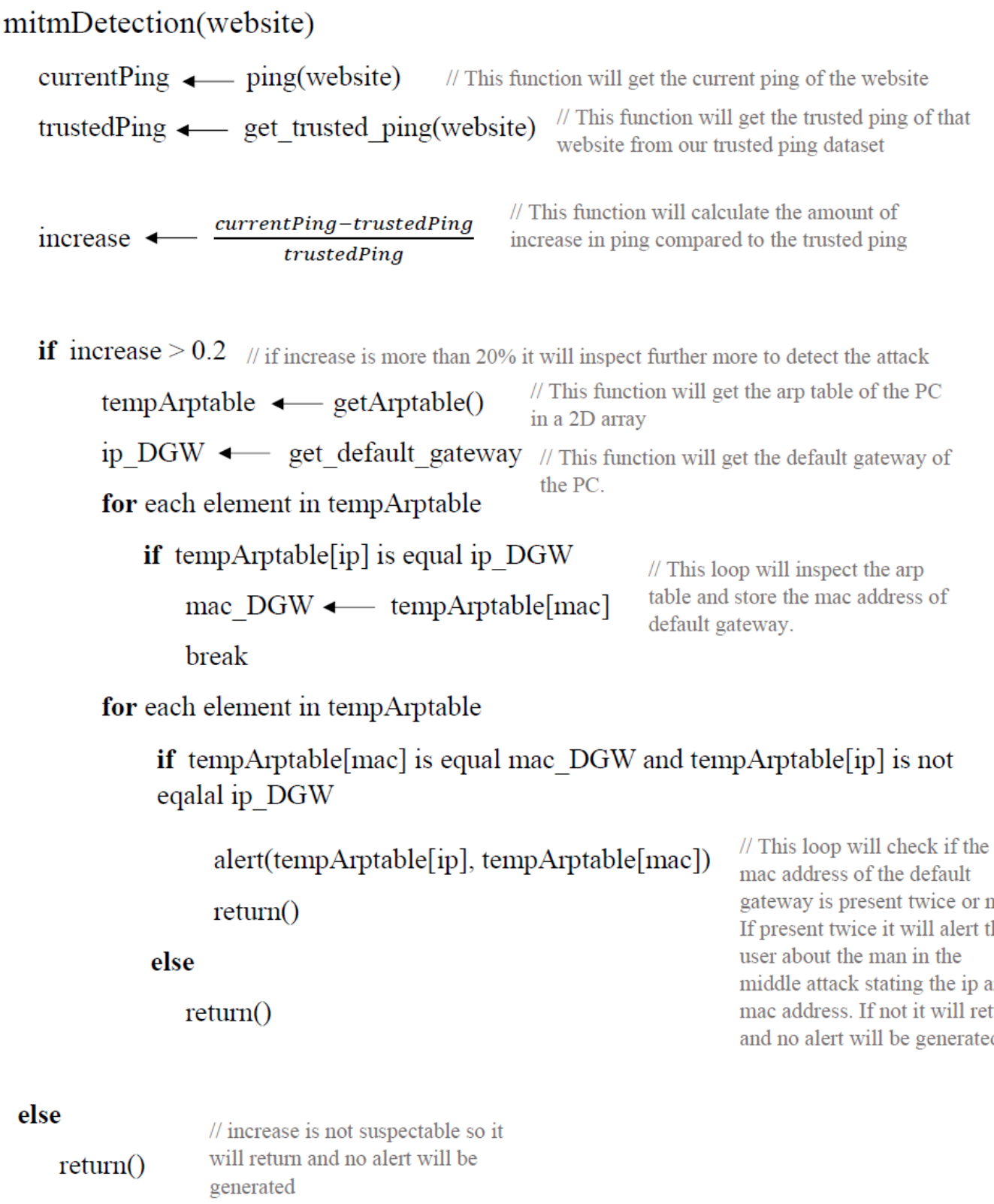

Figure 4.13: Pseudocode

Chapter 5

Results and Analysis

5.1 Result

We conducted a simulated experiment on our developed program. We tried to attack a victim's pc using Ethercap and Wireshark. When the user tried to access a website, our program was successful to detect the attack using time base verification and sending an alert message to the user.

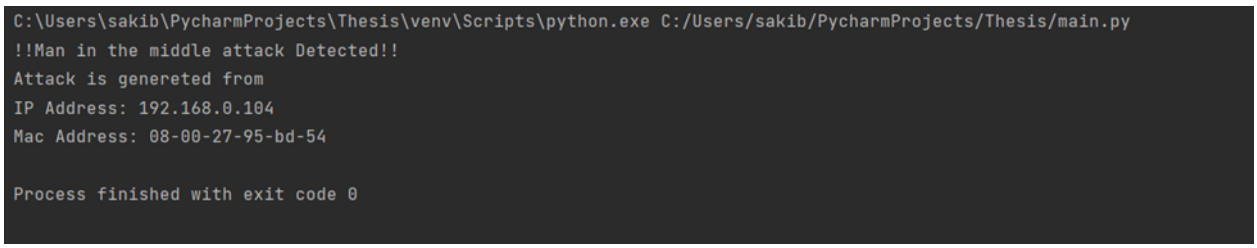

Figure 5.1: Alert message with e attacker's ID

Without the attack, the standard ping time for password.gov.com was 45.67ms. After initiating the attack, the new ping time was 59ms

$$TimeDelay = 59 - 45.67 = 14.67$$

As we are considering the time difference at most 20%, the maximum time should not be higher than (45.67+9.13) = 54.8ms. But the new ping was higher than the expected ping time. In this case, our program checked the ARP table and found multiple entries of the MAC address(Default Gateway). Our program then detected the Man in the middle attack and the alert message is being sent to the victim including ip and mac address of the suspected attacker's pc. (Figure 5.1). Our developed program could detect 17 out of 20 attacks and almost 85% successful to detect the Man in the Middle attack. Our program never detected any false Man in the Attack as our program inspect the ARP table after the ping delay higher than 20%.

Ethercap can only work on one session. In this case, the user should end the current session. He/She should turn off the router and, on the router, and later they should block the attacker's IP address from the router.

5.2 Analysis

In our research work, we focused on the ping time of the connection. To avoid error, we considered a 20%-time delay. Sometimes Ping time varies because of bad weather or huge traffics on a server. But during a Man in the Middle attack, it shows significant time delay. (Figure. 5.2)

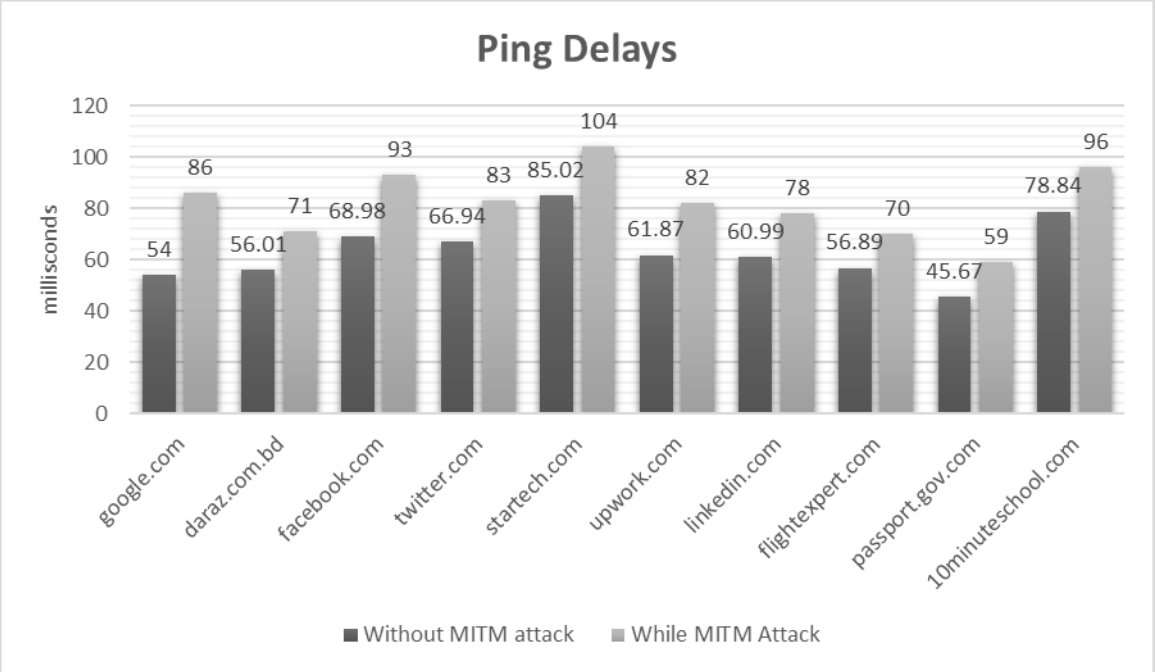

Figure 5.2: Significant ping delays during the attack.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

There are so many possible cyber-attacks on SSL protocol because of its shortcomings. Security mechanisms like HTTPS and Certificate authority are applied to decrease the risk of the attacks. However evasive MITM attack can intercept the certificate authentication process. From a user end, a user never checks the ARP table or ping time of a site when he is accessing. If someone intercepts between a user and a server, a user does not understand someone is intercepting his data. To solve this kind of situation we proposed our time-based verification method where our software automatically checks the communication line and prevent possible Man in the Middle attack.

6.2 Future work

To implement our proposed model, we used a private network. In future works, we want to implement our program on a broad network. Also, we will implement our model on servers and in a bigger network topology. proposed model will make the existing versions of security protocols very strong. The advantage of this model is that it does not require complex methods or any expensive mechanisms. Our model will not affect the existing protocol designs and will work all together to make the protocol stronger

Bibliography

[1] H. lei Zhang, "Three attacks in ssl protocol and their solutions," *Internet: https://www. cs. auckland. ac. nz/courses/compsci725s2c/archive/termpapers/725zhang. pdf [June, 2014]*, 2014.

[2] S. Duddu, C. L. Sowjanya, G. R. Rao, K. Siddabattula, *et al.*, "Secure socket layer stripping attack using address resolution protocol spoofing," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, 2020, pp. 973–978.

[3] A. R. Chordiya, S. Majumder, and A. Y. Javaid, "Man-in-the-middle (mitm) attack based hijacking of http traffic using open source tools," in *2018 IEEE International Conference on Electro/Information Technology (EIT)*, IEEE, 2018, pp. 0438–0443.

[4] J. J. Kang, K. Fahd, and S. Venkatraman, "Trusted time-based verification model for automatic man-in-the-middle attack detection in cybersecurity," *Cryptography*, vol. 2, no. 4, p. 38, 2018.

[5] H. Mohapatra, S. Rath, S. Panda, and R. Kumar, "Handling of man-in-the-middle attack in wsn through intrusion detection system," *International journal*, vol. 8, no. 5, pp. 1503–1510, 2020.

[6] R. Rahim, "Man-in-the-middle-attack prevention using interlock protocol method," *ARPN J. Eng. Appl. Sci*, vol. 12, no. 22, pp. 6483–6487, 2017.

[7] T. Saito, R. Hatsugai, and T. Kito, "On compromising password-based authentication over https," in *20th International Conference on Advanced Information Networking and Applications-Volume 1 (AINA'06)*, IEEE, vol. 1, 2006, pp. 869–874.

[8] C. H. Tan and J. C. M. Teo, "Protection againstweb-based password phishing," in *Fourth International Conference on Information Technology (ITNG'07)*, IEEE, 2007, pp. 754–759.

[9] K. Benton and T. Bross, "Timing analysis of ssl/tls man in the middle attacks," *arXiv preprint arXiv:1308.3559*, 2013.

[10] Z. C. Dong, R. Espejo, Y. Wan, and W. Zhuang, "Detecting and locating man-in-the-middle attacks in fixed wireless networks," *Journal of computing and information technology*, vol. 23, no. 4, pp. 283–293, 2015.

[11] E. d. l. Hoz de la Hoz, G. Cochrane, J. M. Moreira-Lemus, R. Paez-Reyes, I. Marsá Maestre, B. Alarcos Alcázar, *et al.*, "Detecting and defeating advanced man-in-the-middle attacks against tls," 2014.

[12] M. Ren, Y. Tian, S. Kong, D. Zhou, and D. Li, "An detection algorithm for arp man-in-the-middle attack based on data packet forwarding behavior characteristics," in *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, IEEE, 2020, pp. 1599–1604.

[13] K. Benton, J. Jo, and Y. Kim, "Signaturecheck: A protocol to detect man-in-the-middle attack in ssl," in *Proceedings of the seventh annual workshop on cyber security and information intelligence research*, 2011, pp. 1–1.