National College of Ireland

Enhance MITM attack detection with response time in Secure web communication

MSc Research Project

Cybersecurity

Hari Haran Rajendran

Student ID: X21156077

School of Computing

National College of Ireland

Supervisor: Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Hari Haran Rajendran
Student ID:	X21156077
Programme:	Cybersecurity **Year:** 2022
Module:	Internship
Supervisor:	Imran Khan
Submission Due Date:	15th December 2022
Project Title:	Enhance MITM attack detection with response time in Secure web communication
Word count 6431	
Page count: 23	

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ..... R. Hari Haran .....

Date: .....12/15/2022.....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Contents

Table of Figures

Enhance MITM attack detection with response time in secure web communication

Hari Haran Rajendran
X21156077

Abstract

The data encryption technique known as Transport Layer Security is used to establish a private connection between a client and a web service (TLS). The handshake protocol constructs an encrypted tunnel, enabling secure communication between two parties. The looming threat of Man-in-the-Middle attacks has emphasized how extremely vulnerable online data connections and financial activity are, despite the supposed security that the TLS protocol offers. This research compared the variances in time responses between an attack and a conventional SSL session in order to demonstrate the survivability of lowering the imminent threat that attacks like the man-in-the-middle posture to the SSL and TLS communication. The results of this research have been used to demonstrate how well an optimised SSL/TLS MITM detection system has been enforced.

Keywords: MITM, Response time, SSL/TLS handshake.

1 Introduction

Network privacy is enabled by the data encryption Secure Sockets Layer and Transport Layer Security. (Dierks & Rescorla, 2006). The secure session is established between the client and the server during the handshake protocol, and messages are exchanged during the record protocol. Even though the SSL/TLS protocol's main objective is to create secure sessions, the session is insecure due to several concerns. For instance, a session could be easily intercepted in the middle by the attacker, who could then pose as the client while being the server. A Man-In-The-Middle (MITM) attack is possible because of client and server cannot notify whom they are communicating to. [1]According to OWASP, MITM is being used not only as a strategy but also as an assessment framework when creating websites and applications, or it could be explained that it is used to assess online vulnerabilities. A cyberattack known as a man-in-the-middle occurs when two parties employ a transmission technique that utilizes shared or public keys as a security measure.

In essentially, cryptography codes are used to prevent the loss and duplicity, public key encryption is employed to confirm client and server uniqueness, and symmetric encryption algorithm is used to protect messages sent across the connection (MAC). A certificate is used by the SSL/TLS protocol to thwart MITM attacks. The trustworthy third party, also referred to as the Certificate Authority, issues the certificate and ensures the identity of the server (CA). The SSL certificate is insufficient for detecting and preventing MITM attacks due to a vulnerable CA, a fake certificate, and the certificate's expiry. Even so, the relatively short board of the bucket, which includes things like X.509 certificates, validating codes, client implementation, and so forth, is what ultimately determines the level of SSL security(Gerretzen, 2022). By taking advantage of these vulnerabilities, a hacker can simulate each destination of a connection utilising ARP spoofing, proxy redirection, as well as other computer hackers tools, resulting in an MITM assault. After hearing the key exchange between the two legitimate transacting entities, the hacker goes on to establish two separate connections with them using new keys. If the breach happens, the attacker will be able to see, duplicate, and potentially tweak all information being transmitted before giving them to the intended recipient by using the false key that was previously mentioned.

The fictitious SSL communication equivalent of the preceding is as follows:

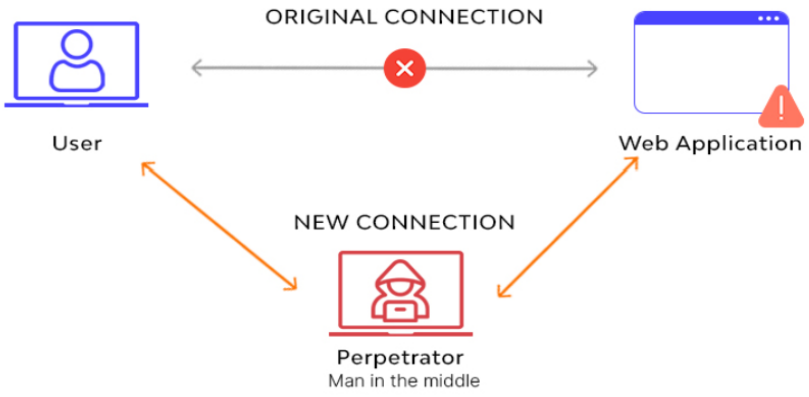

Figure 1: SSL communication during MITM.

[1] https://www.owasp.org/index. php/Man-in-the-middle_attack

• In this instance, the user starts an SSL handshake for a web application without recognizing that an attacker (Man-in-the-Middle) has taken control of the connection.

• The attacker starts a unique SSL session with the website.

• After starting the SSL session with the web application, the attacker creates and sends a certificate that would seem to be from the user to the website.

• If the user accepts this certificate, he or she will have unknowingly started an SSL session using the attacker's key.

• The hacker can now use the SSL session he previously established with him to decrypt all messages sent by the client and send them to the web server.

• Hackers can now view or modify plain text data while transmitting communications over SSL sessions without the user or web application noticing.

The attack is made possible because the user in the hypothetical scenario accepted the hacker's certificate after the hacker infiltrated the web application. This study focuses on analyzing MITM attacks for Web communication from the standpoint of network monitoring to see if any communication protocol discrepancies that could suggest MITM attack are present ever occur. A time difference analysis was conducted to determine whether a slow reply time at the start of the Signing process might allow the suspect's phase of certificate creation to be identified.

1.1 Research Question

What can be done to improve the detection of Man-in-the-Middle attacks during secure encryption communication between clients and web applications? To prevent false positives and boost accuracy, how can one determine the time response for a typical SSL communication while the MITM attacks and feed security detection systems?

2 Related works

Various identification methods have been suggested by research groups in past studies to identify MITM attacks. This portion will put this previous research into perspective and focus on focusing the key elements that were used by in all of the cases mentioned, as well as the remarkable technical progress that makes the possibility of MITM attack detection a possibility.

2.1 Literature Review

This (Folarin, n.d.) to lessen the unacceptable risk that harm like the MITM pose to the TCP and SSL hand shank, the research used analytics to understand discrepancies and other potential behavioral variances between a model predictive and a standard SSL session. After the time required to complete a TCP and SSL communication has been completely collected, the mean round trip time for each web service is calculated using the measurements gathered during the simulated MITM attack and those obtained from the developed program. The program is then created in Python, and this script interprets information from an excel file that has been used to store the output from Cain and Abel as well as the mean Response time derived during MITM attack, and the mean response time obtained when there wasn't MITM. The outcomes of this research have been used to demonstrate how well the MITM detection system for web service has been proven true.

2.2 MITM attack detection

One of the most common tactics used in computer-based eavesdropping is the Man-In-The-Middle (MITM) attack. MITM attacks could indeed successfully launch Denial of service (DoS), DNS spoofing, and port snatching attacks. MITM attacks are especially effective in LAN environments, where they are traditionally carried out via ARP poisoning. MITM attacks of all types have a variety of potential dangers for customers, such as attempting to steal online account user ids and login details, snatching local ftp ids, ssh or telnet sessions, and so on. This study focuses on various forms of MITM attacks, their implications, and possible solutions under different conditions, giving customers the choice of one of several solutions. (Nath Nayak & Ghosh Samaddar, 2010) propose employing ARP request packet to the network's gateway, tracking the arp-a table database, using static entry in the Layer-3 switch or in the gateway, using static entry in the victim's machine, and limiting ICMP packet. To secure transactions such as money transfers, online payments, and e-commerce, secure internet sites typically use HTTPS connections. Although SSL and HTTPS protocols significantly improve the security of Online transactions, they still have some flaws that can be abused to launch an assault such as the MITM attack. This study demonstrated that an MITM threat against SSL and HTTPS interactions is feasible; transactions can be broken into, and encrypted credentials can be displayed in plain text. The proposed algorithm (DepMAC-IP) by (Ahmed & Hazza, 2010) may mitigate and protect such attacks. The resilience of this algorithm is that it checks the ARP cache to see if the ARP table has been poisoned before two parties connect. The other major advantage of this algorithm is that it uses DepMac-IP to assign IP addresses to nodes in a distinctive manner, potentially resulting in a completely secure LAN. Many key reinstallation breaches, one of the more recent WPA2 attacks, call for a man-in-the-middle (MitM) position to be established between the customer and Access Point (AP). Those who all

specifically use a multi-channel method to find the MitM role. By duplicating every frame that a real AP sends to a separate channel, the threat pretends to be a valid AP in this approach. While doing so, the malicious node impersonates a legitimate user by mimicking all frames sent to the channel of the real AP by the customer. The malicious user can consistently exploit (encrypted) traffic by duplicating frames between the two channels. To stop multichannel MitM threats while maintaining functionality with older devices, an extension to the 802.11 standard was suggested by (Vanhoef et al., 2018). The rest of the MitM attacks are weaker and only feasible under certain circumstances, even though our extension does not completely prevent all MitM different variations. Consequently, it is much more difficult to exploit current vulnerabilities in secured Wi-Fi networks when using this extension. Whilst also communication-based train control (CBTC) systems are essential for the effective and constant of urban rail transits, their high rate of network penetration makes them vulnerable to Man-in-the-Middle (MitM) breaches. (Y. Li et al., 2021) suggested a novel EI-enabled CBTC system as a means of reducing the effects of MitM on the CBTC system. The detection and defense stages of a cross-layer defense strategy were presented. Long Short-Term Memory and Support Vector Machine based detection technique for the cross-layer detection phase to combine the detection probability estimated from the train control factors series and activity log files. To determine the best defense strategy against MitM attacks for the cross-layer defense stage, they built a Bayesian game-based defense design. The precision of the throughput distribution can further increase the effectiveness of the defensive system scheme, which in all simulation scenarios outperformed the conventional defense method.

(Brooks & Yang, 2015) investigated the software-defined networking security flaws in this study. In this research, they looked closely at OpenDayLight SDN controller's security flaws. A man-in-the-middle threat using ARP poisoning was successfully deployed to exploit the traffic between a client and the OpenDayLight controller out of all the potential attack vectors. There aren't many effective ways to prevent ARP poisoning and doing so can be very challenging. On the controller device, one approach is to statically define the gateway's ARP value. However, because prevention strategies were not examined in this article or study, it is impossible to assess the efficacy of this method in preventing the MitM attack that was conducted here. A serious problem that SDN developers are advised to address in the upcoming months and years is controller vulnerabilities. The existing wireless link-based source identification, which categorizes packet sources based on physical-layer link signatures, is subject to novel attack scenarios. Given that it is typically more difficult to alter or forge, a link signature is thought to be a more trustworthy signal than an IP or MAC address for determining the origin of a packet. Therefore, it is anticipated that it will be used as potential verification against DoS and impersonation threats. Analog MITM, a novel attack model aimed at link-based source identification, was suggested and assessed by (Tung et al., 2016). AMITM can create 90% more falsified packets that are classified as being sent from authorized transmitters, according to an assessment utilizing commercial APs. Also suggested is a defense against AMITM that doesn't change the transmitter's design or the wireless protocol as it is currently used. The Internet of Things (IoT) is a network of constrained devices that do sensing work. Each device has a distinct address that serves as a means of identification, and it also employs the Constrained Application Protocol (CoAP), one of the primary web transfer protocols. CoAP makes use of UDP as a transfer protocol and DTLS as a secure platform. As a result, CoAP attacks could be attributed to threats on UDP or DTLS. According to Request for Comments (RFC) 7252, which includes threats like Sniffing, Spoofing, Denial of Service (DoS), Hijacking, Cross-Protocol threats, and other threats like Replay attacks and Relay attacks, Man-In-The-Middle attack is one of the most major security issues in CoAP. (Arvind & Narayanan, 2019)To start an active interception between the client and the server, a proxy system was set up on the client side. The project will be improved further to offer remedies to lessen these assaults.

2.3 SSL and Time response Analysis

In this 2009 study, (Aziz & Hamilton, 2009) looked at a class of protocols in use by mobile systems for timely attack detection, such as wireless sensor networks, and concluded that these protocols have defects that can be abused by the same kind of cyber-attacks that MiTM assaults belong to. Without having to fully simulate the entire circuit, static timing analysis is a computation technique used to determine how long a digital circuit will take to carry out a process. All highly complex circuits with high performance have the characteristic of utilizing a timing frequency for an effective framework. This study suggests a static analysis algorithm that uses exact timing to detect man-in-the-middle attacks on mobile processes. The timestamps of TCP packet headers are used in this paper to suggest a technique for identifying man-in-the-middle threats. The latency can be determined by calculating from these timestamps, and it is able to ascertain if the packets have abnormally long waits by making a comparison of the delays in the tcp connections to data collected from previous sessions. In (Vallivaara et al., 2014) brief test case, we demonstrate how to identify and configure a threshold parameter that reliably and with little chance of false positives identifies man-in-the-middle threats. As a result, it can be applied as a straightforward preventative measure against hacking attempts. The technique is only applicable to non-mobile systems currently, in which the differences in the delay are consistent and relatively small. TLS and other dependable protocols are used by networks to reduce risks posed by various hacking attempts, such as the MiTM. The interaction of public keys is authenticated by TLS using a key exchange protocol. Contrary to common notion, Chaum's protocol, a different method for detecting MitM assaults, does not depend on the accuracy of the tradable public keys. There are three stages in Chaum's protocol's operation. Two entities that are interacting swap public keys in the first phase, and then both sides start creating random numbers in the second. The first entity ties itself to the newly established string using cryptography after obtaining the string from the second entity and sends it to the second entity. To verify that each side has both the necessary strings, the third stage requires both communication sides to use four arbitrary "scenarios". In the incident of a MiTM assault, the protocol requires the MitM to ensure that both sender and receiver have distinct pairings of strings. (Sherman et al., 2017) performed a test to demonstrate the usefulness of the third scenario by identifying a text messaging MitM abuse using timing analysis. They pointed out that the protocol forces the man in the middle (MitM) into a situation where he has no choice but to create latencies that are evident and can be used by communication entities to ascertain the man's existence. In contrast to Interlock, Zfone, and other comparable protocols, Chaum's protocol protects against attacks that are more powerful. But nevertheless, because no natural external factors were considered during the research, there is a high possibility of false positive diagnosis. To achieve more precise findings that can be generally recognized, research should be done by examining the various delays to the hackers and creating statistical interpretations of the boundaries used in the identification of a threat. To master both data breaches and protection, a complete knowledge of network activity norms and anomalies is required. (Crume, 2008) gave a comprehensive report on a technique he created for identifying man-in-the-middle attacks. The study claims that the in-topic technology comprises three distinct parts: a technique, an activity monitoring system, and a software suite for detecting server-based man-in-the-middle attacks. He illustrates that the event monitoring system, one part of his system, records all IP addresses and UserIDs interacting with the website as well as the timestamp of each conversation. The author gave a similar illustration of how a previously unidentified IP exceeding a limit, such as "UserID amount within a specific time period," could prompt an immediate investigation or, better yet, the fully automated launch of defense. Some intriguing features of the system and security plan are shown above, such as the capability to adapt the boundary to the risk tolerance levels of various firms and organizations, optimize detection, and means to mitigate that can reduce server damage. In addition to web spoofing/phishing attacks, the procedure can be enhanced to include more MiTM possibilities. The ease of use of MitM makes it a common attack vector. Current techniques, however, either lack portability, have a high rate of false positives, or are not generic enough.

(Mirsky et al., 2018) suggest Vesper, a brand-new plug-and-play MitM detector for local area networks, in this publication. In the field of acoustic signal computation, impulse response analysis serves as inspiration for the method used by Vesper. Like how echos in a cave could indeed model the structure and shape of the surrounding area, a brief and severe pulse of ICMP echo requests can model the connection between two nodes. To simulate the typical patterns of the echoed pulses and recognize when the conditions change, Vesper utilizes neural networks known as autoencoders. By employing this method, Vesper can identify MitM attacks with high precision and little network overhead. They test Vesper on LANs that have servers, PC workstations, and security cameras. They also investigate several potential combative attacks on Vesper and show how Vesper defends against them. In this study, (Z. Li et al., 2020) introduced a brand-new two-step method for spotting potential SSL MITM attacks in the wild. They discovered suspicious hosts by using the Censys "untrusted" tag, rescanned them with targeted website domains, retrieved the associated untrusted certificates, and then looked to see if they were MITM attacking sites in the open. They discovered 322,831 forgeries, the majority of which are produced by corporate firewalls and TLS proxies like Lancom System, Fortinet, Technicolor, Ubiquiti, and others. These forgeries that have been made public may have been caused primarily by default settings. Finally, they explain the constraints of their approach and offer advice on how to guard against SSL MITM attacks. The data encryption structure of the SSL digital authentication signature system is designed to raise the level of security and privacy. To tackle the issue of low cyclic shift key coding and decoding rate and solid sensitivity, a data encryption technology of SSL digital authentication signature system based on privacy protection is recommended. The floating-point data of the SSL digital authentication signature system are first arithmetic encoded, and the random linear encryption key of the system is generated using Logistics chaotic linear mapping. Finally, the effectiveness of encryption is assessed using data simulation analysis. (Dun-Yi, 2020) Their findings show that the reliability of data encryption using the SSL digital authentication signature system has risen, the risk of data decryption has decreased, and the efficiency of secure data transfer has enhanced. (Jonas et al., 2019) This study indicates an intelligent system to guard against SSL Stripping-based session hijacking attacks. Security and usability must be carefully balanced in the system's design. A strong solution against SSL Stripping is created by considering typical user activity to security warnings and combining it with very well statistical and machine learning techniques. Depending on the significance of each webpage from a security perspective, users are presented with warning messages at varying levels. A central database server collects and combines user responses to alerts to produce a modified and enhanced rating scheme for websites. Users are protected and educated by the system without experiencing needless annoyance.

During this survey, discussed various detection techniques and their limitations against various sophisticated MiTM attacks. We concluded that these issues could be resolved by combining SSL handshake and its timing response. Section 3 of this research work further explains how MitM simulation deployed, and time response analysis have both been implemented utilizing python coding.

3 Research Methodology

3.1 Objective

When this research was finished, the obtained results were used to determine whether a detection system that accurately highlights the occurrence of MiTM threats has been developed. An evaluation of the proportion of derived false positives was used to calculate the recommended system's detection accuracy. The main goal of this research project is to show that a system for discovering MiTM attacks can be implemented using a parameter SSL handshake's time response, and the experiment's recorded data is analyzed using python coding.

3.2 Overview of methodology

Our suggested solution was implemented using Python to determine the existence of a man in the middle attack by looking at how long it typically takes to complete an SSL/TLS handshake and by looking at the typical traffic patterns estimated throughout this process. Data extracted from the simulation of a MiTM attack using the setup of Kali Linux, Ettercap, and MitM proxy tools was compared to data collected from the Wireshark records captured on the client machine during this investigation. This solution for SSL/TLS man-in-the-middle attack detection issues was made possible in large part by research from Samuel Folarin, Gopi Nath Nayak, and Gopi Nath Nayak.

3.3 Detection features and Application

The deciding factors/features used in arriving at a conclusion about the existence of the attack have a significant impact on the rate of success (or lack thereof) and accuracy of a man-in-the-middle attack detection system. A client that initiates an SSL handshake and terminates the information exchange upon receiving a certificate from the remote entity was created to compare how long it typically takes to complete one to a MiTM attack. The client computer's Wireshark software recorded the timestamps for the creation of a TCP interaction and the time it took to validate the secure socket layer information received from the remote entity. There is typically a significant difference between the time it takes to set up communication and the time it takes for certificate authentication. This is because many MiTM attacks start the creation of certificates once connectivity is established with the targeted attack victim. Since the network transit time exists between the communication establishment time and the server authentication return time, subtraction was used to calculate the Response time. The average amount of time the remote server appears to take to respond to several TCP handshakes was calculated to be the round-trip time. To determine a typical authentication creation time and round-trip time, a list of 30 web applications that use SSL connections was made and manually implemented without an attack. For 30 web applications, data samples are manually collected to determine the average under ideal circumstances and to also represent some shaky network conditions. Once the average RTT for each website has been established, the pre-set RTT obtained from a simulation of an MITM attack using the Kali Linux setup and tools like Ettercap and MITM proxy is compared to the average RTT under normal circumstances.

Figures below show correspondence graphs that illustrate the SSL/TLS handshake process used to account for timing differences.

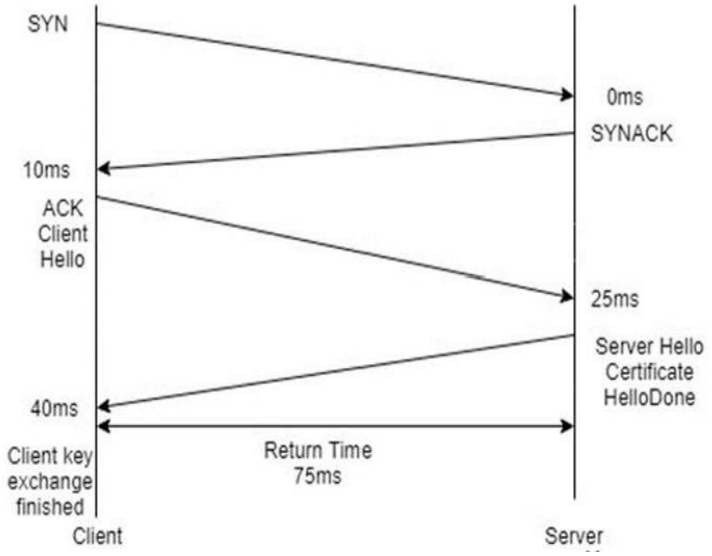

Figure 2: Diagram depicting time rate during it a normal SSL/TLS handshake.

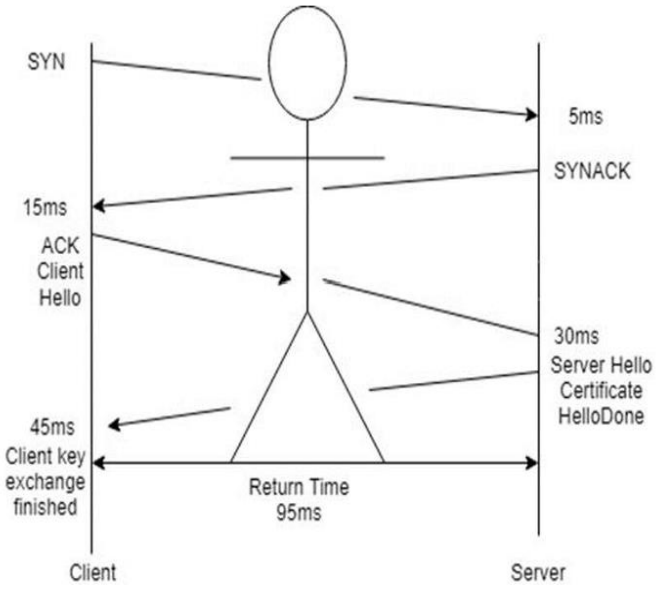

Figure 3: Diagram showing the rate of SSL/TLS handshakes under MITM attack.

3.4 MITM attack Deployment

It can be designed to use a certificate supplied by users to sign all the certificates it produces for the web servers that it imitates. Ettercap is a program in the Kali Linux operating system that performs Address Resolution Protocol poisoning and MitM proxy feature deploys MitM attack on TLS/SSL sessions by intercepting SSL connections. Consequently, if a forger obtains a trusted CA signing certificate with a private key, "PKI-related" attacks may be made use of it. It was possible to simulate an MITM attack using a Kali Linux virtual machine, its tools Ettercap and MitM proxy, and IP tables set up to direct web traffic from a client VM hosted on Virtual Box to gather sample data for 30 web applications of our choosing. After the attack was over, the average RTT value was determined by calculating the time it took to create the TCP communication, the time it took to create the SSL/TLS communication, and both

of those times. To perform a comparative analysis and identify the presence of an attacker, this calculated RTT separately for TCP and SSL shake between the client and web application.

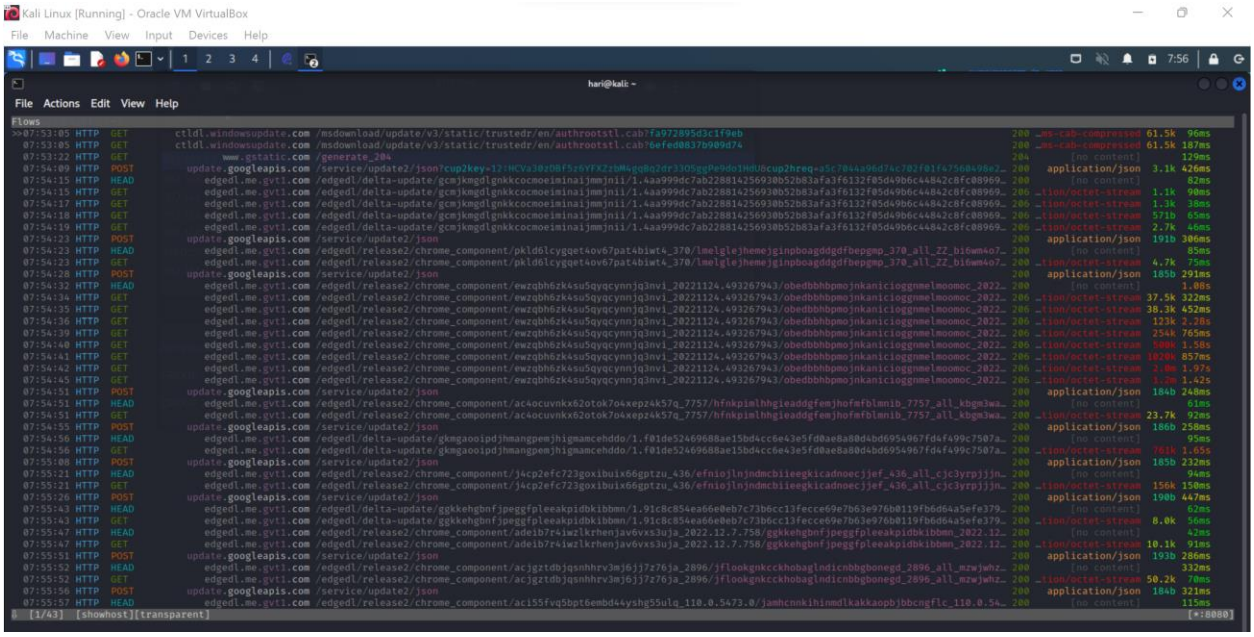

Figure 4: MITM flows open-source interactive HTTPS proxy

4 Design Specification

The experiment is conducted on a virtualization environment Oracle VM virtual box using two VMs Kali Linux and Windows 10. Tools like Ettercap and MITM proxy are used to carry out the MITM attack on the Kali Linux VM serving as the middle device. As a client machine Windows 10 VM has been used where open-source packet analyzer tool Wireshark is installed to analyze TCP and SSL communication packets. For the timing analysis of the web applications' TCP and SSL/TLS communication sample data, Python 3.10 was used. Python is one of the programming languages used for data analytics of sample data to perform visual representation. Its mobility, versatility, and interactive elements are the reason for this. PyCharm is the Integrated Development Environment (IDE) employed. Python code can be written and run using the PyCharm development platform.

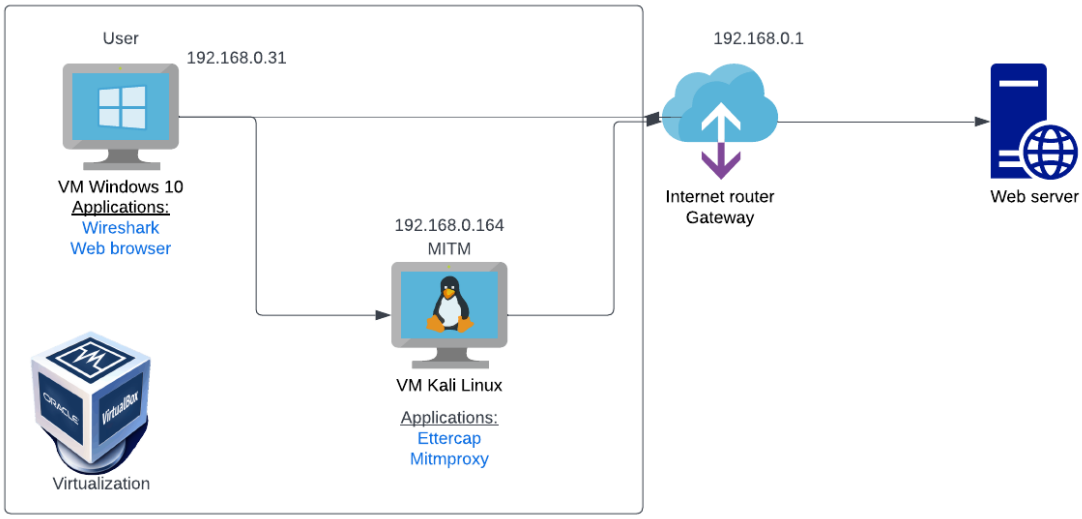

Figure 5: Implementation Stage

4.1 Proposed model

The proposed system would include the average value of SSL handshake between client and server to detect the existence of an MITM attack in the infrastructure. The average value is determined by calculating the round-trip time of TCP and SSL handshake separately for 30 different web applications with MITM and without MITM attack. RTT value of TCP and SSL communication for each website has been found out by capturing the live traffic using Wireshark on the client VM Windows 10. A program uses a pre-set MITM RTT obtained during a simulated attack and a second pre-set RTT obtained from a simulation that doesn't include the presence of the MITM to perform a comparative analysis to determine whether an MITM is present. With the aid of Ettercap and the MITM proxy tool on the Kali Linux VM, the simulated MITM attack was made real.

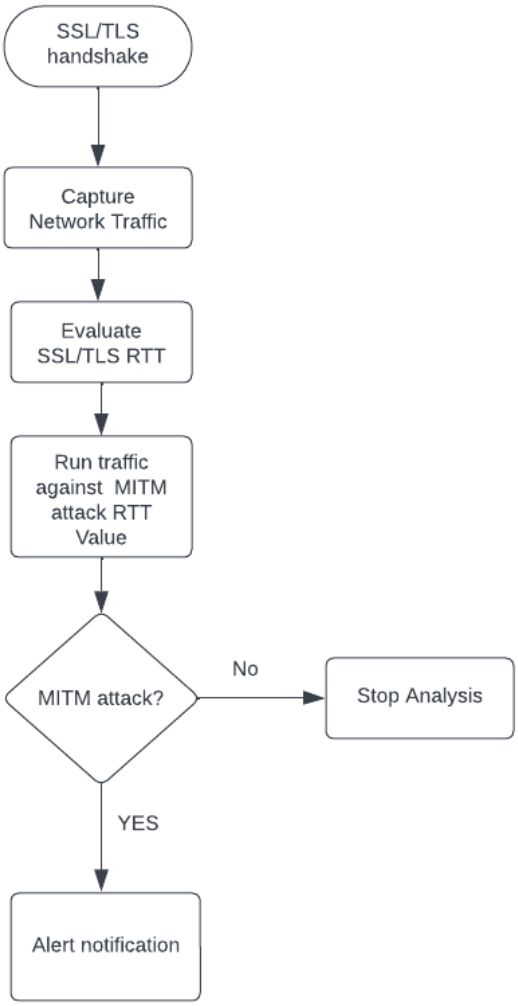

Figure 6: Diagram of an activity showing the suggested detection method.

5 Implementation

This section outlines the procedure used to put the suggested solution into practice. The implementation setup was deployed to generate TCP and SSL communication for websites and with simulation of MITM attack. A Python program was developed for comparative analysis for the sample data in the experiment.

5.1 Man-in-the-Middle attack simulation

The setup is built on the Virtualization environment Oracle virtual box with two VMs running windows 10 and Kali Linux and connecting internet router with bridged adapter mode. The attack is launched from a Kali Linux virtual machine that serves as the middle device and a Windows 10 virtual machine as the client. The IP address of the router is the same gateway for both VMs. Using the tool Ettercap, ARP poisoning was successfully deployed to make Kali Linux a middle device. The validation of ARP Poisoning is done from the Windows 10 client machine by comparing the ARP cache table before and after the ARP Poisoning attack.

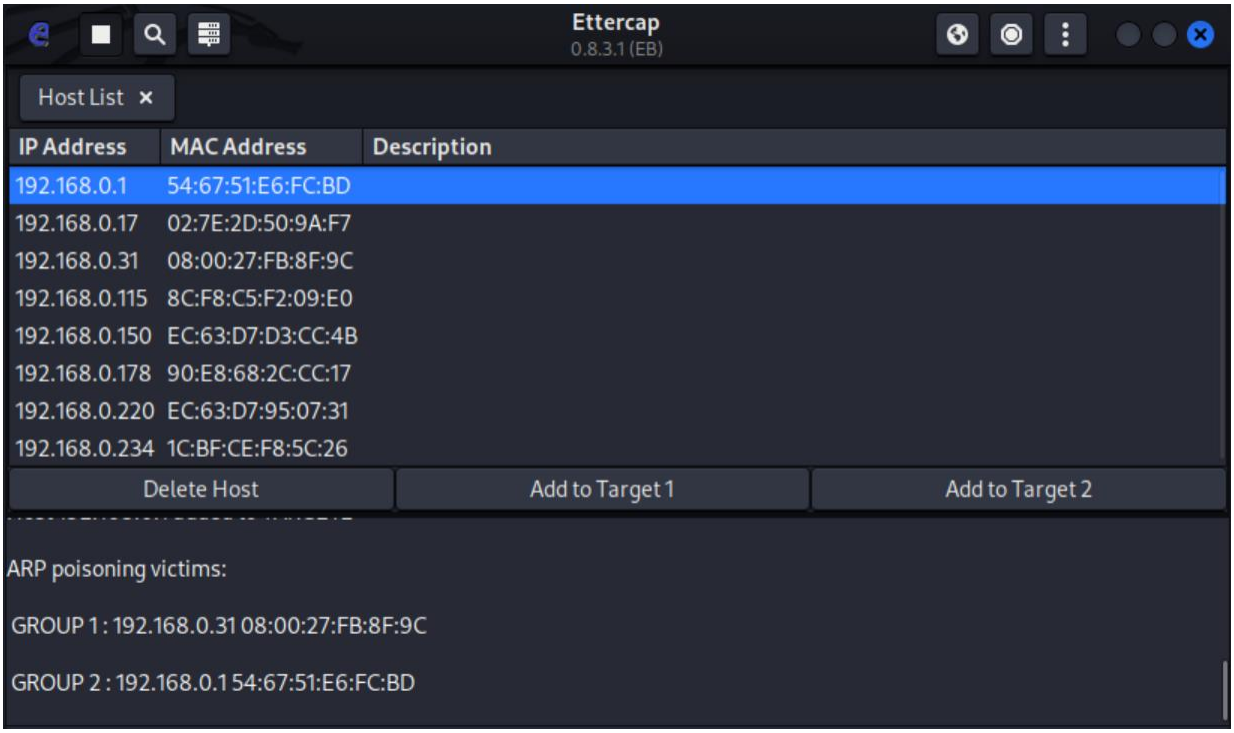

Figure 7: ARP Poisoning using Ettercap

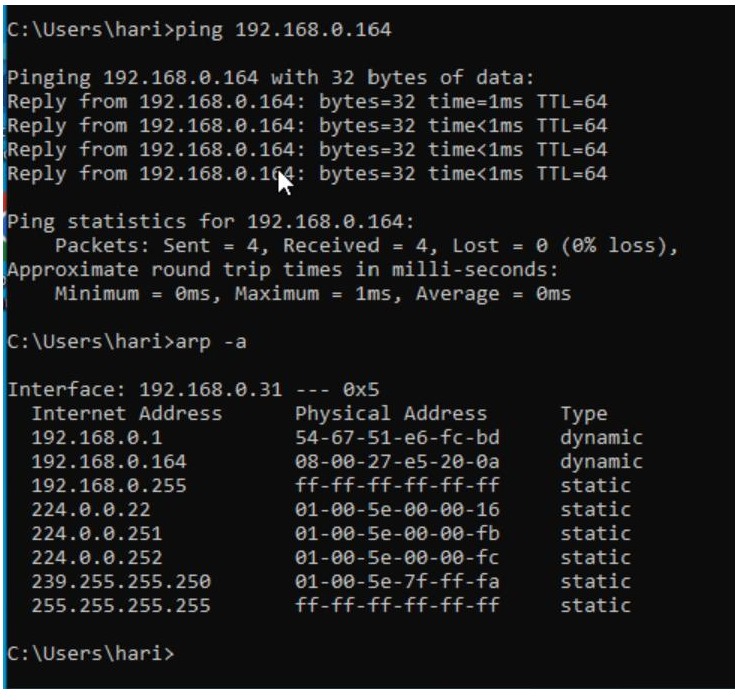

Figure 8: Client Machine's ARP table before ARP Poisoning

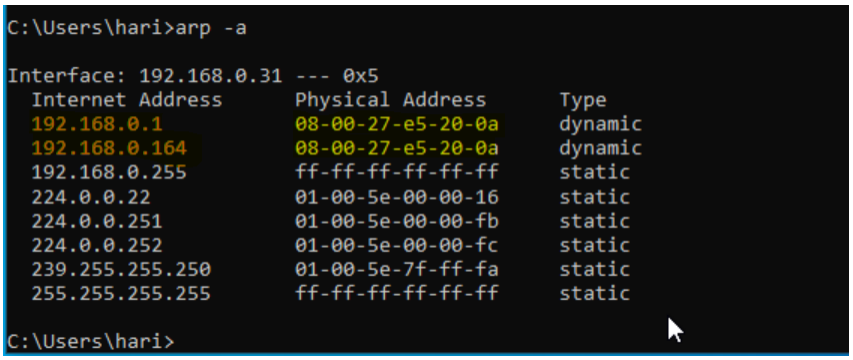

Figure 9: Client Machine's ARP table before ARP Poisoning

5.2 Man in the Middle attack for HTTPS traffic using Mitmproxy

On the Kali Linux VM, mitmproxy is activated. It is a command-line tool that logs all traffic and serves as an HTTP and HTTPS proxy. To achieve transparent mode, the iptables redirection mechanism is integrated with mitmproxy. Because the attackers won't have control over the client machine in a real-time man in the middle attack scenario, we replicated it using mitmproxy in transparent mode. When a transparent proxy is being optioned to opt, traffic is diverted at the network layer into a proxy without the need for client configuration. To create a redirection mechanism that transparently reroutes a TCP connection intended for a server on the Internet to a listening mitmproxy, configured an iptables ruleset on Kali Linux. To execute the man in the middle attack in transparent mode on Kali Linux, we turned on IP forwarding and turned off ICMP redirects. MITM attacks are avoided by the HTTPS protocol. The only thing users need to understand about the HTTPS protocol is that a trusted Certificate Authority (CA) is used to sign certificates. If a trusted CA signs the certificate, our browsers assume that we are communicating with the person we believe to be ourselves. To help overcome this, mitmproxy has

generated a certificate and private key that is specific to Kali Linux and is in /usr/local/share/ca-certificates.

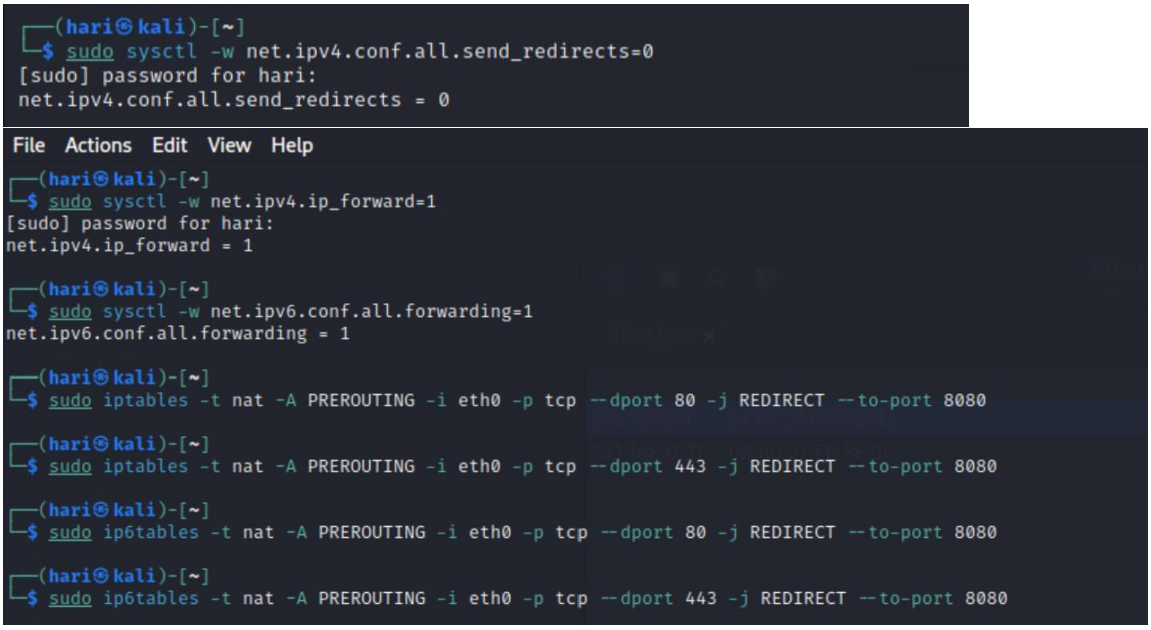

Figure 10: IP forwarding and IP table configurations

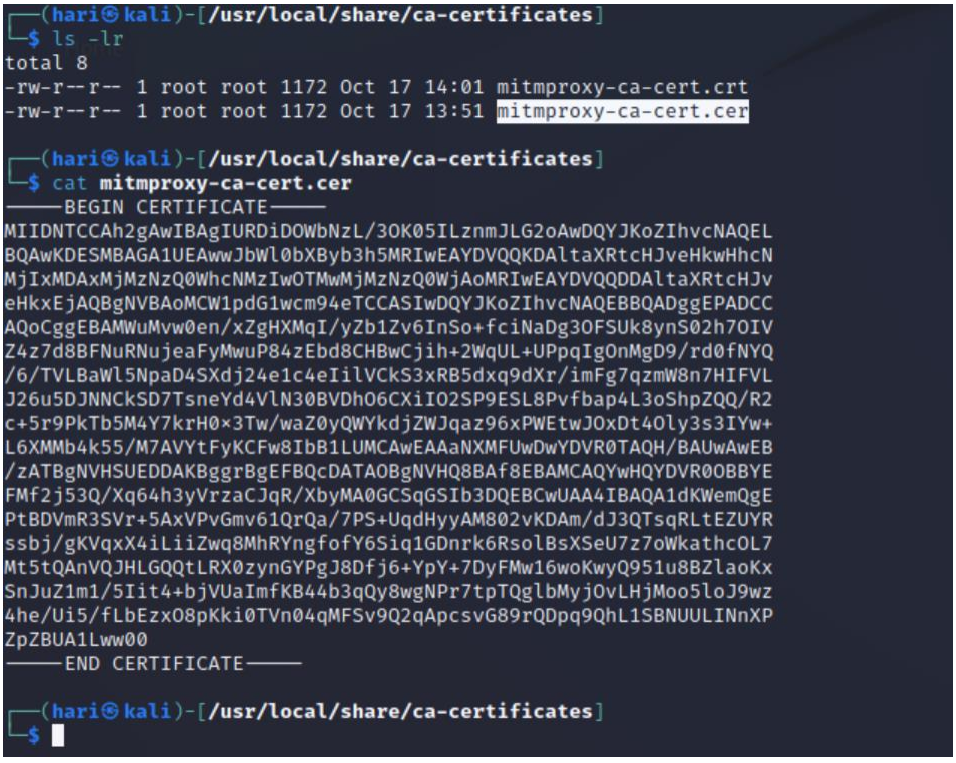

Figure 11: MITM proxy SSL certificate

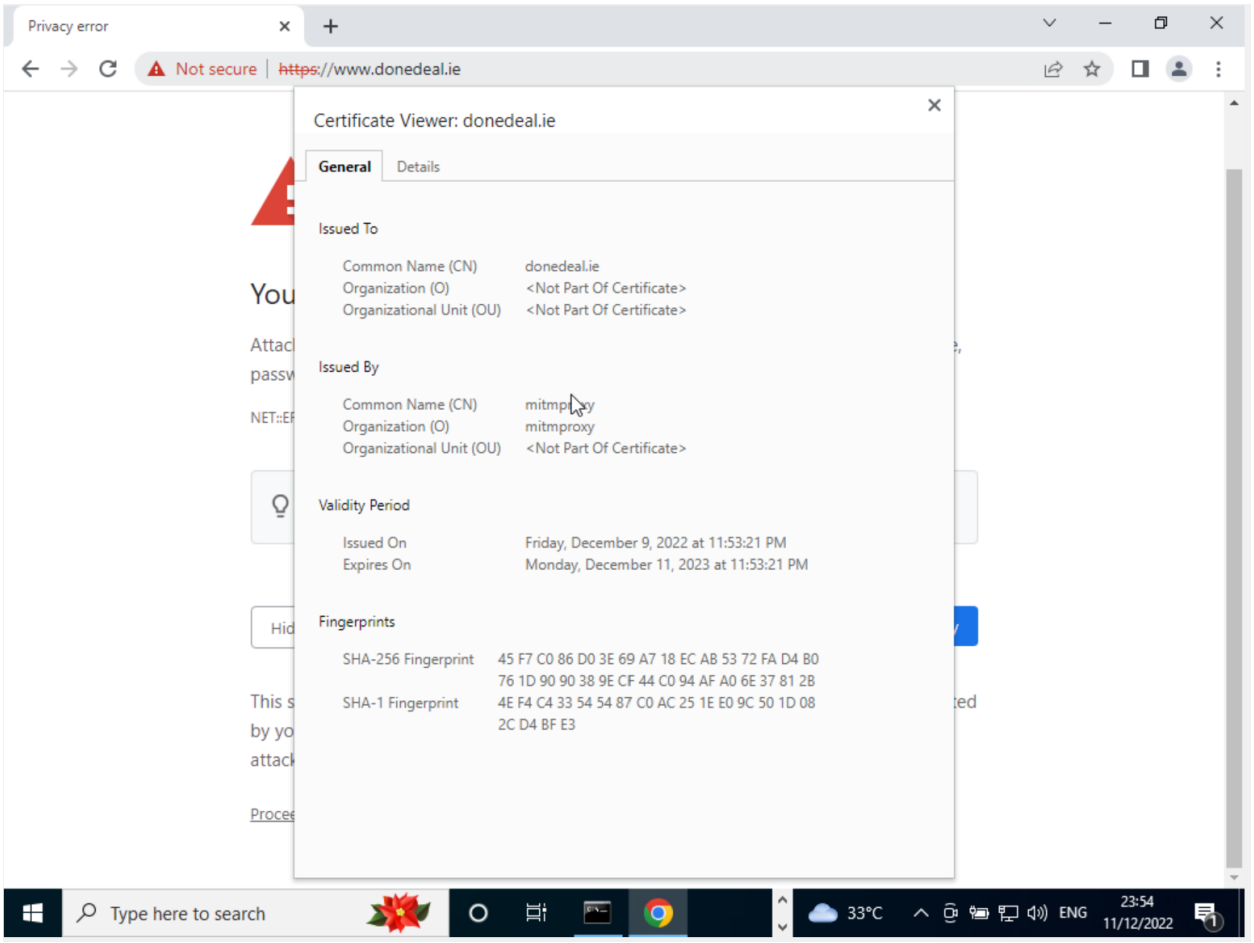

Figure 12: MITM proxy certificate reflection on client's browser

5.3 Implementation of timing analysis

We installed Wireshark on the client machine Window 10 to capture the packet of the TCP and SSL shake between the client and web application. The https connections were initiated from web browser on client machine for each URL parallelly Wireshark application runs behind to capture the traffic. Initially we captured the traffic of TCP and SSL communications without MITM attack and repeated the same procedure to collect the timestamps data with MTIM attack. Using filter options and other features in Wireshark to capture the TCP and SSL steam flow with sequence timestamps for each website. The experiment was conducted using 30 different domain names and determined the time taken to finish a TCP Handshake as well as an SSL handshake. The obtained results are then saved and tabulated in an Excel spreadsheet.

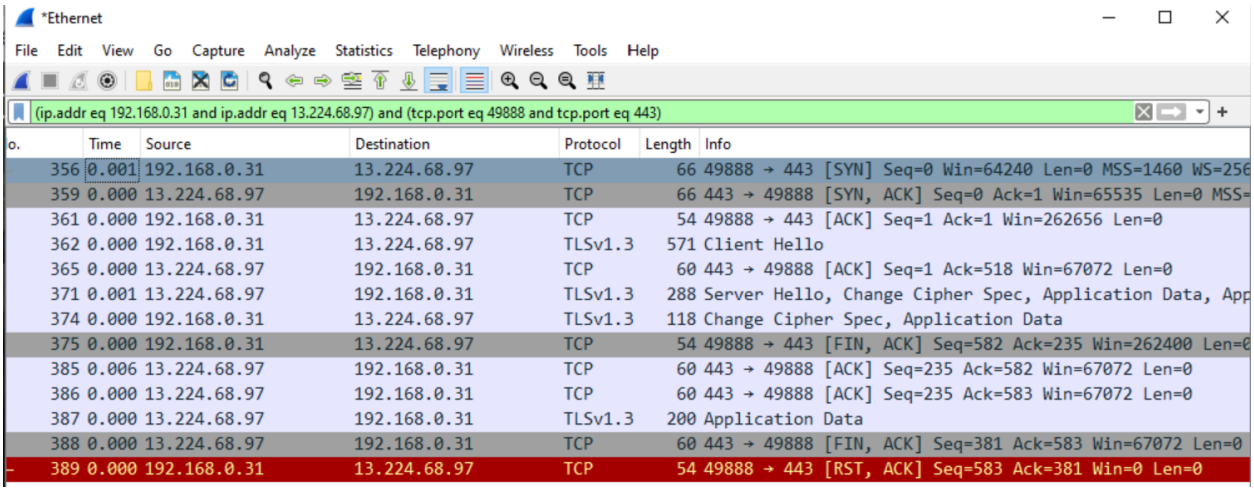

Figure 13: Wireshark capture of TCP and SSL Shake traffic before MITM attack

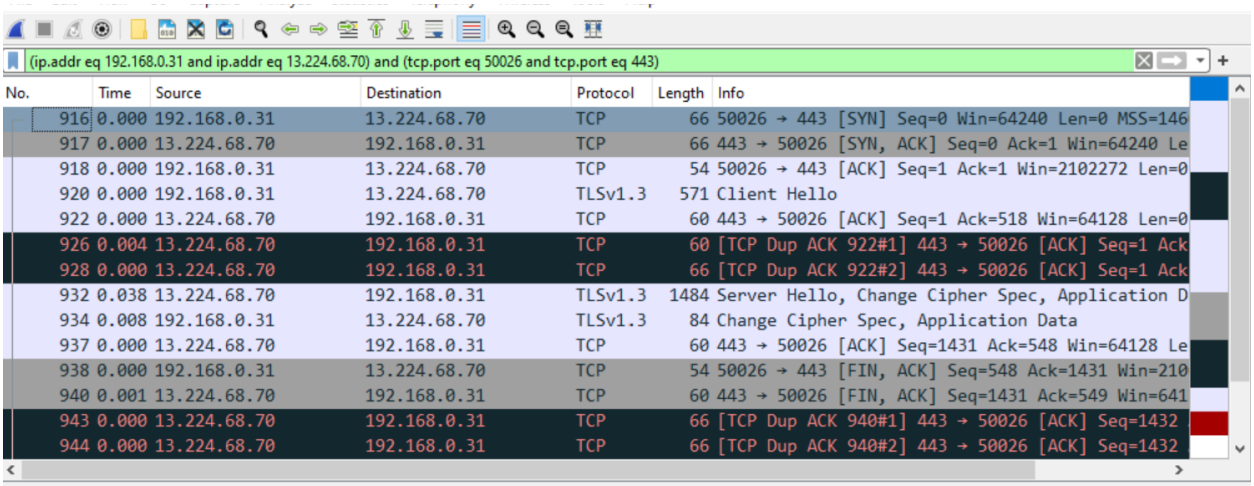

Figure 14: Wireshark capture of TCP and SSL Shake traffic after MITM attack

5.4 Implementation of timing comparative analysis

Following the completion of the time required to complete a TCP and SSL handshake, the average RTT for each domain is calculated separately for TCP and SSL conversation for more discrete parameter value to feed the MITM detection system during the simulated attack. A program is developed using python, this code reads from an excel file which has been used in storing the output generated from simulated setup which contains the RTT value derived for 30 different web applications during MITM attack and RTT gotten when there was no presence of a man in the middle. To determine the MITM attack in visual representation, a comparative analysis was performed between the timestamps of TCP and SSL before and after the MITM attack.

Web Application	TCP Handshake			SSL Handshake		
	Protocol	Response before MITM	Response After MITM	Protocol	Response before MITM	Response After MITM
Donedeal.ie	TCP	1	0	TLS1.3	1	46
www.aerlingus.com	TCP	16	1	TLS1.3	9	90
Bestbuy.com	TCP	14	0	TLS1.3	10	30
www.bbcgoodfood.com	TCP	31	0	TLS1.3	8	80
Dominos.ie	TCP	2	1	TLS1.3	3	66
ebay.ie	TCP	59	0	TLS1.3	11	444
tripadvisor.ie	TCP	9	1	TLS1.3	8	23
circlek.ie	TCP	20	15	TLS1.2	9	28
myhome.ie	TCP	17	1	TLS1.2	12	78
ie.indeed.com	TCP	4	0	TLS1.3	11	32
kahoot.it	TCP	8	1	TLS1.3	1	25
news.sky.com	TCP	18	1	TLS1.3	9	16
webmd.com	TCP	23	0	TLS1.2	3	10
www.nhs.uk	TCP	25	1	TLS1.3	6	20
www.unitconverters.net	TCP	59	2	TLS1.2	7	24
dublincoding.ie	TCP	24	0	TLS1.2	3	3
targettransport.com	TCP	17	14	TLS1.2	17	32
www.opentable.ie	TCP	31	4	TLS1.3	4	31
www.lush.com	TCP	6	2	TLS1.3	10	20
www.forbes.com	TCP	5	4	TLS1.3	5	28
www.nokair.com	TCP	21	0	TLS1.2	9	14
just-eat.ie	TCP	33	0	TLS1.3	21	12
www.yelp.com	TCP	17	0	TLS1.3	7	9
bigten.org	TCP	4	1	TLS1.2	1	7
www.livescience.com	TCP	10	0	TLS1.3	1	13
soundcloud.com	TCP	15	0	TLS1.3	1	15
www.youthworkireland.ie	TCP	8	0	TLS1.2	6	14
tyrecentre.ie	TCP	0	0	TLS1.3	2	10
www.space.com	TCP	14	0	TLS1.3	3	12
joolstv.com	TCP	0	0	TLS1.3	3	4
Average	TCP	16.6	1.6	TLS	6.7	41.2

Figure 15: Sample data set collected during the research

6 Evaluation

Comparative analysis of the generated TCP response time dataset before and after MITM attack is discussed in subsection 6.1 to demonstrate the effectiveness of MITM detection method. Subsection 6.2 discusses our findings from an assessment of SSL response time datasets generated before and after an MITM attack when connecting to each domain server. A Python code for evaluation was created using the pandas and matplotlib libraries.

6.1 Comparative analysis of TCP time responses obtained with and without MITM attack

The outcome obtained when the evaluation code visually represents the difference in round trip timing results of TCP connections for 30 selected domains are shown in the figure. According to the analysis report, there is no increase in time response for TCP communication after an MITM attack. The outcome led us to conclude that the parameter TCP response time value is useless for identifying MITM attacks.

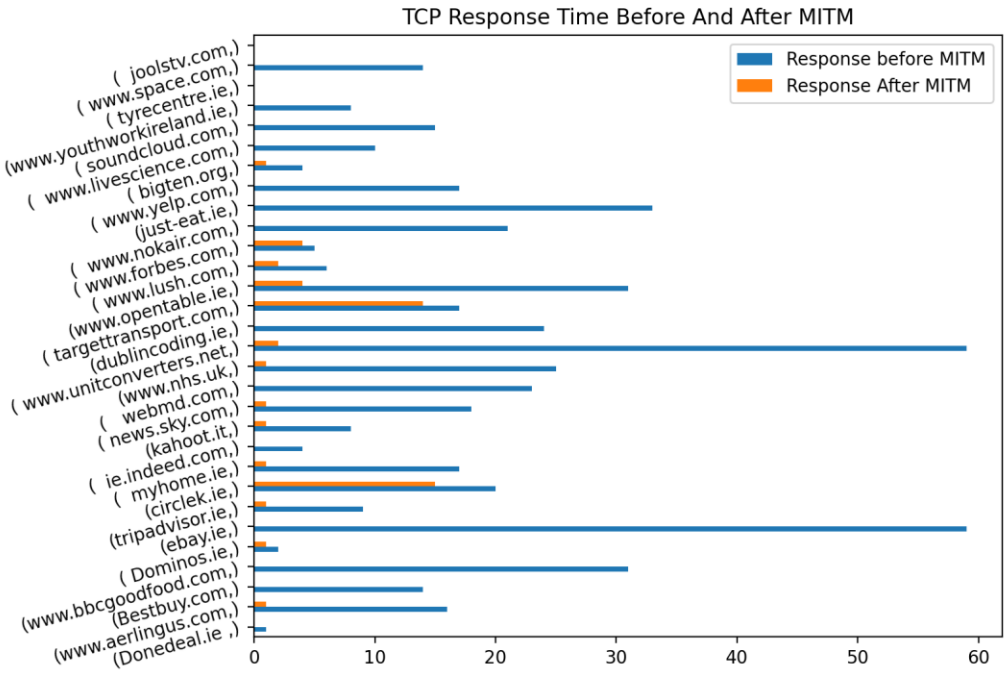

Figure 16: Histogram of TCP RTT results gotten with and without MTIM attack

6.2 Comparative analysis of SSL/TLS time responses obtained with and without MITM attack

The figure displays the outcomes of using our written program to represent graphically the variance in round trip timing results of SSL/TLS connections for 30 chosen domains. The analysis report states that

after an MITM attack, the time response for SSL/TLS communication dramatically increases. The results led us to draw the conclusion that the parameter mean value SSL/TLS response time is crucial for spotting MITM attacks in communications between web applications.

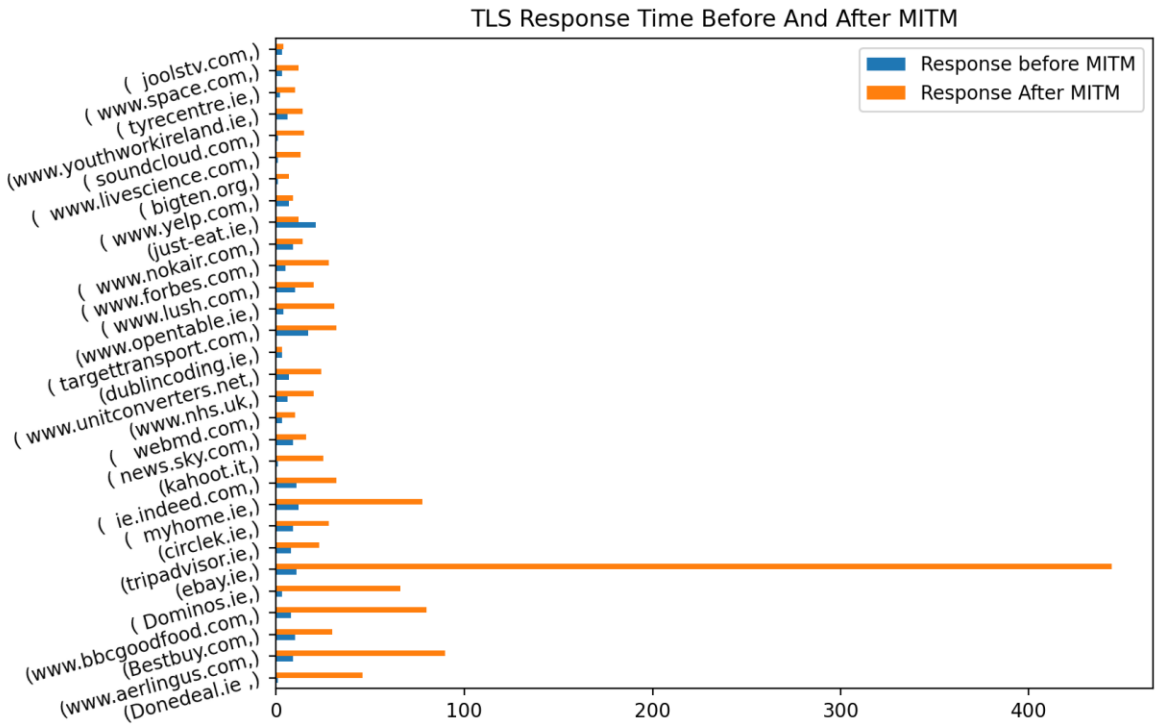

Figure 17: Histogram of SSL RTT results gotten with and without MTIM attack

6.1 Discussion

In this study, we analyzed the response time of 30 websites across the globe, and through a collection of 30 samples, we were able to calculate the average RTT for TCP and SSL communication of web applications. Based on the datasets we collected, we discovered that an average RTT for TCP was 16.6 milliseconds and for SSL was 6.7 milliseconds during a typical client-server connection without the existence of an MITM. The lowest average RTT was about 1.ms for a TCP connection and the highest average RTT was about 41.2ms for an SSL/TLS connection when a connection was simulated with the existence of an MITM. Since the timing differences for TCP connections are quite different and will not be helpful in determining the MITM attack, the consistently perceptible timing differences for SSL/TLS connections from the 30 samples collected can be used as key parameters to detect the existence of an MITM threats. Due to time and high-tech resource limitations, more specifications ought to have been set to account for natural occurrences that might cause timing latency on networks, but this would have further enhanced the suggested system's detection ability.

7 Conclusion and Future Work

The found parameter with the MITM detection system demonstrated that MITM attacks for https traffic have similar response time patterns. In the framework of our studies, the trial that was employed to replicate attacks revealed considerable differences in the length of response time for SSL communication to finish their round trips when trying to connect to various websites across the globe while in a threat. Future studies may focus on investigating the various reasons why SSL communication delays occur, and other factors might lead in a detection scheme returning false-positive results and suggesting mitigation strategies.

References

Arvind, S., & Narayanan, V. A. (2019). An Overview of Security in CoAP: Attack and Analysis. 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 655–660. https://doi.org/10.1109/ICACCS.2019.8728533

Aziz, B., & Hamilton, G. (2009). Detecting Man-in-the-Middle Attacks by Precise Timing. 2009 Third International Conference on Emerging Security Information, Systems and Technologies, 81–86. https://doi.org/10.1109/SECURWARE.2009.20

Brooks, M., & Yang, B. (2015). A Man-in-the-Middle attack against OpenDayLight SDN controller. Proceedings of the 4th Annual ACM Conference on Research in Information Technology, 45–49. https://doi.org/10.1145/2808062.2808073

Crume, J. L. (2008). Detecting and defending against man-in-the-middle attacks (United States Patent No. US20080295169A1). https://patents.google.com/patent/US20080295169A1/en

Dierks, T., & Rescorla, E. (2006). The Transport Layer Security (TLS) Protocol Version 1.1 (Request for Comments RFC 4346). Internet Engineering Task Force. https://doi.org/10.17487/RFC4346

Dun-Yi, Y. (2020). Data Encryption Method of Ssl Digital Authentication Signature System Based on Privacy Protection. 2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), 40–44. https://doi.org/10.1109/ICMTMA50254.2020.00016

Folarin, S. (n.d.). IMPROVED SSL/TLS MAN-IN-THE-MIDDLE ATTACK DETECTION TECHNIQUE USING TIMING ANALYSIS AND OTHER BEHAVIORAL ANOMALIES. 19.

Gerretzen, B. M. (2022, July). Information leakage via Certificate Transparency [Info:eu-repo/semantics/bachelorThesis]. University of Twente. https://essay.utwente.nl/91730/

Jonas, M. A., Hossain, Md. S., Islam, R., Narman, H. S., & Atiquzzaman, M. (2019). An Intelligent System for Preventing SSL Stripping-based Session Hijacking Attacks. MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM), 1–6. https://doi.org/10.1109/MILCOM47813.2019.9021026

Li, Y., Zhu, L., Wang, H., Yu, F. R., & Liu, S. (2021). A Cross-Layer Defense Scheme for Edge Intelligence-Enabled CBTC Systems Against MitM Attacks. IEEE Transactions on Intelligent Transportation Systems, 22(4), 2286–2298. https://doi.org/10.1109/TITS.2020.3030496

Li, Z., Xiong, G., & Guo, L. (2020). Unveiling SSL/TLS MITM Hosts in the Wild. 2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE), 141–145. https://doi.org/10.1109/ICISCAE51034.2020.9236866

Mirsky, Y., Kalbo, N., Elovici, Y., & Shabtai, A. (2018). Vesper: Using Echo-Analysis to Detect Man-in-the-Middle Attacks in LANs (arXiv:1803.02560). arXiv. https://doi.org/10.48550/arXiv.1803.02560

Sherman, A. T., Seymour, J., Kore, A., & Newton, W. (2017). Chaum's protocol for detecting man-in-the-middle: Explanation, demonstration, and timing studies for a text-messaging scenario. Cryptologia, 41(1), 29–54. https://doi.org/10.1080/01611194.2015.1135487

Tung, Y.-C., Shin, K. G., & Kim, K.-H. (2016). Analog man-in-the-middle attack against link-based packet source identification. Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing, 331–340. https://doi.org/10.1145/2942358.2942361

Vallivaara, V. A., Sailio, M., & Halunen, K. (2014). Detecting man-in-the-middle attacks on non-mobile systems. Proceedings of the 4th ACM Conference on Data and Application Security and Privacy, 131–134. https://doi.org/10.1145/2557547.2557579

Vanhoef, M., Bhandaru, N., Derham, T., Ouzieli, I., & Piessens, F. (2018). Operating Channel Validation: Preventing Multi-Channel Man-in-the-Middle Attacks Against Protected Wi-Fi Networks. Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks, 34–39. https://doi.org/10.1145/3212480.3212493