National College of Ireland

IMPROVED SSL/TLS MAN-IN-THE-MIDDLE ATTACK DETECTION TECHNIQUE USING TIMING ANALYSIS AND OTHER BEHAVIORAL ANOMALIES

MSc Internship
Cybersecurity

Samuel Folarin
Student ID: X18133495

School of Computing
National College of Ireland

Supervisor: Dr. Ross Spellman

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Samuel Folarin
Student ID:	X18133495
Programme:	Cybersecurity **Year:** 2018/2019
Module:	Internship
Supervisor:	Dr. Ross Spellman
Submission Due Date:	12th August 2019
Project Title:	IMPROVED SSL/TLS MAN-IN-THE-MIDDLE ATTACK DETECTION TECHNIQUE USING TIMING ANALYSIS AND OTHER BEHAVIORAL ANOMALIES

Word Count:5941 **Page Count:** 19

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: .....

Date: .....

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):	

IMPROVED SSL/TLS MAN-IN-THE-MIDDLE ATTACK DETECTION TECHNIQUE USING TIMING ANALYSIS AND OTHER BEHAVIORAL ANOMALIES

Samuel Folarin
X18133495

Abstract

Even with the necessary protection that the TLS protocol is assumed to provide, data communications and financial transactions carried out online have been noted to be at great risk due to the impending danger of Man-in-the-middle attacks. This research was conducted to confirm the possibility of mitigating the continuous threat that attacks such as the man-in-the-middle constitute to the SSL and TLS key exchange by analyzing differences in time and other possible behavioral anomalies between a simulated attack and a standard SSL session through the use of machine learning. The results gotten in this research have been used to demonstrate the successful implementation of an improved SSL/TLS MiTM detection system.

Keywords: Man-in-the-middle attack, Machine learning, timing analysis, behavioral anomalies, SSL/TLS key exchange.

1 Introduction

Preventing malicious entities from gaining access to information that have been tagged sensitive has been a recurring challenge when it comes to the security of the cyber space and restricting the unauthorized exposure of such sensitive data in today's cyber environment has proved to be quite difficult. A prominent attack strategy out of the many being utilized by cyber-attackers is the man-in-the-middle attack. According to OWASP, MiTM is not only an attack technique, it is also an assessment technique utilized during the developmental step of a web application or it can be said to be utilized for the assessment of Web Vulnerabilities[1]. The 2015 nakedsecurity.sophos.com article[2] in which 49 suspects were busted across Europe

[1]https://www.owasp.org/index.php/Man-in-the-middle_attack
[2]https://nakedsecurity.sophos.com/2015/06/11/49-busted-in-europe-for-man-in-the-middle-bank-attacks/

on the suspicion of using man-in-the-middle attacks to discover and ambush cash transaction emails further cements our believes on the gravity and the sophistication of this cyberattack strategy. A man-in-the-middle attack is a cyber-attack that occurs when 2 communicating parties employs a communication mechanism that involves the use of public or shared keys as a security measure. Some articles and reports sometimes refer to it as "TCP-Hijacking", "Monkey in the Middle attack" or "bucket brigade attack"[3]. As described in the case above, the key exchange between the two legitimate communicating parties is intercepted by the attacker and he goes ahead to establish two self-reliant connections with the said parties by utilizing new keys. If the attack is successfully carried out, the attacker has the ability to view, copy and possibly alter all data being communicated before passing them forward to the legitimately intended user through the use of the fake key mentioned earlier.

The hypothetical SSL/TLS version of the above is as follows:

Figure 1: Man-in-the-Middle attack

- In this case, Roland initiates an SSL handshake intended for Folarin without realizing that the connection has been taken over by Kelly (Man-in-the-middle);
- Kelly initiates his own separate SSL session with Folarin
- Once the SSL session with Folarin has been initiated, Kelly goes further to generate and send a certificate that appears to be from Folarin to Roland
- If this certificate is accepted by Roland, he would have unknowingly established an SSL session using Kelly's key
- All messages sent by Roland can now be decrypted by Kelly and then sent over to Folarin via the SSL session he established with him earlier
- Kelly can now pass on messages over the SSL sessions while viewing/altering the plain text data without Roland or Folarin suspecting.

In the hypothetical case above, Roland accepted the certificate from Kelly that impersonated Folarin which is what makes the attack possible. This project focuses on looking at SSL/TLS man-in-the-middle attacks from a network tracking perspective to evaluate if certain behavioural anomalies occur during TCP connection that may be a pointer to the presence of a man in the middle attack. Specifically, an analysis of time difference was performed to determine if the phase of certificate generation by the attacker can be noticed due to long response time upon the start of the SSL handshake.

[3]Ben Zahler and Isabel Steiner *Man in the middle (MITM) attack*

The literature review section discusses previously conducted studies in detecting MITM attacks that are directly related to our current research problem. The techniques and the methods engaged in conducting the study as well as the structure of the proposed technique were explained in the Methodology section. The Design specification section details the techniques, architecture and framework that have been used during the implementation process as well as the associated requirements. The implementation section describes our derived outputs as well as tools and languages you used to produce the outputs. The evaluation section provides a comprehensive analysis of the results gotten and findings made during this research while the Conclusion and Future work section states how successful this research has been in answering our proposed research question and achieving our objectives as well as areas of this study that can be considered for future work alongside the commercialisation potential of the study.

2 Literature Review

Multiple research works have been conducted in the past and researchers have recommended diverse detection methods to identify the man-in-the-middle attack. This section would put into perspective, these past related studies and focus on bringing to limelight the important elements that have been utilized in all the discussed cases as well as remarkable technological progress that makes the detection of man-in-the-middle attacks a possibility.

2.1 Related works

One of the newly created strategies utilized by cyber-attackers in accomplishing MITM attacks is the Stealth MITM (SMITM) attack. This strategy involves creating a fake frame ARP response protocol structure and performing an exploitation using the WPA2 key management in such a way that it directly carries out an ARP poisoning to the unsuspecting victim. A Wireless Intrusion Detection System (WIDS) solution that was reported as successful in the detection of SMITM and other related ARP poisoning and IP Spoofing attacks was proposed by Vikas Kumar et al. (2012). The proposed system was simulated by the authors by utilizing an NS-3 network simulator. It was discovered that even though the system had the ability to utilize a single sensor for the purpose of detecting a static attacker, combined effort is required between the nodes of the sensor for a mobile ARP poisoning attacker to be accurately detected. The daily increase in the quantity of available IoT devices has in turn created an aggressive rise in the volume of data that providers of cloud services process. This rise in data volume has in turn created an increase cloud services latency which has caused an increase in many IoT application latency. A fog layer that allows the processing of data without users having to go to the data centre was created by service providers, this solution that was assumed to eradicate the problem of latency gave birth to a cloud service vulnerability exploitable to carry out a MiTM attack since techniques traditionally developed for the purpose of preventing intrusion are not fog level applicable. Intrusion Detection system (IDS) and Intrusion Prevention System (IPS) were researched by Farouq Aliyu et al. (2018) as possible solutions to the detection of MiTM attacks. Upon deployment of the proposed IDS-IPS system, it was discovered that a minimal latency

overhead of 40ms was produced but when closely observed, a discovery shows that this minimal latency overhead produced was at the expense of the time it took for the investigation to be completed. This investigation time is an inverse proportion of the network latency and energy overhead.

2.2 Timing Analysis

Static timing analysis is a method of simulation utilized in the calculation of the time a digital circuit is expected to use in fully completing a process execution without the necessary requirement of fully simulating the entirety of the circuit. A general attribute of all heavy-performance complex circuits is to utilize a timing frequency for operational functions. In the research conducted by Benjamin Aziz and Geoff Hamilton (2009), they analysed certain class of protocols such as wireless sensor networks utilized for attack detection in a timely manner by mobile systems and concluded that these protocols have vulnerabilities which are exploitable by class of cyber-attacks in which MiTM attacks falls. A static analysis algorithm which utilizes precise timing was proposed as a solution for detecting MiTM attacks occurring on mobile communications. For an increased efficiency of the proposed algorithm, more protocol security properties such as the minimum/maximum time taken for a complete authentication process in a real-time system should have been included in the research since it is possible for an intruder to easily exploit a slow protocol to create a DOS attack. A different strategy that possesses similar fundamental attributes for detecting man-in-the-middle attacks was proposed by Visa Villivaara et al. Their method involves making use of the timestamps of TCP packet headers for calculating delays. They explained that upon calculation of these delays, a comparison can be carried out between the mean delay value gotten from a current connection and data collected from previous connection sessions. The result gotten from this comparison is then used as a benchmark for detecting any long delays in the packets which seem suspicious. They went further to show that a parameter can be set as threshold to detect man-in-the-middle attacks accurately with the probability returning false positives being low. Even though the results of their tests show that unless in the event of a rare occurrence the proposed method can effectively detect MiTM, the solution is solely limited to non-mobile systems and for appropriate functionality, an internet connection that is highly reliable is needed.

Certain sturdy protocols including TLS are utilized by networks for mitigating various cyberattack risks such as the MiTM. A public key infrastructure is utilized by TLS for authenticating public keys exchange. In a contradicting view, Chaum's protocol which is another protocol utilized for the detection of MitM attacks does so without presuming the authenticity of the exchanged public keys. has Three execution phases are contained in Chaum's protocol. Two communicating parties exchange the public keys in the first phase while the generation of a random sting by both parties begins in the second stage. A cryptographic manner is used by the first party to bind itself to the generated string before sending the string to the second party upon receiving their string. The third Phase involves both communicating parties utilizing 4 random "scenarios" for the verification of the possession of the two correct strings by each party. In the occurrence of a MiTM attack, the above explained protocol forces the MitM to cause both communicating parties to be in possession of different pairs of strings. An experiment was conducted by Alan T. Sherman et

al. in a bid to demonstrate the effectiveness of the third scenario through the use of timing analysis to detect a text-messaging MitM attack. They explained that the protocol puts the MitM in a situation whereby he has choice than to cause delays which are noticeable and can be utilized by communicating entities in discovering the man in the middle`s presence. Upon comparison to Interlock and Zfone and other such protocols, it was discovered that Chaum`s protocol protects against attacks which are more powerful attacks but the possibility to give false positive detection is a high since no natural external factors were considered in the research. For more accurate outcomes that can be generally admissible to be derived, research should be carried out by conducting an analysis that studies the various delays to the attacker and there should be a statistical definition of the boundaries utilized in the identification of an attack.

2.3 Behavioural Anomalies

Mastering both the act of data safety and theft requires a deep comprehension of the standard and irregular behaviour of network. An anomaly is any digression from what is normally expected or a difference from standard. A detailed report was given by Jeffery L. Crume of a system created by him for the detection of man-in-the-middle attacks. The said system as stated in the report is made up of 3 distinct parts, a method, an activity recording system and a program product that can be used for detecting server-based man-in-the middle attacks. He explains that one part of his system which is the activity recording system monitors all IP addresses and UserIDs interacting with the server alongside each session`s occurrence time. This activity recording system also contains an activity analysis system that constantly checks if the session quantity tracked to a single IP address within a defined period of time is acceptable or not as a method for detecting dubious IP addresses alongside a system for mitigation which decides the action to be executed when such dubious IP address is recognized. There is a distinct attribute observed to be shared by all successful server-based man-in-the-middle attacks which causes the server to believe a sizeable quantity of users for no reason are using the same IP address to login to the server. A relatable example of a previously unidentified IP exceeding an established limit such as “UserID quantity within a specific time period” causing it to be under immediate scrutiny or even better, an automated mechanism for defence to be immediately initiated was given by the author. The above depicted system and protection strategy has certain points of interest which incorporates the tuning of the boundary to meet risk tolerance level of various companies/organisations, the automation of the detection and countermeasure implementation which can help in reducing harm caused to the server. The procedure can be stretched out past web spoofing/phishing attacks to more MiTM situations.

When a local area network (LAN) is involved, to successfully execute a MitM attack involves following a 3-step prototype; (i) procure network access (ii) trap network traffic in motion (iii) control, adjust, or drop the traffic. Yisroel Mirsky et al. portrayed the man-in-the-middle attack as a basic yet prominent cyberattack where a mischievous entity ambushes network traffic and in the process puts the confidentiality, integrity, and availability of the system at risk. The authors explained that the available solutions for this attack are numerous and contended that these available solutions either need convey ability, give false positives

which are extremely high or they are simply not conventional. They proposed a fit-in-and-play MitM identifier for LAN which utilizes a procedure resembling impulse response analysis utilized in areas used for the processing of acoustic signals. This proposed strategy is same as how echoes in a cave catch the structure of the earth and furthermore equivalent to how a brief and quick beat of ICMP echo requests builds the connection between two network hosts. They clarified that the proposed detector utilizes neural networks in the creation of a structure of the standard pattern of the pulses that were echoed, and detects any environmental changes. Despite the fact that the exploratory outcomes demonstrate that Vesper (the proposed system) can distinguish MitM assaults, more research work should be carried out through the application of more ping strategies (e.g., TCP SYN), techniques for mitigation, and stretching out the strategy to work over Wi-Fi systems, virtual private systems and so on.

TLS is required for the creation of virtual private systems and one of the major security parts of TLS configuration is key exchange and the authentication procedure which is executed utilizing certificates. In the event that a key exchange procedure is unsecure, a man-in-the-middle (MITM) vulnerability might be exploitable. In general practice, certificates utilize Public Key Infrastructures (PKIs) and trusted certificate authorities (CAs) are in turn utilized by these PKIs for approval. As of late, different security worries that could prompt a man-in-the-middle attack if PKIs are utilized have emerged. Enrique de la Hoz et al. proposed a distributed certificate assessment framework named MIDAS and explained that it can be utilized in the detection of propelled man-in-the-middle attacks. The authors clarified that the framework is based on network monitoring and management systems that have existed in the past with a goal of creating a new method that permits the effective analysis of the certificates by the hosts to determine their authenticity during TLS. Despite the fact that the framework looks encouraging, more work ought to be done to create practical Bayesian systems fabricated explicitly to imitate highly-sensitive real life network occurrences that would help the early detection of suspicious or phony certificates and thus trigger some type of defence implementation against the attack and furthermore keep records of the details of the attacker.

Over the span of this survey, we talked about different detection procedures and their shortcomings against various refined MiTM attacks. We decided that utilizing a mix of timing and behavioural anomalies would fix these shortcomings. Section 3 of this research work further explains how timing investigation and other behavioural analysis have both been implemented utilizing machine learning.

3 Research Methodology

3.1 Objective

The derived outcomes when this experiment was concluded were utilized in deciding whether a detection system that precisely points out the presence of MiTM attacks has truly been created. The detection accuracy of the proposed system was determined using an assessment

of the proportion of false positives inferred. The fundamental target of this research work is to demonstrate that a system for detecting MiTM attack can be executed utilizing a mix of timing and other behavioural anomalies through Machine Learning as referenced in section one.

3.2 Overview of methodology

Our proposed solution was actualized by using Machine Learning to decide the presence of a man in the middle through an examination of the time taken in finishing a typical SSL/TLS handshake and an examination of the typical traffic behaviours expected during this process. A comparative analysis was then carried out on the data gotten from this investigation against data gotten from the simulation of a MiTM attack through the use of Cain & Abel. Research works from Kevin Benton, Ty Bross and Jeffery L. Crume gave major direction on providing this solution for SSL/TLS man-in-the-middle attack detection issues.

3.3 Overview of machine learning

Machine Learning as referenced before above is a written program which gains knowledge from continuous encounters concerning a given class of assignments and execution. As expressed by Mohssen M.Z.E et al; 2016, Machine learning is a top developing science that has an incredibly wide scope of applications, it wasn't always a well-favoured territory of computer science and artificial intelligence (AI) but has now developed to turn into a forefront research solution in both AI and PC frameworks engineering with investments in both software and hardware which have progressed at a rate that is only equalled by a similar sort of investments in blockchain innovation over the previous decade. With the utilization of machine learning as the accessible programming/instruments and PC programs, it is conceivable to achieve a detection technique that produces results with an increased accuracy since it employs more than one determinant factor to distinguish a man-in-the-middle attack.

3.4 Detection features and Application

The rate of success or lack thereof and precision of a man-in-the-middle attack detection system is highly dependent on the determining factors/features utilized in arriving at a conclusion about the attack`s presence.

To gauge how long it takes to finish a typical SSL handshake, contrasted with the time it takes during a MiTM assault, a program that starts an SSL handshake and stops the communication upon the receipt of a certificate from the remote party was written. The program does a calculation of the time it takes to create a TCP communication and the time it takes to get a reply containing the certificate from the remote party. Since a ton of MiTM attacks begin the creation of certificates once a communication is set up with the targeted attack victim, a usually large distinction between the time the time it takes to set up the communication and the certificate authentication time would be present. subtraction was done to compute the RTT, since the transport time of the network is present between the connection establishment time and the server authentication return time. The RTT was

determined to be a gauge of the average time it takes the remote server to react to several TCP SYN bundles. A rundown of 15 domains that utilize SSL was created and the code was run without an attack to set up a standard authentication creation time and round-trip time (RTT).

The written code gathers 25 test samples in a 5-minute time span for each domain so as decide the average during normal conditions and furthermore to represent certain precarious network conditions. Upon determining the average RTT for each domain, a comparison of the average RTT under normal conditions is done against the pre-set RTT gotten from a simulation of a MITM attack using Cain & Abel by a second written program to detectt the presence of a MITM.

Sketched out in the figures underneath are correspondence graphs demonstrating the SSL/TLS handshake procedure used to break down timing differences.

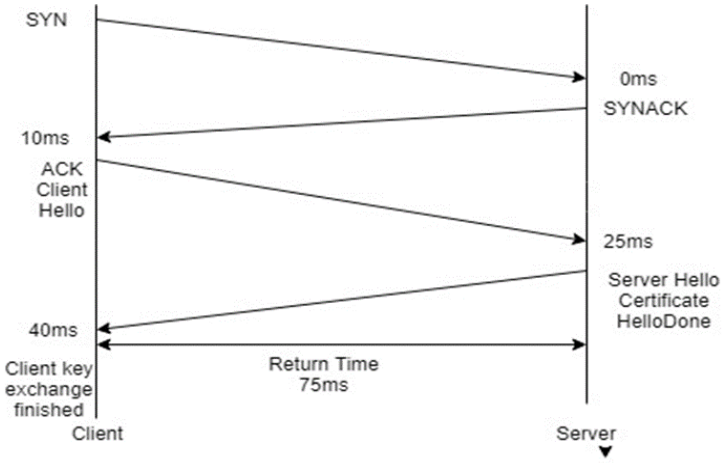


Fig. I: Diagram depicting time rate during it a normal SSL/TLS handshake

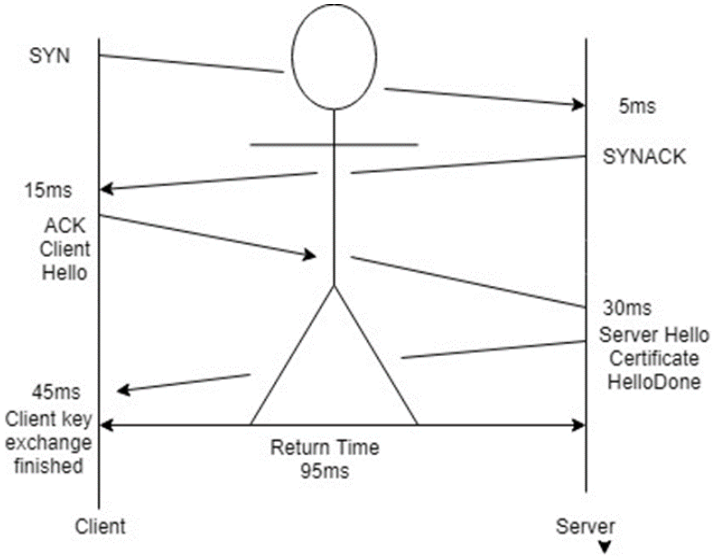


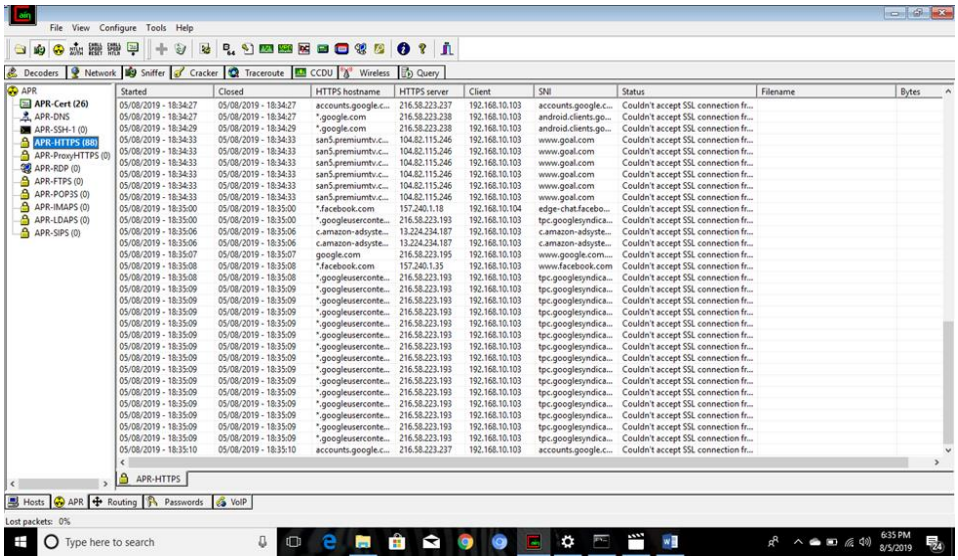
Fig. II: Diagram depicting SSL/TLS handshake time rate under MITM attack

According to the diagrams above, it is worthy to take note of the fact that the aggregate time taken to finish the SSL/TLS handshake with a man-in-the-middle present has expanded by

25ms when contrasted with the time it takes to finish a similar handshake under ordinary conditions.

3.5 MiTM attack Deployment

Cain & Abel is an application that performs Address Resolution Protocol poisoning MITM attacks on TLS/SSL sessions and can be designed to utilize a certificate provided by users to sign all of the certificates it generates for the web servers that it imitates. Therefore, if an impostor attains a trusted CA signing certificate with a private key, it could be used for "PKI-related" attacks. A MITM attack was simulated to collect 25 samples on 15 domains of our choice with the use of the Cain & Abel application. Upon completion of the attack, the time taken for the creation of the TCP communication and the time taken to get a reply containing the certificate was calculated and extracted and the average RTT was calculated for each domain. This calculated RTT was used as a pre-set parameter for our written code as a basis for comparative analysis to determine an attacker`s presence.



4 Design Specification

Python 3.6 was used to implement the Timing Analysis and Behavioural Analysis. Python is one of the Programming Language majorly used for machine learning based algorithm. This is because of its portability, flexibility and Interactive features. The Integrated Development Environment (IDE) used is PyCharm. PyCharm is a development environment for writing and executing python code. It was developed by JetBrains as a cross-platform IDE for Python. It runs on Windows, Linux, MacOS. It provides tools and feature to write and build different applications efficiently and effectively.

4.1 Proposed model

The proposed system would include a home page. This home page which will be used by clients to confirm the presence and none thereof of a man-in-the-middle follows the secure coding principles stated by OWASP. At the click of the start button on the home page, the

script connects to the server of any inputted URL and begins the SSL/TLS key exchange. Once the exchange has been completed, the script terminates and network traffic data is captured. The program then goes further to calculate the RTT (round trip time) which is an estimate of the average time taken to respond to several TCP SYN packet by the remote server. Once this calculation is completed, a comparative analysis is carried out by a second program using a pre-set MITM RTT that was gotten during a simulated attack and a second pre-set RTT gotten from a simulation that doesn't include the presence of the MITM to detect the presence of a MITM. Our simulated attack was actualised with the use of Cain & Abel.

Below is an activity diagram of the above explained model.

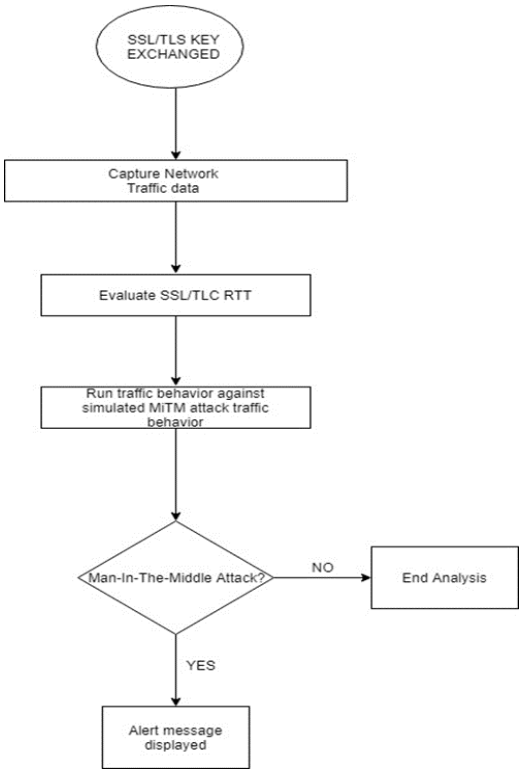


Fig. III: Activity diagram depicting the proposed detection technique

5 Implementation

This section describes the process adopted in the implementation of the proposed solution. A program that initiates a Secure Socket Layer (SSL) handshake was developed. The program ends the connection the moment the certificate has been gotten from the remote party. Another program was developed for comparative analysis.

5.1 Simulation of Man-in-the-middle attack

The live process of simulating our attack was carried out using Cain & Abel. Gadgets used include a Network Router, 3 different PCs running on Windows Operating System. The 3 PCs were connected to the Internet via the Internet Router with one of the PC running the Cain & Abel MITM application. Upon successful connection of the 3 PCs to the internet, the

PC with the MITM software was used to sniff into the Internet activities of the remaining PCs. The diagrams below show the vivid steps followed in carrying out the Live test.

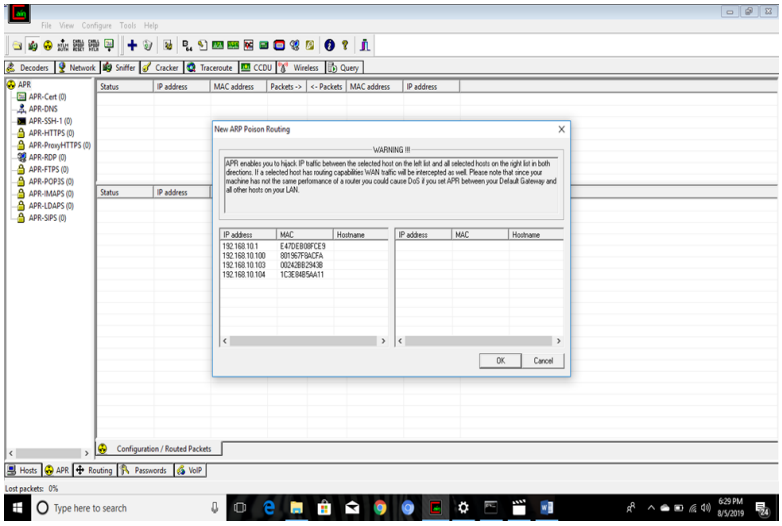


Fig 5.1: Cain & Abel interface showing beginning of MITM attack

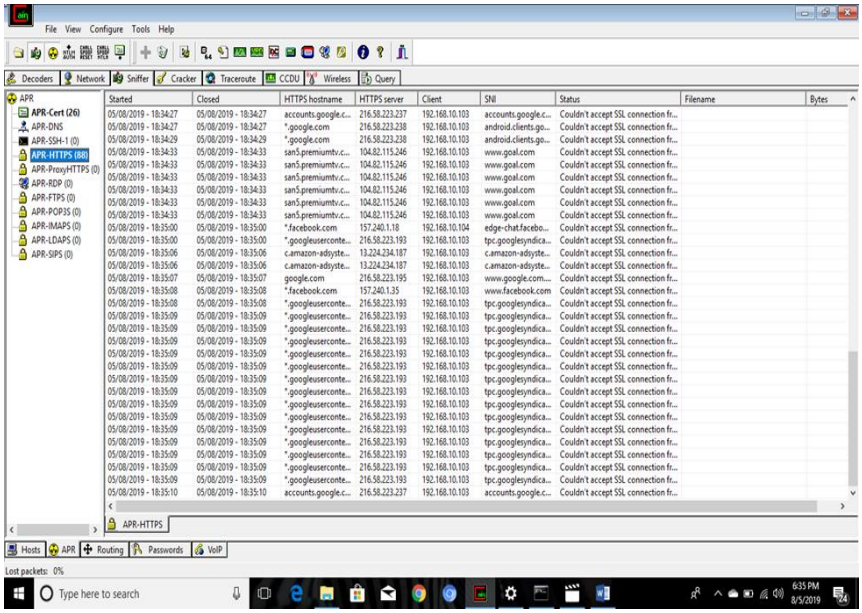


Fig 5.2: Connection being made to the selected URLs

5.2 Implementation of timing analysis without MITM

The Python application is a programming language used for machine learning. We developed a program which runs a test using different domain names and calculates the time taken to finish a TCP Handshake as well as an SSL handshake. The application runs in a loop of 25 tries returning the corresponding time taken for TCP and SSL for each URLs. The result gotten is then stored and tabulated. It is worthy of note that the output of the program also

depends on the speed of Internet Connectivity. I already specified some selected URLs for testing in our program.

```
def model_create(self):

    addresses = ['www.google.com','www.facebook.com','www.yahoo.com','www.amazon.com','iczn.org',
                 'nairaland.com','jumia.com','konga.com','python.org',
                 'www.instagram.com','cnn.com',
                 'www.stackoverflow.com','outlook.live.com',
                 's.yimg.com',
                 'widgets.outbrain.com','onesignal.com',
                 'www.baidu.com','static.eu.criteo.net',
                 'www.bing.com','www.youtube.com',
                 'yastatic.net','www.twitch.tv',
                 'www.ebay.com','www.netflix.com',
                 'medium.com','www.paypal.com']
```

Fig. 5.3: Default selected URLs

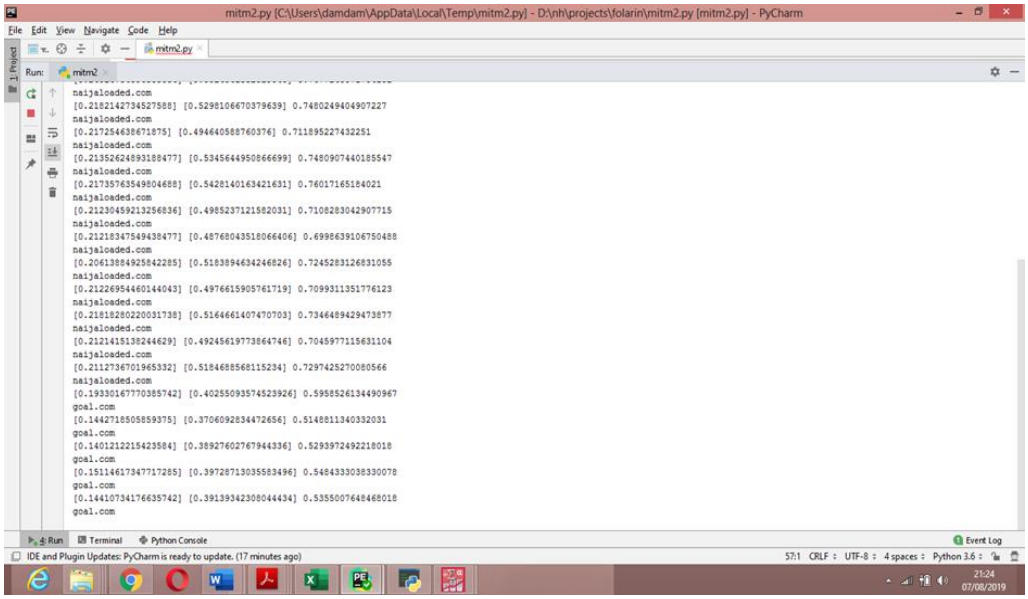


Fig. 5.4: Result showing timing output of the selected URLs

5.3 Implementation of timing comparative analysis

Upon complete capturing of time taken to complete a TCP and SSL handshake, the average RTT for each domain is calculated from the 25 samples collected during the simulated attack and that derived from our written program. Another program is then developed using python, this code reads from an excel file which has been used in storing the output generated from both CAIN/ABEL and our previously implemented program which contains the average RTT derived during MITM attack and the average RTT gotten when there was no presence of a man in the middle. This second written program provides the user with a start page that allows him enter a choice URL from the 15 default sample domains and conduct a network test to detect the presence of an attacker. If the RTT gotten during the test for a particular domain is > the pre-set RTT for the absence of MITM attack or = to the pre-set RTT for the presence of MITM attack, an alert message is displayed saying “PRESENCE OF MAN-IN-THE-MIDDLE” otherwise “NO MAN-IN-THE-MIDDLE” is displayed.

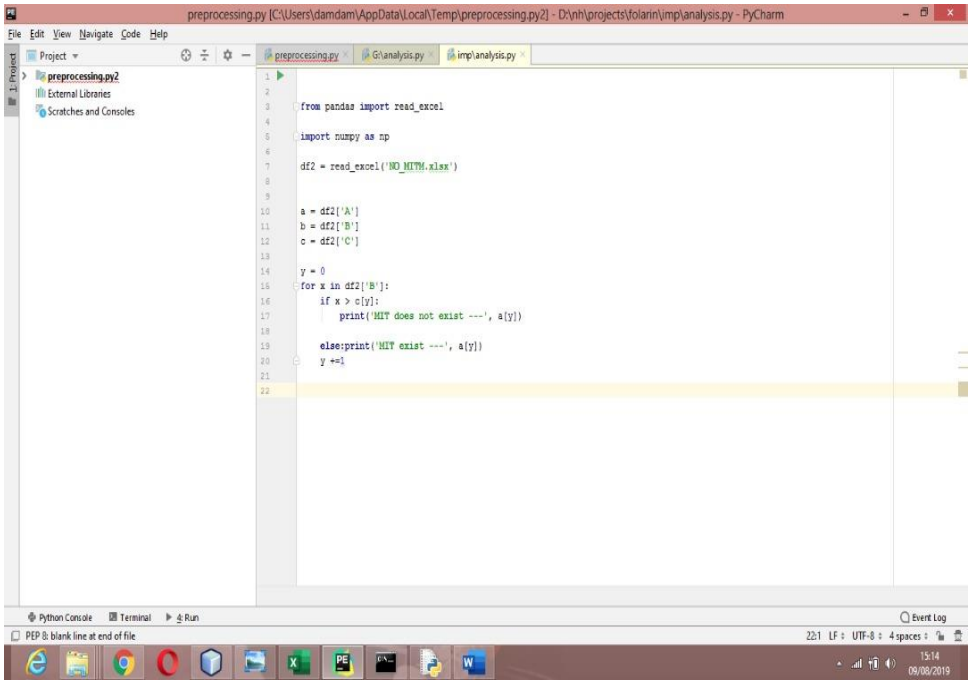


Fig 5.5: Parameters set as determinants for carrying out comparative analysis

Fig 5.6: Home page for users to carryout analysis on choice domain

6 Evaluation

To verify the quality of our man-in-the-middle detection technique implemented, an analysis of the generated MITM attack dataset is discussed subsection 6.1. while subsection 6.2 discusses our discovery from the analysis of the dataset generated when connection was made to each domain server without a MITM attack. The results gotten in 6.1 and 6.2 were then

compared against each other in section 6.3 to confirm if truly the presence of a MITM has any effect on the time taken to complete an SSL/TLS handshake and if it is worthy of being used as a parameter for detecting such attacks. Microsoft excel tool Pak was employed for this data analysis purpose.

6.1 Investigation of RTT without the presence of MITM/Case Study 2

Figure 6.1.1 shows the results gotten when our written program was used to generate round trip timing results for the 15 selected domains. 25 samples were collected and an average RTT was calculated to be used as a set parameter in section 6.3

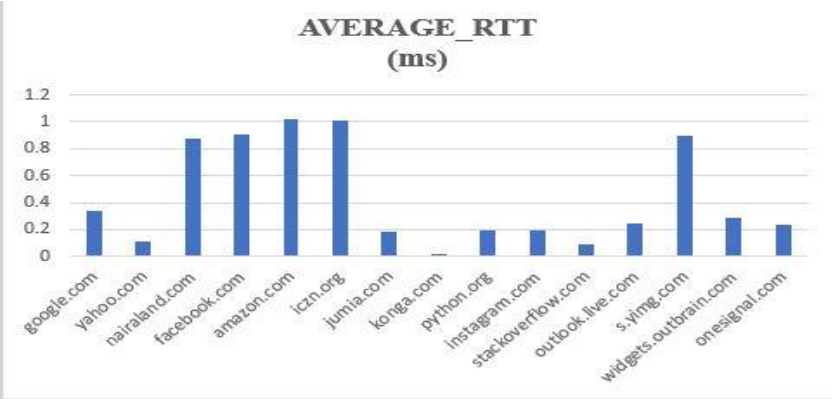


Fig 6.1.1: Histogram of average RTT results gotten without the presence of MITM

6.2 Investigation of RTT with the presence of MITM/Case study 1

Figure 6.2.1 shows the timing results gotten from the simulation of man-in-the middle attack carried out with the use of Cain & Abel which shows an increased time taken to complete RTT when compared with the average RTT gotten in section 6.1 above

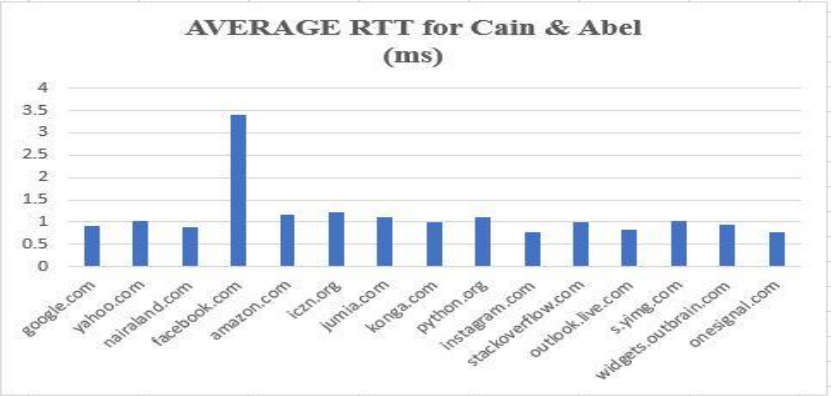


Fig 6.2.1: Histogram of average RTT results gotten with the presence of MITM

6.3 Comparative analysis of results derived with and without the presence of MITM/Case Study 3

Figure 6.3.1 Shows the result which gives evidence that indeed the time taken for a complete round trip during a man-in-the-middle attack is noticeably increased when compared to the time taken for a round trip when there`s no presence of man-in-the-middle. This evidence is a pointer to the fact that even though timing differences might be minimal, it should be

considered as a very important factor when trying to create a system for the detection of the presence of a man-in-the-middle on a network. This guarantees the confidentiality and the integrity of information passed on that network since the man-in-the-middle can be detected before any real damage can be carried out.

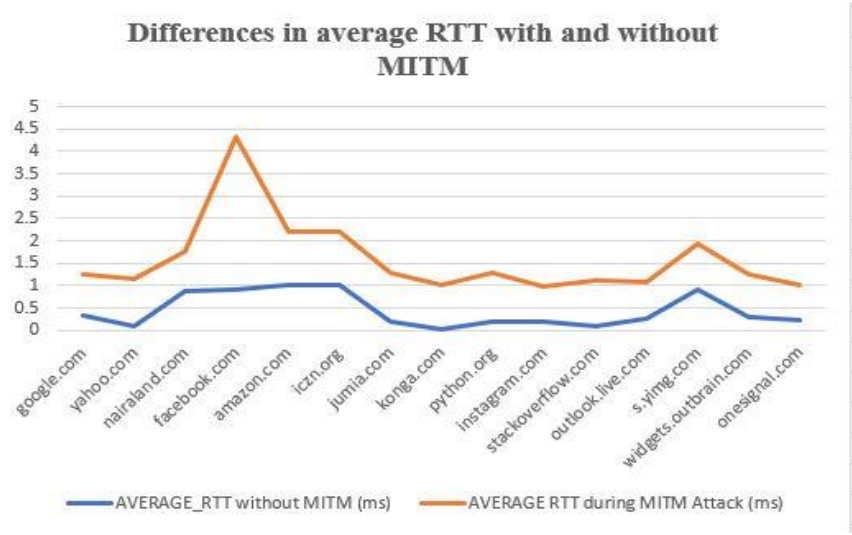


Fig 6.3.1: Line chart representation of round-trip differences with and without MITM

6.4 Discussion

In the MITM detection experiment, we measured the RTTs of 15 domains across the globe and an average RTT was derived for each domain through a collection of 25 samples. Based on our collected samples, we observed that in a normal connection between the client and the server without the presence of a MITM gave an average RTT that was constantly below 1.0ms. While a connection that was simulated with the presence of a MITM gave the highest average RTT of about 3.4ms which was for facebook.com. This constantly noticeable differences in timing from the 25 samples collected were used as parameters set for our written comparative analysis program to effectively detect the presence of a MITM attack. Due to the constraint of time and high-tech resources, more parameters would have been set to account for natural occurrences that may cause timing delays on networks as this would further increase the detection ability of the proposed system.

7 Conclusion and Future Work

Our proposed system demonstrated that TLS/SSL MITM attacks have network patterns that could be detected by conducting active research on the modus operandi of MITM attacks. Cain & Abel which was used for simulating our attack revealed a high variance in round trip time taken when connecting to various websites throughout the world during an attack which was taken as a positive outcome in the course of our research. Future work can focus on analyzing the various causes of delays during SSL/TLS key exchange that might make a detection system to give false positive results and propose strategies on mitigating them.

References

Alan Johnston, Avaya, Inc., Washington University in St. *Louis*- January 20 2014 *"Detecting Man in the Middle Attacks on Ephemeral Diffie-Hellman without Relying on a Public Key Infrastructure in Real-Time Communications"*

Alan T. Sherman, John Seymour, Akshayraj Kore & William Newton *Chaum's protocol for detecting man-in-the-middle: Explanation, demonstration, and timing studies for a text-messaging scenario* Cryptologia Journal Volume 41, 2017 – Issue 1

Benjamin Aziz and Geoff Hamilton. *Detecting man-in-the-middle attacks by precise timing* The Third International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2009, 18-23 June 2009, Athens/Glyfada, Greece

Bishop, C.M *Pattern Recognition and Machine Learning, Springer,* ISBN 978-0-387-31073

Brian Hernacki and William E. Sobel *Detecting man in the middle attacks via security transitions* United States patent. Patent no. 8,561,181, B1. Symantec Corporation, Cupertino CA(US)

Cyber Defense Lab *Animation of Chaum's protocol for detecting a man-in-the-middle*

E. de la Hoz, G. Cochrane, J. M. Moreira-Lemus, R. Paez-Reyes, I. Marsa-Maestre, and B. Alarcos, *"Detecting and defeating advanced man-in-the-middle attacks against TLS,"* in *2014 6th International Conference on Cyber Conflict (CyCon 2014)*, 2014, pp. 209–221.

Farouq Aliyua, Tarek Sheltamia, Elhadi M. Shakshukib *A Detection and Prevention Technique for Man in the Middle Attack in Fog Computing* The 9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2018)

Folarin Samuel, 2019 *Improved SSL/TLS man-in-the-middle attack detection technique using timing analysis and behavioral anomalies* Research in Computing, National College of Ireland

Jeffery L. Crume *Detecting and defending against man in the middle attacks* United States patent. Patent no. 8, 533, 821, B2. International business machines Corporation, Armonk NY(US)

John R.; Bennett, Forrest H.; Andre, David; Keane, Martin A. *Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming*. Artificial Intelligence in Design '96. Springer, Dordrecht. pp. 151–170. doi:10.1007/978-94-009-0279-4_9

Kevin Benton and Ty Bross. *Timing Analysis of SSL/TLS Man in the Middle Attacks* arXiv:1308.3559v1 [cs.CR] 16 Aug 2013

Maryam Mousaarab Najafabadi *Machine Learning algorithms for the analysis and detection of network attacks* a dissertation Submitted to the Faculty of The College of Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy. Florida Atlantic University Boca Raton, FL August 2017

Mitchell, T. (1997). *Machine Learning*. McGraw Hill. p. 2. ISBN 978-0-07-042807-2

Mohssen M.Z.E Mohammed, Muhammad Badruddin Khan and Eihab Bashier Mohammed Bashier (2016), *Machine Learning: Algorithms and Applications* CRC PressISBN: 9781498705387

Ronnie et al. Radboud University *Persistent effects of man-in-the-middle attacks*

Ted G. Lewis and Peter J. Denning. *The Profession of IT Learning Machine Learning A discussion of the rapidly evolving realm of machine learning* communication of the acm december | Vol. 61 | No. 12

Vegard Flovik. *How to use machine learning for anomaly detection and condition monitoring; Concrete use case for machine learning and statistical analysis* Towards Data science Dec 31, 2018

Vikas Kumar et al. *Detection of Stealth Man-In-The-Middle Attack in Wireless LAN* 2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing

Visa Villivaara et al. *Detecting Man-in-the-Middle Attacks on Non-Mobile Systems* ACM Conference on Data and Application Security and Privacy, 2014 At San Antonio, Texas, Volume: 4th

Yisroel Mirsky, Naor Kalbo, Yuval Elovici, and Asaf Shabtai *Vesper: Using Echo-Analysis to Detect Man-in-the-Middle Attacks in LANs* arXiv:1803.02560v1 [cs.CR] 7 Mar 2018

Ziqian Dong, Randolph Espejo, Yu Wan and Wenjie Zhuang *Detecting and Locating Man-in-the-Middle Attacks in Fixed Wireless Networks* Journal of Computing and Information Technology - CIT 23, 2015, 4, 283–293 doi:10.2498/cit.1002530