Intrusion Detection System on Suricata with Bot Telegram as A Notification System

By

Iqbal Nuril Anwar Bin Tormizi

17004961

Dissertation submitted

in partial fulfilment of the requirements for the

Bachelor of Information Technology (Hons)

SEPTEMBER 2021

Universiti Teknologi PETRONAS

32610 Bandar Seri Iskandar

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Intrusion Detection System on Suricata with Bot Telegram as ANotification System

by

Iqbal Nuril Anwar Bin Tormizi

17004961

A project dissertation submitted to the

Information Technology Programme

Universiti Teknologi PETRONAS

in partial fulfilment of the requirement for the

BACHELOR OF INFORMATION TECHNOLOGY (Hons)

Approved by,

[illegible]

..... DR. MOHD HILMI HASAN
Senior Lecturer
Computer & Information Sciences Department
Universiti Teknologi PETRONAS
32610 Bandar Seri Iskandar,
Perak Darul Ridzuan, Malaysia.

Dr. Hilmi Bin Hassan

Date: 29th November 2021

UNIVERSITI TEKNOLOGI PETRONAS
BANDAR SERI ISKANDAR, PERAK
May 2021

CERITIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original herein have not been undertaken or don by unspecified sources or persons.

[illegible]

____
Iqbal Nuril Anwar Bin Tormizi
Date: 29th November 2021

ABSTRACT

The Intrusion Detection System (IDS) is a type of security system that serves as a protective layer for an organization's information technology infrastructure. IDS technology has evolved tremendously over the years to keep pace with the development of cybercrime. There are many instances when an IDS is bypassed by a novel kind of attack, and a substantial proportion of normal packets are classified as attack packets. This may result in more serious problems if the administrator is not constantly monitoring the network and is unaware of the attack. And by the time they understand it, it is too late. Thus, implementing system notifications through instant messaging platforms such as Telegram may assist users in receiving notifications through their regular mobile application use. Suricata, an open-source program designed specifically for intrusion detection, will be used to identify infiltration of assaults on computer networks, and the system will be able to deliver real-time warnings to administrators through a telegram bot. As attacks become more sophisticated and frequent, the intrusion detection system must be enhanced to guarantee that it is capable of identifying and reacting to new threats.

ACKNOWLEDGEMENT

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

All praise to Allah for the successful completion of this dissertation. I am grateful to God for all of the chances, difficulties, and strength that have been bestowed upon me in order to complete my Final Year Project (FYP).

During this process, I gained a great deal of knowledge and experience, both academically and personally. My deepest gratitude to my Final Year Project (FYP) supervisor, Dr. Hilmi B Hassan for his support and guidance throughout my journey in accomplished my FYP. It has been a great experience and honour to have his as my supervisor.

Next, I would like to give my appreciation to Universiti Teknologi PETRONAS for the knowledge and experience I gain here throughout my years of study. This project would not be possible to be done without the experience and knowledge that I gain throughout my study in Universiti Teknologi PETRONAS.

I want to express my gratitude to all of my dear friends Zahimi, Khairul, and Alif who have stood by me and supported me through thick and thin in completing my project. Thank you for always guiding me and willing to let me learn from them and always willing to share their knowledge to me.

Last but not least, I would like to express my heartfelt gratitude to my dearest father, Tormizi Bin Kasim, and my beloved mother, Aziah Binti Abdullah, for their unwavering support, both mentally and physically, as well as financially. I would not have been able to reach my goals without their help and encouragement.

Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Background of study

Computer technology has advanced tremendously and has continued to take over oursociety to the present day. We have seen growth in a computer network, with the usageof Intrusion Detection System (IDS) which has been widely used to analyze network traffic for suspicious behavior and notifies administrators when such behavior is detected. IDS technology has significantly evolved to keep up with computer crime advancements. This technology was first introduced in the 1980s, and researchers have worked hard to maintain a proper balance between network speed and the ability to detect threats. However, according to Almseidin, Alzubi, Kovacs, and Alkasassbeh (2017), there are numerous threats exist that can undermine the availability, integrity,and confidentiality of computer networks. The primary concern of network intrusion detection systems is that malicious cyber-attacks have increased cybersecurity vulnerabilities and become more sophisticated, resulting in a high number of attack occurrences such as phishing, distributed denial of service (DDoS), and ransomware. This intrusion detection system is an adaptive safeguard technology compared to the traditional security. Modern networked organizational settings need a high degree of security to guarantee the secure and trustworthy exchange of data between different companies. IDS usually will be installed at a location within a network to monitor traffic as well as all devices on the network. As it examines passing traffic and compares it to a database of known threats, it will then identify whether it is an attackor detecting any odd activity.

IDS types may also be classified into two, which are Signature-based intrusion detection system (SIDS) and Anomaly-based intrusion detection system (AIDS). Thakkar and Lohiya (2020) stated that Signature-based or also known as misuse intrusion detection system produces an alert when signature patterns in the signature database is matched. However, based on Almseidin et al. (2017), Signature-based does not generate false alarms, but it is susceptible to being bypassed by a new kind of attack (new signature). In comparison, the same paper also described about Anomaly-based detection, which enables the detection of threats based on previously observed normal behavior. The common issues for Anomaly-based is that it records the highest levels of false positive alarms, indicating that a significant number of regular packets are being treated as attack packets.

Therefore, a real-time notification system is essential, with Intrusion Detection Systems (IDS) playing a key role in detecting and addressing intrusions quickly. This may be resolved by using an instant messenger such as Telegram Messenger, which is extensively used on mobile devices and desktop computers, with over 200 million users and supports a variety of mobile operating systems including Android, iOS, and Windows Phone. Telegram is used as a real-time notification system in situations where system administrators have the ability to communicate with the system in order to get information about the system's status and condition, as well as to modify it. Telegram Messaging will be utilized as an interactive interface for this project, allowing users of this messenger program to receive real-time intrusion warnings and detailed information about them. In this paper, a software called Suricata also will beutilized which is an open source and fast robust network threat detection engine that includes intrusion detection, intrusion prevention, and network security monitoring features. It performs very well at deep packet inspection and pattern matching, making it an extremely useful tool for threat and attack detection, as well as monitoring suspicious activities in the network.

1.2 Problem statement

Since cyber-attacks are becoming more novel and that new threats are constantly emerging, it has become a more serious issue that can be associated with intrusion detection systems, such as the lack of real-time alarm notification from instant messenger to inform administrators in the event of a network intrusion attack.

1.3 Objectives

- To investigate unusual activities and possible attacks using SYN flood attack on a network using an intrusion detection system (IDS).

- To develop a system notification system using Telegram Messenger.

- To test and evaluate the effectiveness of the system.

1.4 Scope of study

According to the above-mentioned objective, the scope of the study focuses on a cyber-attack on an Intrusion Detection System (IDS) with a notification system that makes use of Telegram Messenger primarily for the purpose of detecting when an attack occurs on the system, which is the primary goal of the study. Any organizations or IT administrators that wish to defend their systems from intruders who want to do harm or risk their firm are the primary stakeholders in this system. This research will aim to put testing into action using Suricata, which was selected from among other accessible tools, as well as Hping3, which is a network tool that enables you to transmit manipulated packets across the network.

1.5 Significance

As a result of this research, companies will have access to a tool that will notify them in the event of a cyber-attack that occurs in the course of their normal working day using the Telegram Messenger app. It is anticipated that this technique will lead to a more effective means of preventing IT administrators from being uninformed of an attack or from knowing about it only after the system has been hacked. In certain ways, this technology benefits organizations by putting a solution as well as improving overall system performance at their fingertip.

CHAPTER 2

LITERATURE REVIEW

As we are moving forward in the era of emerging technology, we have witnessed a significantimprovement in business practices in using intrusion detection system (IDS) as their layerof protection as addition to firewalls against the exposures of the Internet. As cyber-attacksbecome more prevalent each year, where companies and organizations must make criticaldecisions and started to invest more on security to protect themselves from cyber-attacks.

2.1 Current implementation

Many researchers have begun conducting a study on intrusion detection system, utilizing a variety of ways to increase the IDS's effectiveness on generating alerts, andnumerous advancements have been produced. A recent study was conducted by Sharma et al. (2012), where observable activity is compared to suspect patterns in intrusion detection systems (IDS), ideally in real time. According to the same researchpaper, they can perform packet logging and real-time traffic analysis on IP networks by using Snort software, which is a free and open-source network intrusion prevention system (NIPS) and network intrusion detection (NIDS). It offers an excellent overview of what is happening in a specific network and enables the automated recording of packets from possible attacks for future reference.

Moreover, Erlansari et al. (2020) mentioned that while the most frequent method of preventing network attacks is to place an administrator, complications may emerge when the administrator is not actively monitoring the network. Thus, the presence of instant messaging apps, one of which is the Telegram application, may assist administrators in receiving real-time alerts. With Snort integrated with IDS to identify early intrusions, it can notify the server administrator through Telegram.

The same research paper also managed to detect attacks with their IDS system such as port scanning, FTP brute force, SSH brute force, and DDoS.

Additionally, based on a research study done by Sulistya and Sasmita (2020), they mentioned that Snort is one of the most widely used network-based intrusion detection systems, with over 400,000 users. However, Snort lacks a suitable GUI (Graphical User Interface), requiring the installation of another program such as BASE. Another issue is that, since an IDS warning system with a web-based interface, such as BASE, cannot inform the system administrator, it is conceivable that users may miss certain attacks, making response impossible. A real-time alerting system like Telegram is needed. This research successfully created an IDS system capable of detecting and preventing threats by blocking IP addresses defined by port, protocol, and time periodduring the block.

2.2 Type of attack

3.1 DoS (Denial of Service)

According to Almseidin et al., (2017), DoS attacks are designed to prevent end users from accessing a variety of services for a short period of time. In general, it wastes network resources and causes the system to become overloaded with unnecessary requests, which is not desirable. This is why denial of service attacks serve as a broad umbrella term for all forms of assaults that try to consume computer and network resources. The same paper also mentioned that Yahoo was the first victim of a distributed denial of service (DoS) assault, which occurred on the same day that DOS recorded its first public strike. At the moment, distributed denial of service assaults is being launched against online services and social networking websites.

3.2 SYN flood

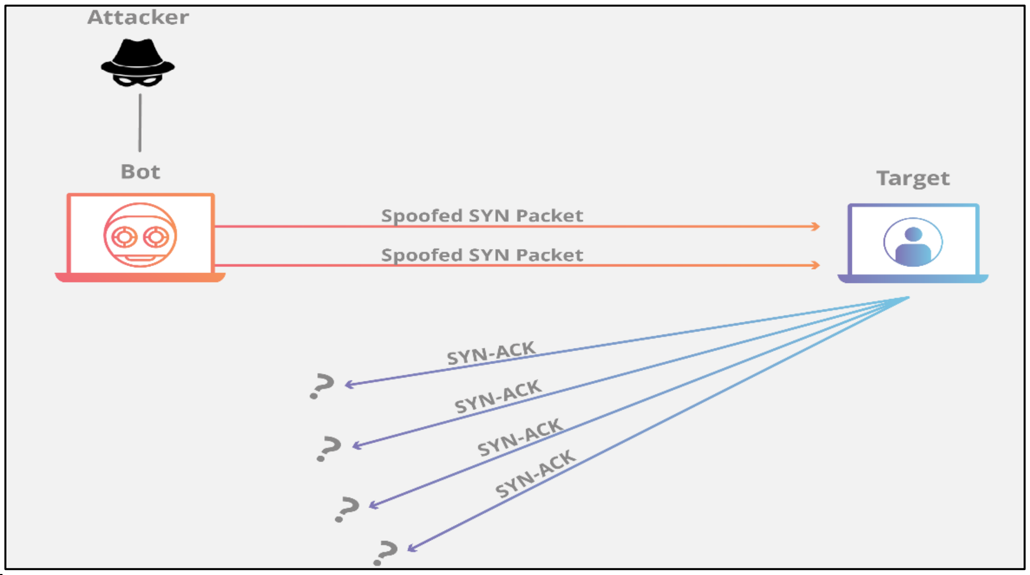


Figure 1: SYN flood attack

A SYN flood, also known as a TCP SYN flood, is a type of denial-of-service (DoS). SYN flood initiates a connection request to a server but does not finish the handshake with the server. This process continues until all open ports are flooded with requests and none are accessible for authorized users to connect to them. This attack takes advantage of the TCP handshake. SYN flood is a series of interactions between two computers in which they attempt to establish a network connection by delivering a large number of TCP "Initial Connection Request" SYN packets with fake source IP addresses to the target. The target machine answers to each connection request and then waits for the last step in the handshake, which never happens, causing the target's resources to be depleted as a result of the prolonged wait. (Nakashima & Oshima, 2006)

3.3 Comparative Study

Table 1: Comparative Study

No.	Author	Title	Findings	Application
1	Nazwita & Ramadhani S. (2017)	Analisis Sistem Keamanan Web Server Dan Database Server Menggunakan Suricata	Suricata generates an alert when an attacker is found on the network and saved in the log file, when the same appears in Web Admin, and directs IPTables to block the attacker's internet address protocol (IP) address, thus disconnecting the attacker from the server.	• Suricata • MySQL • Superscan • Nikto • Nmap
2	Made et al., (2020)	Network Security Monitoring System on Snort with Bot Telegram as a Notification	Snort Intrusion Detection System effectively identified strange packet data throughout ten times of testing of two kinds of infiltration. There is a delay of 4,05 seconds while sending messages from Telegram. This monitoring system's prevention system may block IP addresses defined by port, protocol, and time during the block.	• Snort • Telegram
3	Iqbal (2021)	Intrusion Detection System on Suricata with Bot Telegram as A Notification System	Suricata can detect a presence threat from an attacker and able to obtain an alert from Telegram in real time.	• Suricata • Telegram • hping3

CHAPTER 3

METHODOLOGY

This chapter is divided into 4 subsections: (1) The research methodology, (2) The tools and software required, (3) The data gathering and (4) The project Gantt Chart and Milestone.

3.1 Research Methodology

An appropriate research methodology is essential for any project. A research methodology refers to the specific procedures or tactics that are utilized to identify, select, analyze, and evaluate data pertaining to an individual topic or subject. This will facilitate the creation of a software process model by establishing the sequence of activities of the process and the order in which they are executed.

In order to choose the most appropriate process model for a software project, it is necessary to consider the features of the software project. Therefore, the purpose of this study is to build an intrusion detection and notification system utilizing the Telegram messenger, as some elements must be known prior to development. Many studies have discovered several intrusion detection system features. These include the following:

- It must run continually without human supervision.
- It must observe deviations from normal behavior

These are the primary qualities of the intrusion detection system, and they are also included into this project. All of the requirements must be documented in order to make a decision on the process model. This will aid in the decision-making process throughout the process model selection procedure, which has been utilized for this project. Hence, the Agile Methodology, which is compatible with the development of intrusion detection and notification system has been selected as a guidance for the project initiative across its lifecycle.

3.1.1 Agile Methodology

Agile software development is a set of approaches to software development that are iterative and incremental. The agile process is an effective fit for this project. It enables adaptive planning, evolutionary development, iterative delivery, and a quick and flexible reaction to change. Agile facilitates effective monitoring of project progress throughout the timeline by breaking activities down into tiny pieces and continuously developing the project through design, coding, and testing stages until it is completed. This will aid in identifying areas for improvement and raising the likelihood that the project's outcomes will be successful.

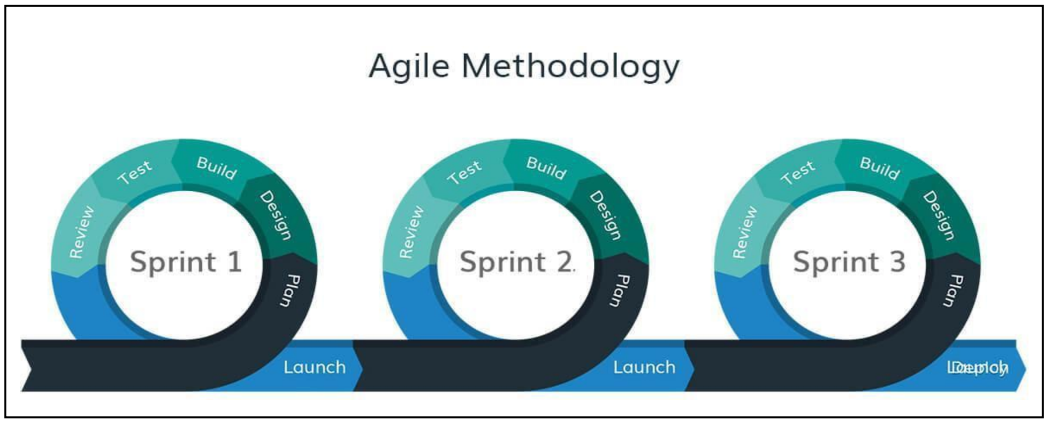


Figure 2: Agile Methodology

A distinct approach is used by the Agile methodology, which breaks the project workload into smaller sub-tasks known as sprint. For each sprint, the duration is between two to four weeks, which allows for an in-depth examination and evaluation of all areas of development throughout the project's lifecycle.

3.2 Project Activities

3.2.1 Planning Phase

Every project planning is critical as it is at the heart of the project life cycle. Determine the goals and plan of the project as it helps to know on what goals and that must be achieved at the end of the project development. This is important as this project have their own objectives that need to be achieved at the end of the project. The planning phase of this research lasted almost three weeks in May 2021. The work completed during this phase included research, literature review, the preparation of project deliverables, and the gathering of requirements. This is to validate the problem statement by detecting gaps in existing research papers and determining the paint points caused by IDS to its users. Several discussion sessions with the FYP supervisor have taken place throughout this planning phase, in order to get professional feedback and advice on the research topic and scope of the project. In order to ensurethat this project is accomplished within a short period of time, the scope will be limited to the implementation of a Telegram bot and how it interacts with Suricata.

This phase culminates with the identification of requirements necessary to accomplish project's goals. The criteria include the following:

- A working IDS notification system that capable of notifying user or IT administrators in the event of a cyber-attack.
- Extensive testing to verify accuracy, which has always been the most challenging aspect of IDS.

3.2.2 Analysis Phase

The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished. The deliverables are broken down into more specific business requirements, and here is where the mostwork is done. The following table summarizes the outcomes of the project'sobjectives:

Table 2: Requirments & Implementations

REQUIREMENTS	IMPLEMENTATIONS
A working IDS notification system that capable of notifying user or IT administrators in the event of a cyber-attack.	Develop a notification system using Telegram bot that will be able to retrieve information of an intrusion attack. Furthermore, to assist users, the system must be capable of generating readable and understandable messages.
Extensive testing to verify accuracy, which has always been the most challenging aspect of IDS	To guarantee that the system is functional, extensive testing will be carried out, with the emphasis only on one type of attack, the SYN flood attack.

3.2.3 Design Phase

Next, during the design phase, the requirements will be transformed into a thorough specification for the system's design. The following are the results of the design phase:

3.2.3.1 System Architecture

Suricata and its interface with Telegram were used as the basis for this study. The design of this system will continuously monitor suspicious data packages and the flow of data entering the computer server through Suricata. The following diagram illustrates and explains the project's topology:

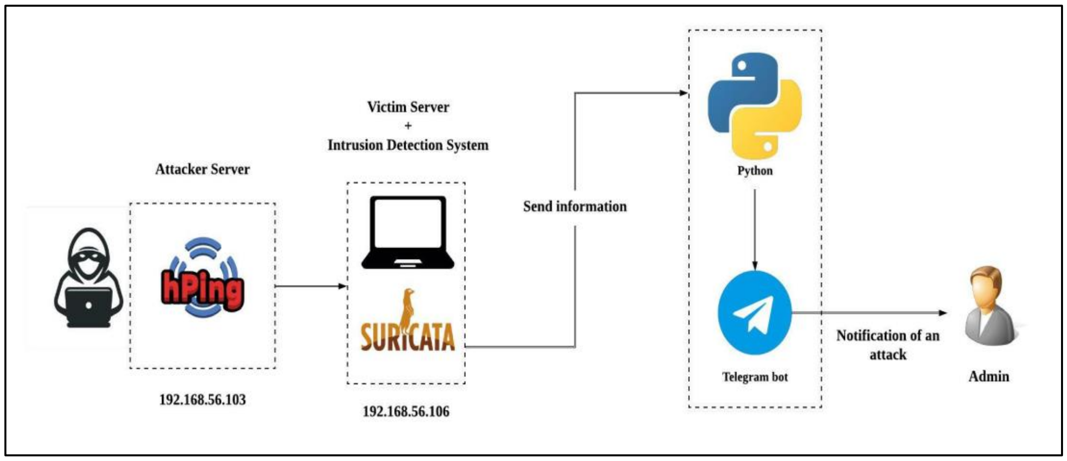


Figure 3: Topology

To begin, this study involves the creation of two virtual machines, one of which will function as the attacker server and the other as the victim. On the attacker's site with the IP address (192.168.56.103), it will send modified packets to the victim computer with the IP address (192.168.56.106) using Hping3 to produce a SYN flood attack, a type of DoS attack. Suricata IDS is installed on the victim computer, which sniffs the network for malicious packets and determines whether they are harmful or not. Each detection result will be saved to the fast.log log file. Python will read the fast.log file and trigger a notification alert through Telegram, notifying the user or administrators of an intrusion on the network.

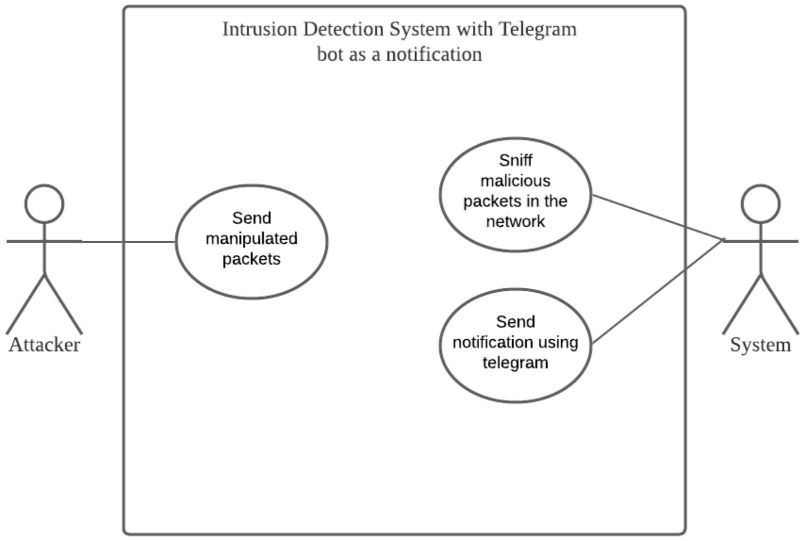

Intrusion Detection System with Telegram bot as a notification
Attacker
Send manipulated packets
Sniff malicious packets in the network
Send notification using telegram
System

Figure 4: Use Case Diagram

3.2.3.2 Use Case Description

Table 3: Use Case Description 1

Send manipulated packets	
High-level description	The attacker will try to send SYN flood attack to the victim computer through the network.
Actors	Attacker
Pre-conditions	1. Both attacker and victim need to turn on 2. Victim's network must be accessible to commit the attack
Post-conditions	1. Victim's network will be attack with malicious packets such as SYN flood
Flow of events	1. Run command: sudo hping3 -S --flood -V -p 80 192.168.56.106

Table 4: Use Case Description 2

Sniff malicious packets in the network	
High-level description	Suricata will sniff packets passing over the victim's network in order to identify intrusions or suspicious activity.
Actors	System
Pre-conditions	Installed Suricata on the victim's computer
Post-	Fast.log file will be populated with logs obtained from network

conditions	sniffing.
Flow of events	To start Suricata, run command: 1. rm -rf /var/run/suricata.pid 2. sudo suricata -D -c /etc/suricata.yaml -I enp0s8 To check fast.log, run command: 3. sudo nano /var/log/suricata/fast.log

Table 5: Use Case Description 3

Send notification using Telegram	
High-level description	Telegram bot will notify users through the Telegram app when an attack occurs on the network.
Actors	System
Pre-conditions	1. Suricata must be running to detect the network 2. Attacker must send attack on the network 3. AlertBot must be started
Post-conditions	1. Notification of the attack will be sent to telegram
Flow of events	Run command: 1. alertBot.py

3.2.3.3 Telegram Messenger API

Telegram Messenger has an API (Application Program Interface) that allows developers to build apps that interface with Telegram Messenger. Through a bot, Telegram Messenger talks with the system server. To be permitted to do Telegram Messenger-related tasks such as sending or receiving messages, the user must register their own bot. Here is a flowchart outlining the steps involved in constructing an individual Telegram bot. The user must first download Telegram on their smartphone and then enter the necessary code to create a Telegram bot. Enter the username of the bot program, then the Telegram server will reply with a bot token, which will be used to connect the bot's script to the Telegram server.

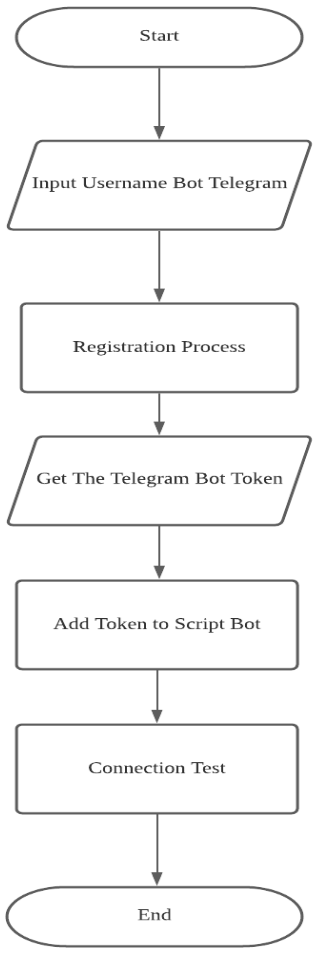


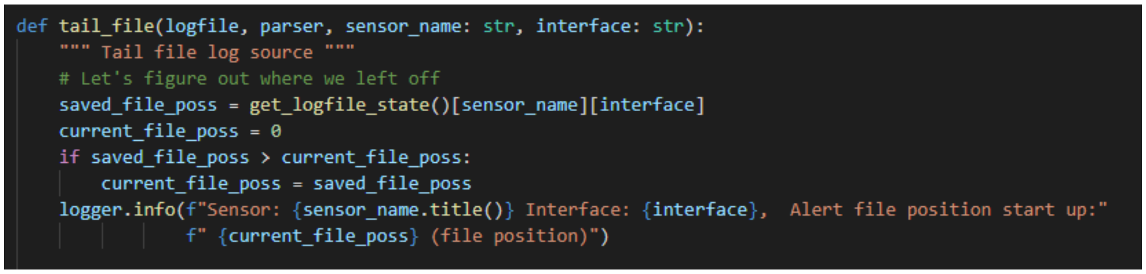
Figure 5: Telegram Messenger API

3.2.4 Development Phase

For the development phase, this is the time when the actual coding will take place, during which the precise logical information recorded in the previous phase will be transformed into machine-executable form. In this research project, the activities involved integration between Telegram Messenger with Suricata.

Suricata detects suspicious packages and sends out Telegram Notifications when they are detected. The Telegram Messenger bot is programmed to deliver notifications to the system administrator in accordance with the scripts. Developing, testing, and fixing the system will take the most timeduring this phase, which will need many iterations throughout the process.

Understanding the code:



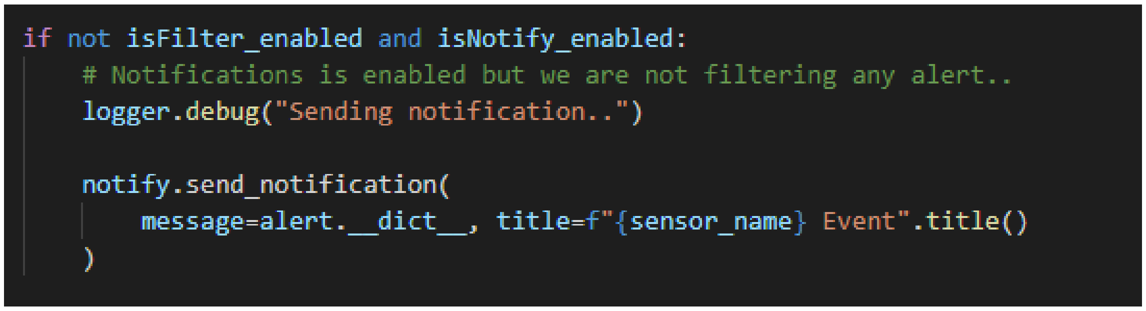
```
def tail_file(logfile, parser, sensor_name: str, interface: str):
    """ Tail file log source """
    # Let's figure out where we left off
    saved_file_poss = get_logfile_state()[sensor_name][interface]
    current_file_poss = 0
    if saved_file_poss > current_file_poss:
        current_file_poss = saved_file_poss
    logger.info(f"Sensor: {sensor_name.title()} Interface: {interface},  Alert file position start up:"
                f" {current_file_poss} (file position)")
```

Figure 6: Understanding Code 1

This section requires us to determine where we left off once we quit using get log file state( ), which returns the previous save file location. Current file poss will now be 0 and if saved file poss exceeds current file poss, the saved file poss will be the most recent.

logger.info( )

logger.info is used to Log an info message. This method is used to forward logs to all the registered output Handler objects. The message that will log is the sensor name, the interface and the file position.



```python
if not isFilter_enabled and isNotify_enabled:
    # Notifications is enabled but we are not filtering any alert..
    logger.debug("Sending notification..")

    notify.send_notification(
        message=alert.__dict__, title=f"{sensor_name} Event".title()
    )
```

Figure 7: Understanding Code 2

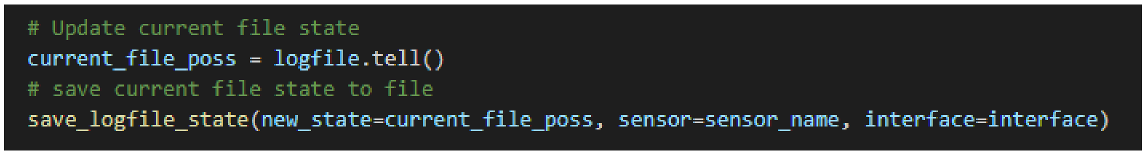
In this section, the notification is enabled, the filter is deactivated. This has no impact since we have enabled notifications but not activated alert filtering.

Logger.debug("sending notification..")

The main concept behind log levels is that we want to be able to modify the degree of information in the logs based on the scenario. For instance, if there is a need in attempting to diagnose a problem, we would want a very verbose log. We may also want to view warnings and mistakes in production. The log level for each component of the system is often controlled by a configuration file setting, making it simple to alter. The code would have a variety of logging statements with varying degrees of detail. To output the value of a variable at any point in time, we can use Logger.debug.

notify.send_notification ( )

This function will send an alert message with the alert's title, the sensor it utilizes, and the event.



```python
# Update current file state
current_file_poss = logfile.tell()
# save current file state to file
save_logfile_state(new_state=current_file_poss, sensor=sensor_name, interface=interface)
```

Figure 8: Understanding Code 3

After notification is sent, the logfile must be updated and saved as a current file state so that we can remember the last part of information that we see before we exit.

3.2.5 Testing Phase

Before the system is made available to the general public, it will be subjected to extensive testing. It must go through a rigorous testing procedure to verify that the system notification functions as intended.

Firstly, system testing. System testing is the first level in which the complete application is tested as a whole. The goal at this level is to evaluate whether the system has complied with all of the outlined requirements and to see that it meets quality standards. The system will be tested by the FYP supervisor. From there, feedback will be given based on the performance of the system in meeting the technical, functional, and business requirements that was predefined earlier. The testing is performedin a realistic environment as near to production as possible.

3.3 Tools and Software

The development of the mobile application's prototype for this study required some of the software tools to ensure that the development phase can be done successfully as per planning. Below are some of the software tools that be used to develop the prototype of this project:

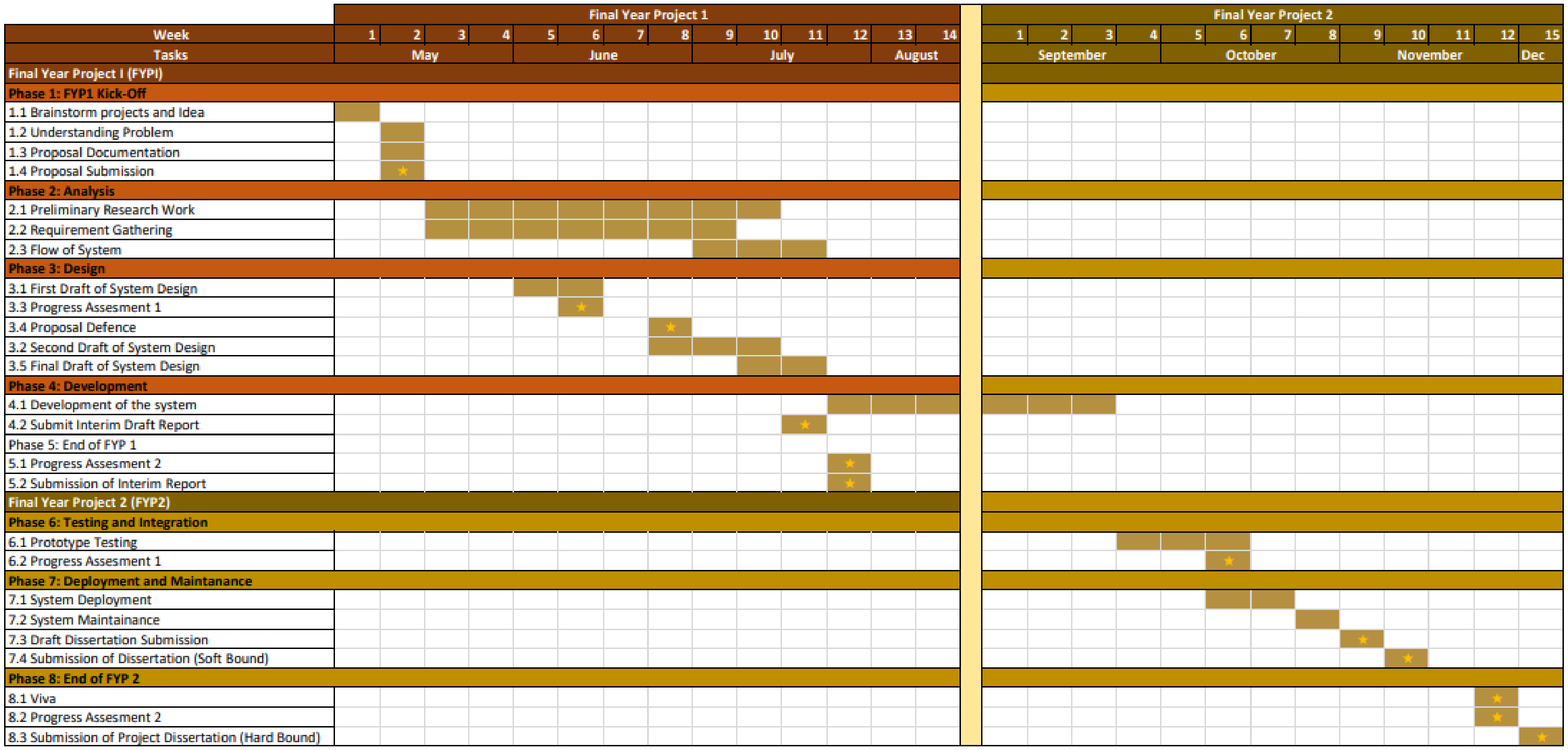
Table 6: Tools and Software

SOFTWARE	DESCRIPTION
	Visual Studio Code Visual Studio Code is a simplified code editor that supports debugging, task execution, and version control. This software is a lightweight but capable source code editor used by many programmers on various projects. It operates on Windows, MacOS, and Linux. This project employs this tool as part of the source code editor throughout the whole development period to help organize the source code.
	VirtualBox VirtualBox is a virtualization software for x86 and x86-64 hardware aimed at server, desktop, and embedded applications. It enables users and administrators to seamlessly run numerous guest operating systems on a single host. VirtualBox is utilized to build two virtual machines for this project, one for the Attacker and one for the Victim where simulation will be conducted on those machines.

	Wireshark Wireshark acts as a network protocol analyzer, or a software that collects packets from the network. Wireshark is being used in this project to examine packets that are coming in from inside the victim network in order to determine whether or not a SYN flood attack is occurring.
SURICATA	**Suricata** It is an open-source network threat detection engine that includes functions such as intrusion detection (IDS), intrusion prevention (IPS), and network security monitoring. It performs very well while doing deep packet inspection and pattern matching, making it particularly valuable for threat and attack detection and identification. Suricata was used in this project to identify incoming packets on the Victim's network, where it was able to capture those packets and produce log files.
hPing	**hping3** hping3 is a network utility that allows you to send custom ICMP/UDP/TCP packets and show target responses in the same way as ping does with ICMP answers. For this project, hping3 is used to create SYN flood attack by manipulating the packets.

	Telegram Telegram is a messaging software that is extensively used by many people due to the fact that it provides privacy, encryption, and an open-source API to users. The messaging app Telegram is being utilized in this project to notify/alert users about an intrusion or attack on the network.

3.4 Gantt chart



	Final Year Project 1														Final Year Project 2												
Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	1	2	3	4	5	6	7	8	9	10	11	12	15
Tasks	May				June				July				August		September				October				November				Dec
Final Year Project I (FYPI)																											
Phase 1: FYP1 Kick-Off																											
1.1 Brainstorm projects and Idea	■																										
1.2 Understanding Problem		■																									
1.3 Proposal Documentation		■																									
1.4 Proposal Submission		★																									
Phase 2: Analysis																											
2.1 Preliminary Research Work			■	■	■	■	■	■	■	■																	
2.2 Requirement Gathering			■	■	■	■	■	■	■																		
2.3 Flow of System									■	■	■																
Phase 3: Design																											
3.1 First Draft of System Design					■	■																					
3.3 Progress Assesment 1						★																					
3.4 Proposal Defence								★																			
3.2 Second Draft of System Design								■	■	■																	
3.5 Final Draft of System Design										■	■																
Phase 4: Development																											
4.1 Development of the system												■	■	■	■	■	■										
4.2 Submit Interim Draft Report											★																
Phase 5: End of FYP 1																											
5.1 Progress Assesment 2												★															
5.2 Submission of Interim Report												★															
Final Year Project 2 (FYP2)																											
Phase 6: Testing and Integration																											
6.1 Prototype Testing																		■	■	■							
6.2 Progress Assesment 1																				★							
Phase 7: Deployment and Maintanance																											
7.1 System Deployment																				■	■						
7.2 System Maintainance																						■					
7.3 Draft Dissertation Submission																							★				
7.4 Submission of Dissertation (Soft Bound)																								★			
Phase 8: End of FYP 2																											
8.1 Viva																										★	
8.2 Progress Assesment 2																										★	
8.3 Submission of Project Dissertation (Hard Bound)																											★

Figure 9: Gantt Chart

CHAPTER 4

RESULTS AND DISCUSSION

This chapter is categorized into 4 subsections: (1) Simulation testing, (2) Monitoring, (3) Results, and (4) Discussion.

4.1 Simulation Testing

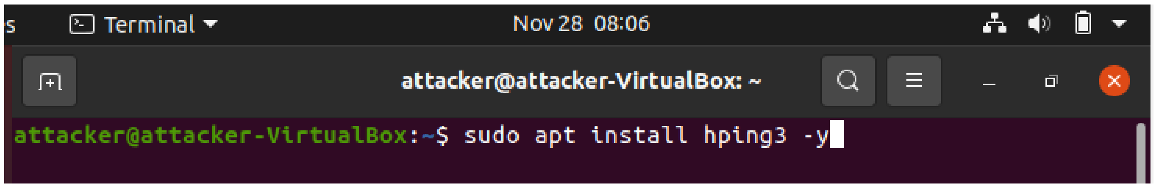
Testing will be carried out in order to implement the Suricata IDS system, which will be responsible for identifying packets that enter the network. In this case, we will establish two virtual machines to imitate the procedure, one for the "Attacker" and another one for the "Victim," using the IP addresses 192.168.56.103 and 192.168.56.105 respectively. Suricata will be installed and configured on the victim site in order to monitor the network.

4.1.1 Attacker

Firstly, we must install hping3 on the attacker virtual computer. Hping3 may be used in a variety of ways, including for network and host testing. As previously stated, the tool hping3 enables the transmission of altered packets. This tool enables user to manipulate the size, number, and fragmentation of packets in order to overwhelm a target and circumvent or attack networks. Hping3 is helpful for security and capability testing and it may be used to determine the effectiveness of firewalls and the capacity of a server to handle a large number of packets.

4.1.2 hping3 Installation

To install hping3, execute the following command:

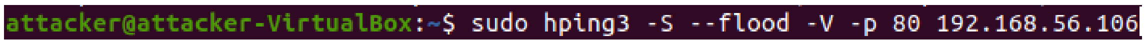


```
attacker@attacker-VirtualBox:~$ sudo apt install hping3 -y
```

Figure 10: hping3 Installation

4.1.3 Testing for DoS Attack (SYN flood)

Next, after installing hping3, the attacker can now use hping3 to conduct testing for DoS attacks using SYN flood. The attacking test will be conducted according to the hping3 rule, which is as follows:



```
attacker@attacker-VirtualBox:~$ sudo hping3 -S --flood -V -p 80 192.168.56.106
```

Figure 11: hping3 Attack Command

Table 7: hping3 Attack Command

RULES	DESCRIPTION
hping3	The tool used to simulate the attack
-S	Set SYN flag
--flood	Sent packets as fast as possible.
-V	Verbose mode
-p	Destination port
Domain/IP	Destination IP address

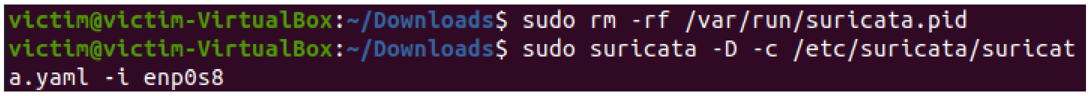
The attacker will now send SYN flood packets to the target port 80 with the destination IP address specified (192.168.56.106).

4.1.4 Victim

Suricata, Wireshark, and Visual Studio Code, among other things, must be installed on the victim's virtual computer before the machine may be used. In this computer, Suricata will be required since it will operate as a sniffing tool, listening to the victim's network computer for traffic coming in and out. Wireshark, on the other hand, will be utilized when the testing begins since it is capable of examining packets that are arriving from inside the victim network in order to detect whether or not a SYN flood assault is taking place. Moreover, Visual Studio Code is needed in the computer to run alertBot code as it will be trigger when the fast.log file capture data in the network.

After completing all the installation, we can start running everything to start testing.

Start running Suricata:



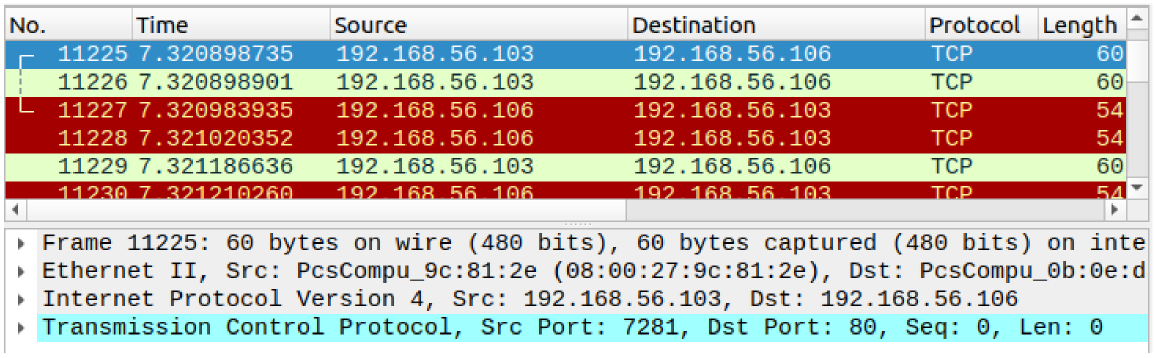
```
victim@victim-VirtualBox:~/Downloads$ sudo rm -rf /var/run/suricata.pid
victim@victim-VirtualBox:~/Downloads$ sudo suricata -D -c /etc/suricata/suricat
a.yaml -i enp0s8
```

Figure 12: Suricata Run Command

Table 8: Suricata Run Command

RULES	DESCRIPTION
Suricata	The tool used to detect the network
-D	Allows you to run at the background and you will be able to use the console for other tasks without disturbing the engine running.
-c	Path to configuration file. (/etc/suricata/suricata.yaml)
-i	Following the -i option, you may specify the interface card from which you want to sniff packets. This option will attempt to acquire data using the most efficient manner possible. In this case, the interface for the victim is enp0s8.

Start running Wireshark:



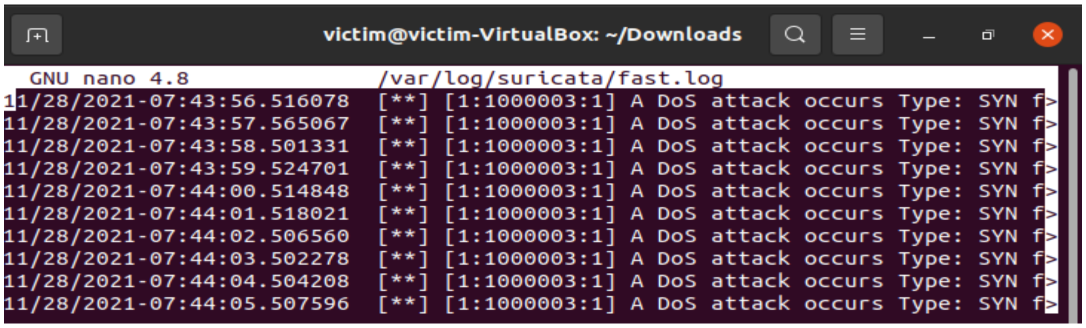
```
No.    Time         Source          Destination     Protocol  Length
11225 7.320898735   192.168.56.103  192.168.56.106  TCP       60
11226 7.320898901   192.168.56.103  192.168.56.106  TCP       60
11227 7.320983935   192.168.56.106  192.168.56.103  TCP       54
11228 7.321020352   192.168.56.106  192.168.56.103  TCP       54
11229 7.321186636   192.168.56.103  192.168.56.106  TCP       60
11230 7.321210260   192.168.56.106  192.168.56.103  TCP       54

Frame 11225: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on inte
Ethernet II, Src: PcsCompu_9c:81:2e (08:00:27:9c:81:2e), Dst: PcsCompu_0b:0e:d
Internet Protocol Version 4, Src: 192.168.56.103, Dst: 192.168.56.106
Transmission Control Protocol, Src Port: 7281, Dst Port: 80, Seq: 0, Len: 0
```

Figure 13: Wireshark Network Traffic

When an attacker attacks, Wireshark may detect the victim's network traffic. The following result depicts a SYN flood attack, where two computers try to establish a network connection by sending a huge number of TCP "Initial Connection Request" SYN packets with bogus source IP addresses to the target. The target computer responds to each connection request and then waits for the final handshake step, which never occurs, depleting its resources.

Fast.log



```
victim@victim-VirtualBox: ~/Downloads

  GNU nano 4.8                    /var/log/suricata/fast.log
11/28/2021-07:43:56.516078  [**] [1:1000003:1] A DoS attack occurs Type: SYN f>
11/28/2021-07:43:57.565067  [**] [1:1000003:1] A DoS attack occurs Type: SYN f>
11/28/2021-07:43:58.501331  [**] [1:1000003:1] A DoS attack occurs Type: SYN f>
11/28/2021-07:43:59.524701  [**] [1:1000003:1] A DoS attack occurs Type: SYN f>
11/28/2021-07:44:00.514848  [**] [1:1000003:1] A DoS attack occurs Type: SYN f>
11/28/2021-07:44:01.518021  [**] [1:1000003:1] A DoS attack occurs Type: SYN f>
11/28/2021-07:44:02.506560  [**] [1:1000003:1] A DoS attack occurs Type: SYN f>
11/28/2021-07:44:03.502278  [**] [1:1000003:1] A DoS attack occurs Type: SYN f>
11/28/2021-07:44:04.504208  [**] [1:1000003:1] A DoS attack occurs Type: SYN f>
11/28/2021-07:44:05.507596  [**] [1:1000003:1] A DoS attack occurs Type: SYN f>
```

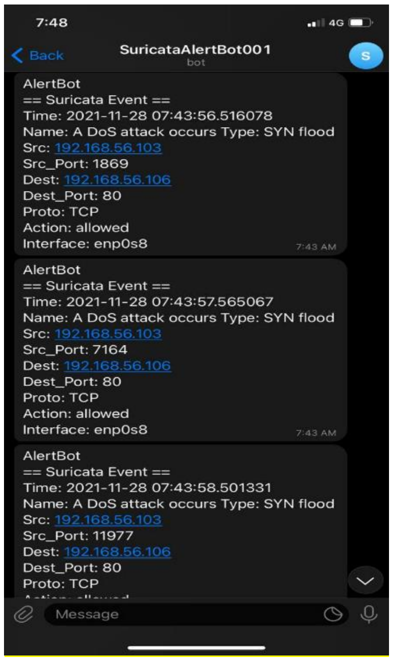
Figure 14: Fast.log

When Suricata is enabled, as well as when the attacker is attacking, Fast.log will record every packet received. This fast.log file will serve as the triggering mechanism for alertBot in the python script, which will then send a notice to the Telegram app, notifying the user with the fields that have been configured earlier.

4.2 Monitoring

The monitoring test was carried out to see if Suricata was able to listen on the victim's network and whether the telegram bot was able to receive the logs and generate a notification on the telegram messenger app in order to inform or alert the user.

4.2.1 Telegram Notification



```
AlertBot
== Suricata Event ==
Time: 2021-11-28 07:43:56.516078
Name: A DoS attack occurs Type: SYN flood
Src: 192.168.56.103
Src_Port: 1869
Dest: 192.168.56.106
Dest_Port: 80
Proto: TCP
Action: allowed
Interface: enp0s8                    7:43 AM

AlertBot
== Suricata Event ==
Time: 2021-11-28 07:43:57.565067
Name: A DoS attack occurs Type: SYN flood
Src: 192.168.56.103
Src_Port: 7164
Dest: 192.168.56.106
Dest_Port: 80
Proto: TCP
Action: allowed
Interface: enp0s8                    7:43 AM

AlertBot
== Suricata Event ==
Time: 2021-11-28 07:43:58.501331
Name: A DoS attack occurs Type: SYN flood
Src: 192.168.56.103
Src_Port: 11977
Dest: 192.168.56.106
Dest_Port: 80
Proto: TCP
```

Figure 15: Telegram Notification

Figure above demonstrates that the Telegram Bot notification worked effectively. The Telegram Bot gave a real-time notification alert to the Telegram app, and it provided information that was consistent with the rules that were previously set up in the dos.rules and the output in the fast.log. Information displayed by the AlertBot includes:

- Time of the attack
- Name of the attack

- Type of attack
- Attacker's IP address
- Attacker's source of port
- Victim's IP address
- The destination of the port that it wishes to attack
- The protocol used

4.3 Results

Based on the testing that has been done to determine the effectiveness of the Suricata intrusion detection system, this study will focus on detecting any network attack and estimating the time it will take to send a Telegram notification.

Table 9: Results

Attack Method	Type of Attack	Attack Time	Suricata Detection	Telegram Notification Time	Explanation
DoS attack	SYN flood	07:43:56	07:43:56	07:43:56	Successful in detecting & sending notification

Suricata IDS is capable of detecting network attack, and the Telegram bot can send notifications or alerts to the Telegram app in real-time, as shown in the table. The conclusion we can draw from this finding is that users or IT administrators will be able to take immediate action after they have been alerted, which will prevent more detrimental things from occurring if they are not informed.

4.4 Discussion

Businesses are increasingly recognizing the need of cyber security protection. Because of the current COVID-19 pandemic, which has forced many people to work from home, the danger landscape has grown significantly. As a result, organizations must build an Intrusion Detection System that is capable of monitoring for security events and network threats. As a consequence of the results above, several areas need improvement. There are a few constraints and challenges that have arisen throughout the creation of this project.

- Inadequate RAM to run 2 virtual machines
- Fast.log unable to log incoming packets

Because of all of the concerns, the development process is taking longer than it should, but the problem has been resolved to the best of ability. Even after the system has been completely established, there is still a lot of opportunity for improvement in the future.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

To summarize, cyber-attacks have the potential to damage businesses and gain unauthorized access to digital networks. Additionally, they may result in the theft of critical and sensitive data, interrupt phone and computer networks, and paralyze systems, making data unavailable. The threats certainly exist, and they are getting increasingly potent and frequent. Thus, we must enhance the intrusion detection system to ensure that it is capable of detecting and responding to new threats. The following three major objectives were accomplished successfully:

Objective 1: To investigate unusual activities and possible attacks on a network using an intrusion detection system (IDS).

- With the help of Hping3, this project has been successfully designed to analyze unusual activities and possible attacks on a network using an intrusion detection system (IDS). The DoS attack with SYN flood type has been used to explore the network.

Objective 2: To develop a system notification system using Telegram Messenger.

- The Telegram SuricataAlertBot001 contains all of the elements that it needs to capture in order to tell users that an attack is taking place by sending them a notification.

Objective 3: To test and evaluate the effectiveness of the system.

- This project has been tested and evaluated successfully, and both Suricata IDS and telegram notification have shown to be effective in detecting and sending alerts using the telegram app.

5.2 Recommendation

Overall, there are still a few parts of this project that need to be addressed in order to make it more reliable and user friendly in the long term. Further research and testing of the intrusion detection system, which uses a Telegram bot as a warning system, should be conducted with various forms of computer network attacks in the future. The greater the number of tests performed, the higher the chance that an IDS will recognize the same attack in the future. Next, to create a dashboard that enables users to see inbound and outbound network traffic. As a result, the user or IT administrator will be able to better monitor and understand the flow of their own network, as well as see a visual representation of how many attacks have occurred, enabling them to recognize that their network is not secure and is vulnerable to cyber-attack.

REFERENCES

Albin, E., & Rowe, N. C. (2012). A realistic experimental comparison of the Suricataand Snort intrusion-detection systems. *Proceedings - 26th IEEE International Conference on Advanced Information Networking and Applications Workshops,WAINA 2012*, 122–127. https://doi.org/10.1109/WAINA.2012.29

Almseidin, M., Alzubi, M., Kovacs, S., & Alkasassbeh, M. (2017). Evaluation of machine learning algorithms for intrusion detection system. *SISY 2017 - IEEE 15thInternational Symposium on Intelligent Systems and Informatics, Proceedings*, *Iv*, 277–282. https://doi.org/10.1109/SISY.2017.8080566

Nakashima, T., & Oshima, S. (2006). A detective method for SYN flood attacks. *First International Conference on Innovative Computing, Information and Control 2006, ICICIC'06*, 48–51. https://doi.org/10.1109/ICICIC.2006.3

Erlansari, A., Coastera, F. F., & Husamudin, A. (2020). Early Intrusion DetectionSystem (IDS) using Snort and Telegram approach. *Sisforma*, *7*(1), 21. https://doi.org/10.24167/sisforma.v7i1.2629

Sharma, M., Kaushik, A., Sangwan, A., & Scholor, M. (2012). Performance Analysis of Real Time Intrusion Detection and Prevention System using Snort. *1. Journal*, *1*(5),1–6.

Sulistya, I. M. A., & Sasmita, G. M. A. (2020). Network Security Monitoring System onSnort with Bot Telegram as a Notification. *International Journal of Computer Applications Technology and Research*, *9*(2), 059–064. https://doi.org/10.7753/ijcatr0902.1004

Thakkar, A., & Lohiya, R. (2020). A Review of the Advancement in Intrusion DetectionDatasets. *Procedia Computer Science, 167*(2019), 636–645. https://doi.org/10.1016/j.procs.2020.03.330