

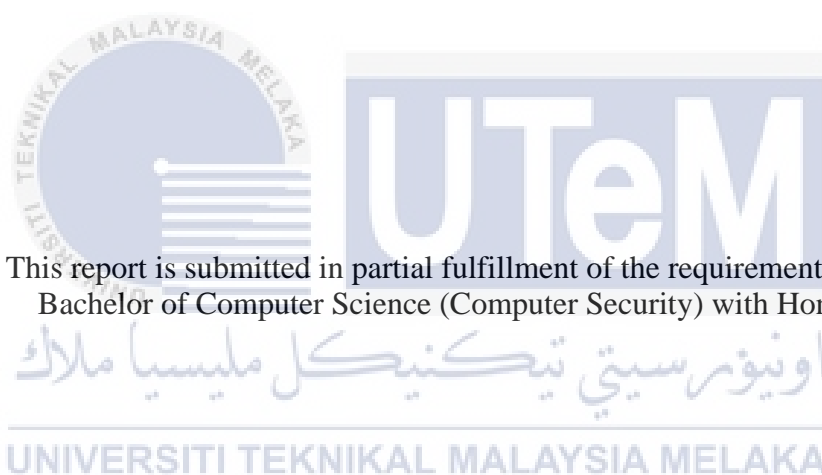
**INTRUSION DETECTION SYSTEM ALERTING SYSTEM USING
RASPBERRY PI**



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

INTRUSION DETECTION SYSTEM ALERTING SYSTEM USING
RASPBERRY PI

SITI NOR DIANA BINTI MAISITA



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I hereby declare that this project report entitled
**INTRUSION DETECTION SYSTEM ALERTING SYSTEM USING RASPBERRY
PI**
is written by me and is my own effort and that no part has been plagiarized
without citations.

STUDENT : SITI NOR DIANA BINTI MAISITA Date : 15 September 2021



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

I hereby declare that I have read this project report and found
this project report is sufficient in term of the scope and quality for the award of
Bachelor of Computer Science (Computer Security) with Honours.

SUPERVISOR : Dr. NUR FADZILAH BINTI OTHMAN Date : 15 September 2021

DEDICATION

One and only one,

Thank you to our Almighty god Allah s.w.t to give me strength and idea on doing this project.

To my soul Mom and Dad,

Thank you always be there for me, fulfill my needs and always pray my best in this journey.

Thank you for believing me to go through this meaningful journey.

I hope both of you would proud of me.

I hope this will give meaningful gift as my successful battle in this journey.

To my lovely Supervisor,

Thank you for guide and encourage me to finish my final year project.

Thank you for spending your precious time and energy to give me the direction to finish this project.

Last but not least,

To all my friend who are with me on doing this project,

Thank you for giving me some idea when I am stuck and give some motivation when I lack spirit.

I hope we can achieve our dream together.

Thank you, I really appreciate it.

Love.

ACKNOWLEDGEMENTS

Most importantly, I might want to say thank a ton to our Almighty god Allah S.W.T that consistently with me when I was glad and battle. I truly appreciate for strength that given to finish and complete my last year project. Other than that, Thank Allah S.W.T show the way to aiding me a long excursion in my life as understudy.

An extraordinary appreciation will be given to my lovely supervisor Dr Fadzilah who has been giving a lot of direction in carrying out this task. Dr Fadzilah likewise gave me direction, consolation, and coordination all through my last task. I am grateful that Dr Fadzilah will forfeit time and energy in controlling me during the trouble of finishing this undertaking and giving advice and comment to make improvement in this ideal report.

I might want to thank all my lovely friends and lecturer who have never been debilitate and upheld as long as I am in UTeM. Furthermore, mother and father consistently care about me by giving all my adapting needs even though they are getting more established. I am the youngest child attempting to achieve their hope where they want to see their child succeed graduate as a degree student.

Finally, I want to thank the University of Technology Malaysia Melaka (UTeM) which gives a stage to creating and improving my abilities in the web climate of innovation. Also, remember to the personnel at the Faculty of Information Technology (FTMK) who sharpened their understudies' abilities without griping. I implore that Allah S.W.T will favour the prescribed procedures they have committed and make UTeM a focal point of greatness for Graduate.

ABSTRACT

Due to worldwide proliferation and rapid progress in Information Technology (IT), networking is the crucial state where everybody is using the network including the small business, office or home also affected. Statistic of cybersecurity cases has been rising to 82.5% during the MCO. 82% cases have been receiving reported from the home user and other. This a big value compared to the last year. This is because due to increasing use of technology during the Covid-19 pandemic. So, they need to secure their network to make sure all the data that has been stored in their personal computer are fully safe from any threat especially port scanning threat. This purpose of study to minimize risk getting attack by detecting port scanning threat. In this project, Snort, Barnyard2, and Telegram has been implemented. Raspberry Pi 3 Model B has been used for developing this system. If home user implementing this project, it can minimize chances getting network breach since all the alerts will be send to the user mobile phone in real time. User can take fast action to prevent the network. Moreover, user also can monitor the current packet incoming by viewing in the webpage. They can analyze number of packets. Hopefully, this project will give a better future although it does not eliminate all the cybercrime cases, but it will minimize the risk getting attack.

ABSTRAK

Oleh kerana penyebaran di seluruh dunia dan kemajuan pesat dalam Teknologi Maklumat (IT), jaringan adalah keadaan penting di mana semua orang menggunakan rangkaian termasuk perniagaan kecil, pejabat atau rumah juga terjejas. Statistik keselamatan siber meningkat kepada 82.5% semasa MCO. 82% kes telah diterima dilaporkan dari pengguna rumah dan lain-lain. Ini nilai yang besar berbanding tahun lalu. Ini kerana peningkatan penggunaan teknologi semasa pandemi Covid-19. Oleh itu, mereka perlu mengamankan rangkaian mereka untuk memastikan semua data yang telah disimpan di komputer peribadi mereka selamat sepenuhnya dari sebarang ancaman terutama ancaman pengimbasan port. Tujuan kajian ini untuk meminimumkan risiko mendapat serangan dengan mengesan ancaman pengimbasan pelabuhan. Dalam projek ini, Snort, Barnyard2, dan Telegram telah dilaksanakan. Raspberry Pi 3 Model B telah digunakan untuk mengembangkan sistem ini. Sekiranya pengguna rumah melaksanakan projek ini, ia dapat meminimumkan kemungkinan terjadinya pelanggaran rangkaian kerana semua makluman akan dikirimkan ke ponsel pengguna secara real time. Pengguna boleh mengambil tindakan pantas untuk mengelakkan rangkaian. Selain itu, pengguna juga dapat memantau kemasukan paket semasa dengan melihat di laman web. Mereka dapat menganalisis bilangan paket. Mudah-mudahan, projek ini memberi masa depan yang lebih baik walaupun tidak menghapuskan semua kes jenayah siber, tetapi akan mengurangkan risiko diserang.

TABLE OF CONTENTS

	PAGE
DECLARATION.....	II
DEDICATION.....	III
ACKNOWLEDGEMENTS.....	IV
ABSTRACT	V
ABSTRAK	VI
TABLE OF CONTENTS.....	VII
LIST OF TABLES	XII
LIST OF FIGURES	XIII
LIST OF ABBREVIATIONS	XV
CHAPTER 1: INTRODUCTION.....	16
1.1 Introduction.....	16
1.2 Problem Statement (PS).....	18
1.3 Research Question (RQ)	19
1.4 Research Objective (RO)	20
1.5 Project Scope	21
1.6 Project Contribution.....	21
1.7 Report Organization.....	22
1.8 Conclusion	23
CHAPTER 2: LITERATURE REVIEW.....	24
2.1 Introduction.....	24

2.2	Theory and Technical Background.....	25
2.2.1	Intrusion Detection System.....	25
2.2.1.1	Intrusion Detection Architectural Model.....	26
2.2.2	Snort.....	30
2.2.2.1	Snort rules.....	31
2.2.3	Port scanning.....	32
2.2.4	Raspberry Pi.....	32
2.3	Comparison of the Raspberry Pi.....	34
2.4	Comparison with the existing system.....	35
2.5	Project Solution.....	38
2.6	Conclusion.....	38
CHAPTER 3: PROJECT METHODOLOGY.....		39
3.1	Introduction.....	39
3.2	Project Methodology.....	40
3.3	Project Milestones.....	43
3.4	Conclusion.....	45
CHAPTER 4: DESIGN.....		46
4.1	Introduction.....	46
4.2	Problem Analysis.....	47
4.3	Analysis Requirement.....	48
4.3.1	Data Requirement.....	48
4.4	Software Requirement.....	49
4.4.1	Snort.....	49

4.4.2	Telegram	49
4.4.3	Barnyard2	50
4.5	Hardware Requirement	51
4.5.1	Raspberry Pi.....	51
4.5.2	Switch	52
4.6	Model Design.....	52
4.7	Conclusion	54
CHAPTER 5: IMPLEMENTATION.....		55
5.1	Introduction.....	55
5.2	Software Development Environment Setup.....	56
5.2.1	Package Dependent Libraries	56
5.2.2	Data Acquisition (DAQ).....	57
5.2.3	Snort.....	58
5.2.4	Rules	59
5.2.5	Barnyard2	59
5.2.6	Database.....	59
5.2.7	Telegram	60
5.2.8	Webpage	61
5.3	Implementation Status	63
5.4	Conclusion	64
CHAPTER 6: TESTING		65
6.1	Introduction.....	65
6.2	Test Plan.....	66
6.2.1	Test Organization.....	66
6.2.1.1	Attacker.....	66

6.2.1.2	User and Administrator.....	66
6.2.1.3	Network Engineer & System Developer	66
6.2.2	Test Environment.....	67
6.2.3	Test Schedule.....	67
6.3	Test Strategy	68
6.3.1	Classes of Test	68
6.3.1.1	Functional Test	68
6.4	Test Design	68
6.4.1	Test Description.....	69
6.5	Test Result and Analysis.....	75
6.5.1	Raspberry Pi 3 And Mirroring Switch Connectivity	75
6.5.1.1	Attacker Side	75
6.5.1.2	Server Side.....	76
6.5.2	Alert To Database Testing.....	77
6.5.3	System Testing.....	78
6.5.3.1	Login Testing.....	78
6.5.3.2	Dashboard Testing.....	79
6.5.3.3	Add New User Testing	80
6.5.3.4	Threat Show Counter Testing.....	81
6.5.3.5	Logout Testing.....	82
6.5.4	Telegram Notification Testing.....	83
6.5.5	Analysis of Ram Usage When Execute All the Script Needed	84
6.6	Conclusion	85

CHAPTER 7: PROJECT CONCLUSION	86
7.1 Introduction.....	86
7.2 Project Summarization.....	87
7.3 Project Contribution.....	87
7.4 Project Limitation	88
7.5 Future Works	88
7.6 Conclusion	89
REFERENCES.....	90
APPENDIX A	91



LIST OF TABLES

	PAGE
Table 1.1 Problem Statement	18
Table 1.2 Research Question	19
Table 1.3 Research Objective.....	20
Table 2.1 Comparison between Raspberry Pi	34
Table 2.2 Comparison Previous Project.....	35
Table 5.1 Implementation status.....	63
Table 6.1 RPi 3 and Mirroring Switch Connectivity	69
Table 6.2 System Login Testing	70
Table 6.3 Add New User Testing	71
Table 6.4 Logout Testing	72
Table 6.5 Web Panel	73
Table 6.6 Telegram Alert.....	74

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF FIGURES

	PAGE
Figure 2.1 Functions of IDS	26
Figure 2.2 Selective phase in identifying anomaly activities	27
Figure 2.3 Classification of intrusion detection system	28
Figure 2.4 Raspberry Pi 3 Model B	33
Figure 3.1 Flowchart of the project	42
Figure 4.1 Flow System Architecture	47
Figure 4.2 Data Flow of the Project.....	48
Figure 4.3 Snort Logo	49
Figure 4.4 Bot Father Logo	50
Figure 4.5 Illustration function of Barnyard2.....	50
Figure 4.6 Raspberry Pi Model 3B	51
Figure 4.7 Tenda Switch.....	52
Figure 4.8 Design Architecture of the Project.....	53
Figure 4.9 Real Design	53
Figure 5.1 Software Flow.....	56
Figure 5.3 Dependency for MySQL Server	56
Figure 5.4 Dependency for library MySQL Client	56
Figure 5.5 Install Auto configure.....	57
Figure 5.6 Install MySQL client	57
Figure 5.7 Install libraries	57
Figure 5.8 Securing the Database	57
Figure 5.9 Install dependency for database	57
Figure 5.10 Install dependency for database	57
Figure 5.11 Install dependency for database	57
Figure 5.12 Install dependency for database	57
Figure 5.13 DAQ installation	57
Figure 5.14 Missing Dependencies.....	58

Figure 5.15 Missing Dependencies.....	58
Figure 5.16 Snort Missing Dependencies	58
Figure 5.17 Snort Missing Dependencies	58
Figure 5.18 Snort Missing Dependencies	58
Figure 5.19 Dependencies LuaJit.....	58
Figure 5.20 Snort configure by enabling sourcefire.....	58
Figure 5.21 Snort installation.....	59
Figure 5.22 sid message map	59
Figure 5.23 Barnyard with MySQL	59
Figure 5.24 Barnyard with snort	59
Figure 5.25 Database configuration.....	60
Figure 5.26 Log count	60
Figure 5.27 Telegram Identification.....	61
Figure 5.28 Webpage Script	62
Figure 5.29 Database Script	62
Figure 6.1 Illustrated the test environment	67
Figure 6.2 Port Scanning	75
Figure 6.3 Snort Alert Testing	76
Figure 6.4 Database Testing.....	77
Figure 6.5 System Testing	78
Figure 6.6 Login Testing.....	78
Figure 6.7 Dashboard	79
Figure 6.8 Add New User	80
Figure 6.9 Successfully Adding New User	81
Figure 6.10 Threat Show Counter Testing	81
Figure 6.11 Logout Testing.....	82
Figure 6.12 Telegram Alert	83
Figure 6.13 Size of RAM Without Any Process	84
Figure 6.14 Size of RAM With Process	84
Figure 6.15 RAM Usage Comparison.....	84

LIST OF ABBREVIATIONS

FYP	- Final Year Project
IDS	- Intrusion Detection System
RPi	- Raspberry Pi
IP address	- Internet Protocol Address
NIDS	- Network Intrusion Detection System
HIDS	- Host Intrusion Detection System
IT	- Information Technology
URL	- Uniform Resource Locator
TCP	- Transmission Control Protocol

CHAPTER 1: INTRODUCTION

1.1 Introduction

Due to worldwide proliferation and rapid progress in Information Technology (IT), networking is the crucial state where everybody is using the network including the small business, office or home also affected. In addition, nowadays, worldwide has been hit with the pandemic Coronavirus or known as Covid19. As the outbreak of coronavirus at the end of 2019 which required humankind to practice a social distancing as a part of prevention step in containing the virus from spreading. As regards to that matter, the government has announced a mandatory lockdown to the entire nation to break the chain of the virus where it prohibits travel in or out from the affected area. Hence, halt the business operation of most of the companies throughout the country. For the sake of business continuity, the management had decided to adopt the concept of telecommuting which enables the worker to work remotely from home. Thus, all the people need to work fully using their own internet. With the rapid proliferation of computer usage and network, security aspect became very critical. Especially in network vulnerability such as port scanning is the most common vulnerability in the network. Port scanning is the first step that attacker will take before launch the attack. As an example, the attacker will scan the IP address to check whether that host are alive or not, then the attacker will scan port that has been opened to breach the user's network or do other attack. Port scanning is a method for discovering hosts' flaws by sending port inquiries. An intrusion detection system (IDS) is one of the popular methods that can detect any suspicious activity within a network. Intrusion detection system (IDS) will be functioning as monitoring any traffic that seems suspicious and unusual activities. In other words, it will analyze the behaviors that has

been breach the access control policy that has been set up by administrator. Meanwhile, the user needs to be always aware about the network status either in a safe mode or being threaten. By using alert system, it can help user to minimize the risk.

Currently, the increasing number of attackers make the network became more unsecure for everyone. Nowadays, everybody has a smart phone, computer, laptop, and other gadget that connected to the internet to do the work and social such as using Facebook, WhatsApp, and other application to connected with each other. Internet are very essential to every people nowadays for each level including kids and senior citizens that surely do not have a good knowledge about the security. People will simply use the internet without thinking any destruction that maybe happen. People also will simply put the easy guess password that attacker can easily breach to the network. Then, the innovative people need to solve this kind of problem to minimize the cyber security cases.



1.2 Problem Statement (PS)

Internet has become the highest essential tools in this modern era. Computer networking become more attractive because of the application such as Local Area Network (LAN), Wireless Local Area Network (WLAN) and Wide Area Network (WAN) that is have been uses in various enterprises, security service, health care and other emergency services. Hence, exposure to intrusion activities has been incremented proportionally because of the increasing number of Internet users worldwide. Because of that, some users are willing to spend money to protect the network form a certain threat such as port scanning, however, there always have several people that do not have an enough or a lot of budgets but want to secure the network also. Since the statistic of cybersecurity cases has been rising to 82.5% during the MCO and more than have percent cases have been receiving reported from the home user and other reported by The Star news. Thus, to setup a network detector is costly. Moreover, nowadays many people lose their jobs and have some payroll deductions that make people hard to spend money to this network equipment. Next, most of the people are not have a good knowledge in IT especially in network and command prompt version with other type of OS such Linux OS. Furthermore, the status of the network is unknown because do not have any alert that can send to the user inform about the current status either in safe environments or has been attack. Table 1.1 shows the summarize of the problem statement.

Table 1.1 Problem Statement

PS	Problem Statement
PS1	Costly to setup a network detector to identify port scanning threat.
PS2	Snort log file are non-user friendly to understand.
PS3	Users are not aware that their network has been attack.

1.3 Research Question (RQ)

To solve the problem statement, it needs to come out with research question. The question arose when further intending to get know more about the research. In this development, we need to answer the question before can proceed the progress. The main problem in this project is about the cost, so that how this project could be solved to this problem. Then, how the user who does not have a good knowledge in IT to understand the snort log file, understand what ingoing in their network are. The last question in this study is about the user knowledge if there someone that are trying to port scanning to their network. This study carries out as the attempt to answer the research questions as follows in Table 1.2.

Table 1.2 Research Question

PS	RQ	Research Question
PS1	RQ1	How to minimize cost to detect the port scanning threat?
PS2	RQ2	How to understand the snort log file?
PS3	RQ3	How user know if their network has been threatened?

This development will be built based on the questions stated above. These questions are valuable in order to develop this project effectively. The questions arise will also be used to accomplish the objectives of this project.

1.4 Research Objective (RO)

Discussing to the project question mentioned previously, there are three objectives that will be used in answering all available questions. The objective will be to make sure that the project is extra structured as proposed. The objective of this project is to implement IDS on raspberry pi since the raspberry pi is affordable. Second objective is user can monitor the number of packets that has been send into their network by monitoring on the website which is user-friendly where user can easily understand. The last objective to achieve is to send the notification alert in Telegram to the user if the rules has been fulfilled. Table 1.3 shows that the summary of research objectives.

Table 1.3 Research Objective

PS	RQ	RO	Project Objective
PS1	RQ1	RO1	Implement IDS on raspberry pi.
PS2	RQ2	RO2	User can monitor and analyze the number of packets that incoming and outgoing into the network by using website.
PS3	RQ3	RO3	User will be receiving the notification in the real time if someone that are trying to port scanning (TCP port) into their network by using Telegram.

1.5 Project Scope

The targeted user for this project will be focused on basic to intermediate level of home user. The project will be driven by Raspberry Pi in making this project and Tenda switch to function as a port mirroring. On a network switch, port mirroring is used to send a duplicate of network packets viewed on one switch port or an entire VLAN to a network monitoring connection on another switch port. It assists administrators in keeping a close check on network performance and notifies them when problems arise. Moreover, Raspbian OS has been used to functioning in this project. Raspberry Pi 3 Model B with 32 GB memory will be use in this project. This project is working by detecting 3 devices and lasting forever if still in the same network. This project also will be detecting for TCP port scanning only.

1.6 Project Contribution

Nowadays, Internet is a complex entity consist of different users, resources, and networks. Today in the mission to ensure all people safe and in a trusted communication in daily usage was very crucial, it is because it need to maintain an intermediary level of security, especially during this pandemic. By implementing this project, it will give the benefit to the basic and intermediate home user. It is because user can easily understand if their network has been attacked because user can monitor by using the website. Moreover, user will be receiving an alert using Telegram. So that user can take any action to prevent their network. User can protect their network with low-cost.

As an attacker the first step to launch any attack they need to know which port are open or the vulnerability from that network. From that, the attacker will start to launch attack to breach the network information. Thus, providing a secure place to every environment in the network is a huge challenging issue. It has become critical since the attackers or intruders are very active to accessing our information over the network.

1.7 Report Organization

Chapter 1: Introduction

In this chapter, it has been discussing about the first planning in doing this project which are project objective, problem statement and project questions. Basically, this is our general information toward this project. Moreover, this chapter also the guide in pointing the main or important information that need to be mark as prior for this project.

Chapter 2: Literature Review

In this chapter, it will be focusing on discovering the previous study or researcher that has been done the similar project, in order to get the useful information on conduct this project. This section also discussing on the general fact that this project will be going. Thus, this section are explaining a first point to this project.

Chapter 3: Project Methodology

This section focuses on the how the methodology or process of the project. It also generally provides the Gantt Chart as the planning to finish this project.

Chapter 4: Design

This part may focus on analyzing the whole discussion over this project assessment with the structural design that was utilized for this project with the help of the need for computer hardware and computer software during the duration of this project.

Chapter 5: Implementation

The test technique will be discussed in detail in Chapter 5 in order to obtain the precision result. The results of this research will be collected to validate that it is comprehensive, and the results will be recorded to generate an assessment, which will then be equated to other methods. Characterize the software nature setup, software

structure management, and the implementation of each process that will be developed as part of the project operation.

Chapter 6: Testing and Analysis

This section summarizes the previous chapter's findings. Encapsulate and illuminate the outcomes of the implementation. Aside from that, it will illuminate how the results are examined from all angles and how they are accomplished.

Chapter 7: Project Conclusion

This part will be describing the summary the whole project in detail, project contribution, and project constraint. All of the processes that are implemented will be briefly detailed in this chapter. Hence, this part will explain any possible work that may be completed in the upcoming.

1.8 Conclusion

In conclusion, by implementing this intrusion detection system in small network, it can increase the level of network security. The general idea and overview of this project has been discussed in introduction. This is followed by the issue statements as well as the aims of this study, which are intended to answer the stated problem. The chapter continues with the scope of the research, which explains the platform and particular tools utilized. It finishes with the project's importance and projected output. The next chapter will discuss the literature review based on the related research on intrusion detection system using snort.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

In this chapter, it will discuss about the theory and technical background of the intrusion detection system, Snort, Telegram and Raspberry Pi. It also will clarify about the workflow of alerting system IDS using RPi. This chapter gave the aggregation from shifted creators and concentrates that have made this undertaking previously. This is the reason by looking into the choice of utilizing the correct procedures is essential to get the best involvement for this undertaking. This chapter also will contain about the previous related distribute data and material or article, past undertaking finding and research that identified with the target in this task.

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2.2 Theory and Technical Background

2.2.1 Intrusion Detection System

System that are functioning to detect of malicious activity such as break-ins, penetrations, and the other forms of computer abuse in a computer system are known as intrusion detection. These malicious intrusion or activities was fascinating from a computer security perspective (Jose et al., 2018). An intrusion detection system (IDS) is a system that detects malicious activity and attacks directed at a network or a single host system (Kyaw et al., 2016). IDS is also considered as a computer and network security application capable of collecting and analyzing data by intercepting inbound-outbound network traffic and identifying harmful actions using specified rules, followed by notifying the user (Tripathi & Kumar, 2018). Furthermore, IDS is also a network security programmed that collects and analyses network data by inspecting incoming and outgoing network traffic to recognize any malicious activity attempts in the network and to identify if the network or system has been hacked by an intruder (Jeremiah, 2019). Due to the gigantic network vulnerability, IDS has become an asset in form on securing the data integrity, confidentiality and availability (CIA triad). As we know, CIA is one of the basic goals to achieve the secure environment. The aim of IDS was to assistance the computer systems on how to deal with the attack, in between the IDS also collecting an information from more than a few different sources within the computer systems and network and comparability this information with previous patterns of discrimination as to whether there are attack or weakness.

2.2.1.1 Intrusion Detection Architectural Model

This IDS architectural model consists of four main functions namely; data collection, feature selection, analysis and action (Ahmed et al., 2018). Figure 2.1 illustrates the IDS Architectural model.

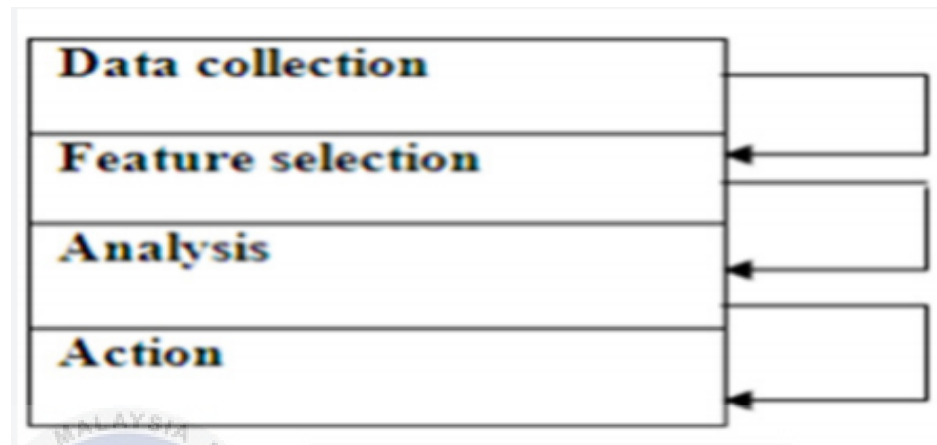


Figure 2.1 Functions of IDS

Based on Figure 2.1, all the elements have been discussed below: -

Data collection: This section was the initiation phase for IDS. It will capture and passes relation data from the monitored system to the following section for carry on the process, nevertheless this process is done by automatically. Sooner than being analyzed, the collected data has been sent to a designated file (Ahmed et al., 2018).

Feature Selection: In this phase, normally users' accomplishment has been dynamically monitored as soon as they have been logged into the system. This phase is functioned to sort out the distinct feature of huge data that has been captured from the network. An instance the data that has been taken is the source and destination IP addresses, type of protocol, header length and size. So that, from these, the users must be deployed the set of rules for managing the alerts to reduce the time to response. This activity has been illustrated in Figure 2.2 where the edge manager will compare the alert with the edge database to see whether it has been in the database or not. If there any similar characteristic, the system will send the general attack alert but if the data are not in the database yet, it will send the specific alert and save to the database for future references.

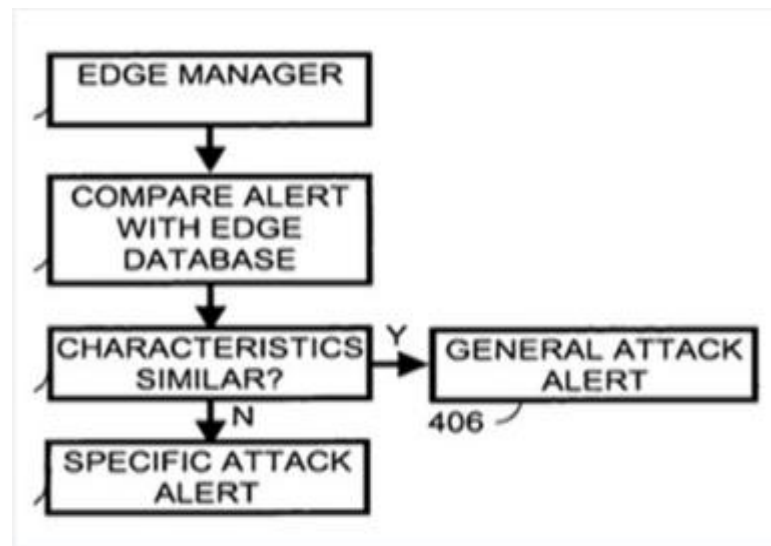


Figure 2.2 Selective phase in identifying anomaly activities

Analysis phase: In these sections, the system will analyze the data that has been collected to find any suspicious or unusual threat. In order to analyze this data profiling and pattern recognition technique has been uses. In response, all the captured activity which considerably deviate from the applied rules are refer as anomalous behavior and be mark as potential intrusions.

Action Phase: The last but not least, this phase was the phase that the system needs to action or reflex against the threat. During this phase, responsive mechanism has been implied. In order to resolve that issue, it has two best ways out which are by sending the alert to system administrator or directly enforce the action against the threat.

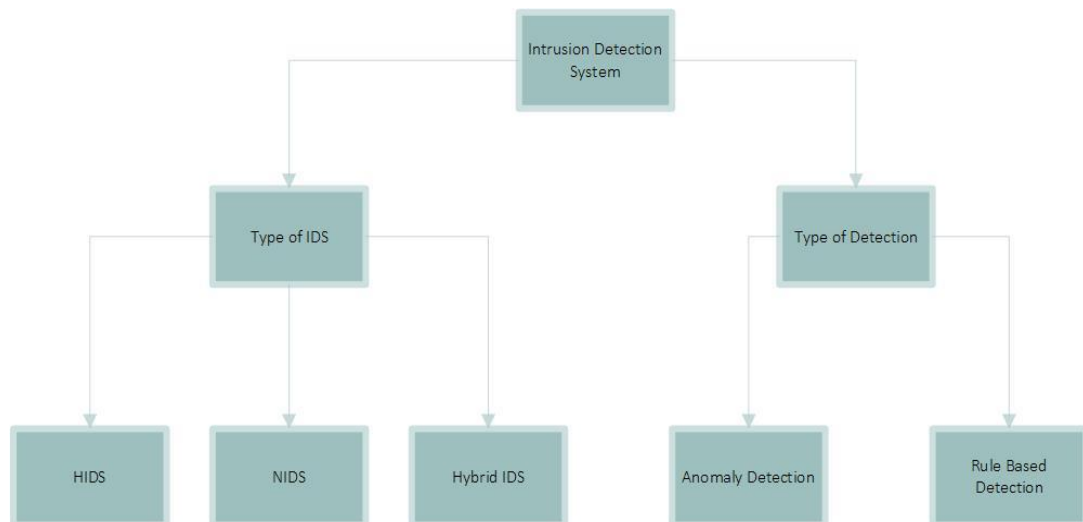


Figure 2.3 Classification of intrusion detection system

Based on figure 2.3, there are two class of IDS which are data collection technique and data analysis technique.

There are 3 types of IDS: -

a. Host-based IDS(HIDS)

This kind is installed on a single device, such as a server or workstation, where the data is processed locally to the computer and is collected from many sources. HIDS can employ both an anomaly detection system and a misuse detection system.

b. Network-based IDS(NIDS)

NIDS are strategically placed in network infrastructure. The NIDS can record and analyze data in order to identify known attacks by matching patterns or signatures in the database, or to detect unlawful behaviour by scanning traffic for unusual activity. Because it collects packets flowing across communication channels, NIDS are sometimes known as "packet sniffers." On the other words, the IDS will scan the whole network by monitoring all incoming and outgoing network packets in order to detect any malicious attempts or unauthorized network access.

c. Hybrid IDS

Management and alerting from both network and host-based intrusion detection devices, as well as providing a natural complement to NID and HID-central intrusion detection management

In addition, there are 2 types of detection: -

a. Anomaly Based Detection

Anomaly Based Detection is founded on user behaviour and network traffic patterns, which are used to form an opinion on network or host system patterns (Tripathi & Kumar, 2018). On the other words, anomaly based or known as behavior-based detection will continuously monitor the network, hosts, and users in order to gather statistics, as well as create warnings when unexpected or aberrant traffic is intercepted. It employs rules that are based on prior attacks. Based on M.Berge on Intrusion detection faq in SANS mentioned that this detection can be divided into two subcategories which is Threshold detection and Profile Based detection. Threshold detection establishes a threshold for how many times a user may do a certain activity or how many times a specific type of network data may transit the network; if the users exceed this threshold, an alarm is delivered. However, Profile Based Detection sets a baseline for usual behaviour and sends an alert if a user or network traffic deviates from this baseline to a significant degree. The baseline profile is established during the learning phase where the IDS learns about the ecosystem and creates a normal profile of the observed system, which can be networks, users, and other systems. The profile could be fixed or dynamic (Ahmed et al., 2018).

b. Rule Based Detection

Rule-based detection, also well-known as signature-based detection, compares a set of predetermined criteria to actual traffic to identify an

intruder or assault (Tripathi & Kumar, 2018). This detection also can be divided into two subcategories which are Anomaly detection and Penetration identification. Anomaly detection employs rules generated by examining preceding attack patterns or malicious traffic signatures, whereas penetration identification employs an expert system containing rules written by security experts that have been used when searching for suspicious behaviour in a network or on a host system. Normally, rule-based detection will use rules that are written by security experts. This signature works similarly to the virus scanner (Ahmed et al., 2018).

2.2.2 Snort

Snort is a software device that provides real-time analysis and sniffing of data packets on a communications network. It is one of the most extensively used and broadly utilized open-source NIDS software devices (Tripathi & Kumar, 2018). Snort has been developed by Martin Roesch (Karahana & Kaya, 2020). Snort may act as a packet analyzer and pre-processor as well as a detection engine-logger on Internet networks. It can also conduct protocol analysis, content searching, and matching, and it can identify a wide range of attacks and probes, including buffer overflows, CGI attacks, port scans, OS fingerprinting attempts, and more. Furthermore, snort is free and works on a wide range of systems. The data packet will be collected in real-time and stored in the 'tcpdump' format for further examination (Tripathi & Kumar, 2018).

Snort typically runs in three different modes which are sniffer mode, packet logger mode and intrusion detection system.

a. Sniffer mode.

Sniffer mode is a mode which snort provides access to the administrator to dump the data that is contained in the header and each data packet. To activate snort in sniffer mode, snort will be capturing the network data and display it on the monitor screen continuously until service can be stopped by using the shortcut keys Ctrl+C.

Then use the command “./snort -d” to run on all versions of snort or with use the command “./snort -dv” or “./snort -d -v” to display IP header packets (layer3) and TCP, UDP, and ICMP (layer 4) packets to the screen monitors.

b. Packet logger

The packet logger mode is different from the packet sniffer, it is because in this mode it will be log the data packet and header. This packet header and data packet will be written into host hard disk, specifically where the snort has been running.

c. Intrusion detection system

In this mode, snort uses rules to inspect the IP packets. When an IP packet matches the characteristic of a given rules, snort will display as alert. It also will detect any attack that has been carried out through the network. To uses this IDS mode, it is necessary to setup various rules that will distinguish a normal packet from a packet that carrying an attack.

2.2.2.1 Snort rules

Rules function by designating specific activities as illegal, and any activity observed that fits any of those rules is marked as an incursion or attack. Because the criteria used to identify malicious traffic or users may be established by an administrator, no prior knowledge of the attack is required. The primary advantage of utilizing rules versus signatures is that rules do not require any data to be gathered from previous attack (Kyaw et al., 2016). Snort rules are divided into two logical sections which is rule headers and rule options. It contains the rule's action, protocol, source and destination IP addresses and netmasks, as well as source and destination port information, in the rule header (Sabekti, 2018). While in rule options, it comprises of alert messages and evidence on which portions of the packet should be examined to decide if the rule action have to be performed. An example of snort rules as below (Tripathi & Kumar, 2018).


```
Alert TCP any 22 -> $HOME_NET any (msg: "TCP detected"; GID:1; sid:10000001; rev:001; classtype: tcp-event;)
```

2.2.3 Port scanning

Port scanning is a process to find out any vulnerabilities on open port in a network. The weaknesses of the system will be obtained from the port scanning. Basically, the port scanning system is easy to detect, but the attack will use various methods to hide the attack. As an example, many networks don't create connection log files, so an attacker can send an initial packet with something SYN but no ACK and get a response back (other than a SYN if a port is open) and then stop at that port. This is often called a SYN scan or half open scan, even if it doesn't get logs, for example it might end up as a denial services attack on another host or device connected to the network or port for an open connection. The attacker will send another packet on a port that is still not on the network, but nothing happens to the log file, error file or other device. Any combination of flags other than SYN by itself can be used for port scanning purposes. During the network scanning process, the attacker can gather information about the specific IP addresses that can be accessed over the internet, their targets' operating systems, system architecture and the services running on each computer. In addition, the attacker also gathers details about the networks and their individual host systems. The most common objectives that are encountered during the hacking phase are discovering live hosts, IP address and open ports of live hosts running on the network, discovering open ports where the best means to break into a system or network, also discovering operating system and system architecture which referred to as foot printing and attacker will launch the attack based on the operating system's vulnerabilities. The more important is to be identifying the vulnerability and threat that are present in any system, where the attacker can compromise the system or network by exploiting these vulnerabilities and threats.

2.2.4 Raspberry Pi

RPi is a credit-card sized computer devised by Eben Upton and low cost. This Raspberry Pi is a computer that compiles and controls the algorithms of the Internet of Things (IoT) or robotic projects that we envisage. It will be powered by a Linux-based operating system. The OS that will used in this project is Raspbian. In additions, in

this project will use Raspberry Pi 3B which includes 802.11n WiFi, Bluetooth 4.0 and a quad-core 64-bit ARM Cortex A53 running at 1.2 GHz. Based on the specifications, it is practically used for wireless connection because it has a WiFi module.

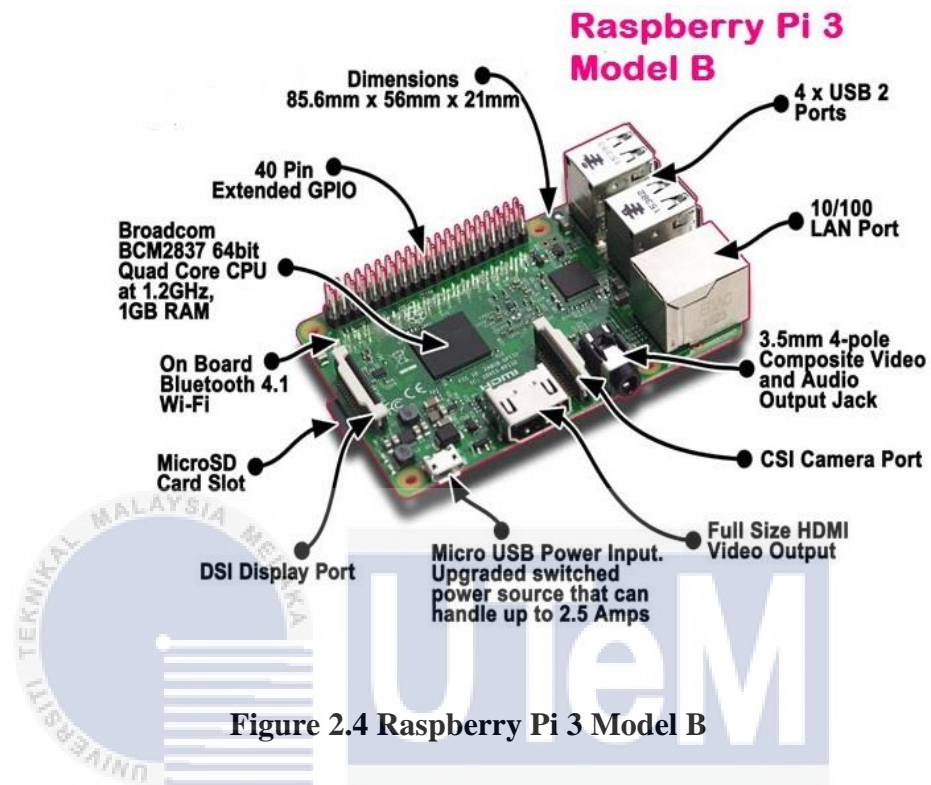


Figure 2.4 Raspberry Pi 3 Model B

2.3 Comparison of the Raspberry Pi

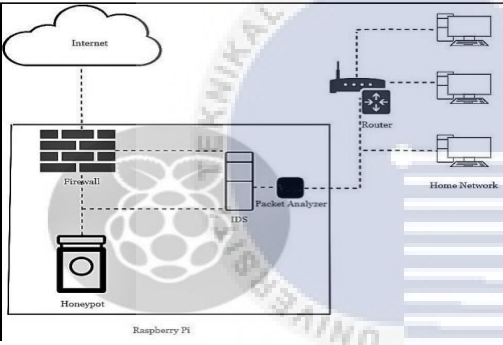
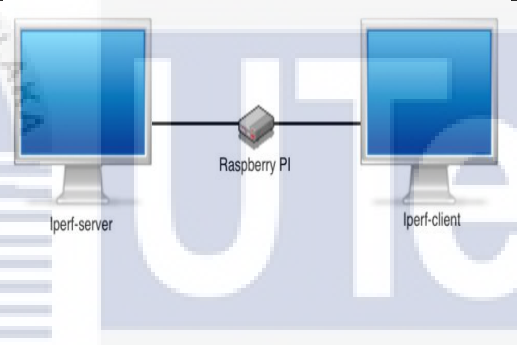
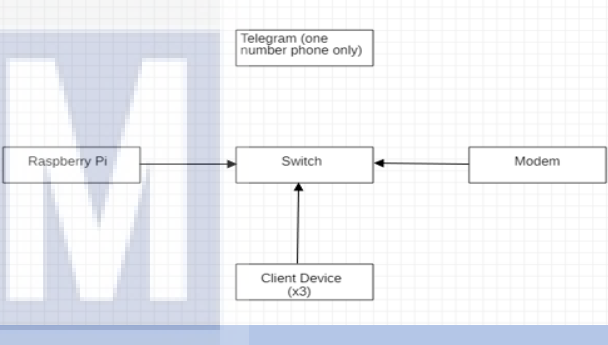
Raspberry Pi consists of a several versions which the latest is RPi 4. Table 2.1 below shows the comparison between version of RPi according to the specification for each RPi. Its shows what are specification of RPi has been upgrading from the first RPi to the third generation.

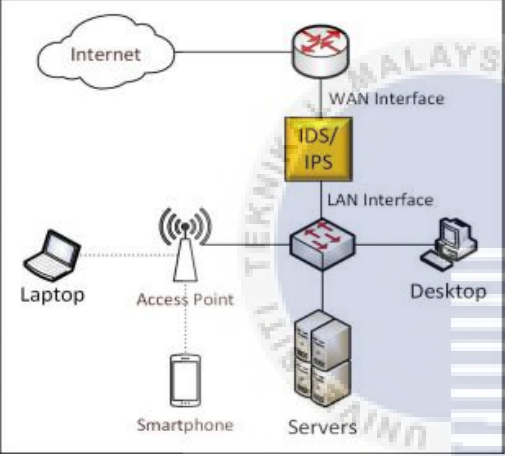
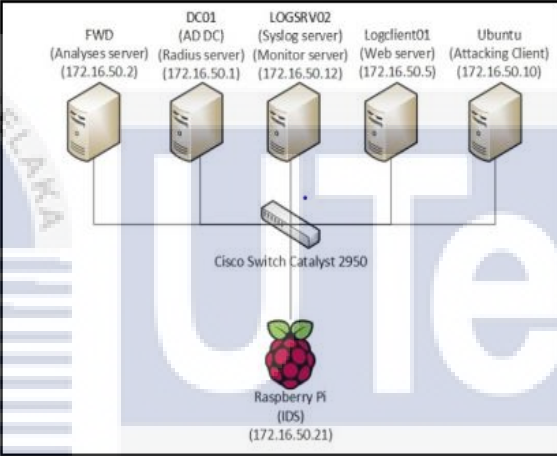
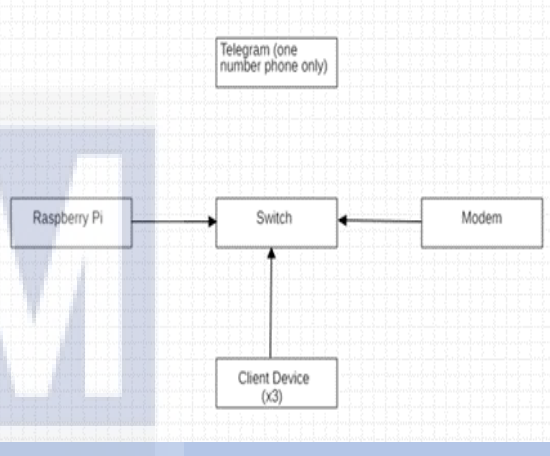
Table 2.1 Comparison between Raspberry Pi

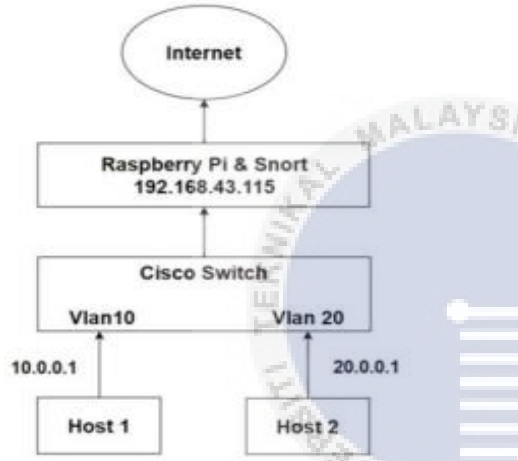
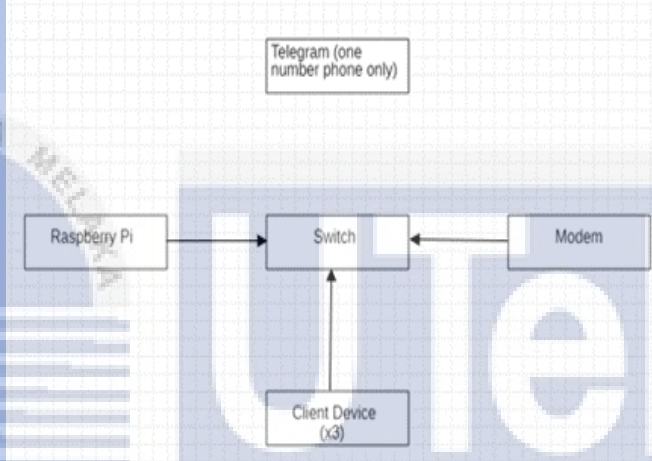
Raspberry Pi model B+	Raspberry Pi 2 model B	Raspberry Pi 3 Model B
<ul style="list-style-type: none"> - July 2014 - A single core ARM1176 processor running at 700MHz - 512 megabytes of memory - Four USB ports - One 10/100 Megabit/s Ethernet port - One micro-sd card for storage - Low power consumption. 	<ul style="list-style-type: none"> - February 2015 - A quad-core ARM Cortex-A7 CPU processor running at 900MHz - 1024 Megabytes of memory - Four USB-ports - One 10/100 Megabit/s Ethernet-port - One micro-SD card slot for storage 	<ul style="list-style-type: none"> - February 2016 - A Quad Core 1.2GHz Broadcom BCM2837 64bit CPU - 1GB RAM - BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board. - 100 Base Ethernet - 40-pin extended GPIO - 4 USB 2 ports - 4 Pole stereo output and composite video port - Full side HDMI - CSI camera port for connecting a Raspberry Pi camera - DSI display port for connecting a Raspberry Pi touchscreen display - Micro SD port for loading operating system and storing data - Upgrade switched Micro USB power source up to 2.5A

2.4 Comparison with the existing system

Table 2.2 Comparison Previous Project

Existing system	1. Raspberry Pi as an Intrusion Detection System, a Honeypot, and a Packet Analyzer	2. IDS on Raspberry PI	Intrusion Detection System Alerting System Using Raspberry Pi
Target User	Intermediate home user and small companies	Intermediate home user and small companies	Home User
Architecture			
Model	Raspberry Pi 3 Model B	Raspberry Pi 2 Model B	Raspberry Pi 3 Model B
Software and hardware	Raspbian-Stretch-2018-Desktop-Image – OS Snort Barnyard2 – Output module for snort and processes the alerts generated by snort into a database format Pulepork – Rule management tool based on PERL	Arch Linux Arm – OS Iperf – Measure network performance Sar – gather information about the system Snort – real-time traffic analysis	Snort Raspbian Buster – OS Barnyard2 Telegram Tenda Switch (TEF1210P-8-150W) Nmap – port scanning Maria Database Apache PhpMyAdmin

Existing system	3. Intrusion Detection and Prevention System for Production Supervision in Small Businesses Based on Raspberry Pi and Snort	4. Pi-IDS: Evaluation of Open-Source Intrusion Detection Systems on Raspberry Pi 2	Intrusion Detection System Alerting System Using Raspberry Pi
Target User	Small business	SOHO (small office/home), educational environment	Home User
Architecture			
Model	Raspberry Pi 3 B	Raspberry Pi 2 Model B	Raspberry Pi 3 Model B
Software and hardware	Ubuntu – OS Snort Kali linux	Snort IDS Bro IDS Raspbian – OS	Snort Raspbian Buster – OS Barnyard2 Telegram Tenda Switch (TEF1210P-8-150W) Nmap – port scanning Maria Database Apache PhpMyAdmin

Existing system	5. Raspberry Pi Firewall and Intrusion Detection System	Intrusion Detection System Alerting System Using Raspberry Pi
Target User	Mid-range business	Home User
Architecture		
Model	Raspberry Pi 3	Raspberry Pi 3 Model B
Software and hardware	Cisco Catalyst 3560-CG switch Snort	Snort Raspbian Buster – OS Barnyard2 Telegram Tenda Switch (TEF1210P-8-150W) Nmap – port scanning Maria Database Apache PhpMyAdmin

2.5 Project Solution

Based on the Table 2.2, it shows the comparison between the previous project and this current project. Roughly, the previous study has been using Cisco switch which is it will spend a lot of money to one hardware because Cisco switch is costly compared to Tenda switch its more affordable and cost-effective. In previous study, the researcher focusing on investigate how the IDS are functioning in the RPi in the medium size of network. The researcher is not focusing to home user who is not able to understand the log, but they are focusing for the intermediate level and small organization that with specialist for the network. Meanwhile, this project is focusing to sending an alert after the detection some abnormal activities that triggering the snort rules.

2.6 Conclusion

Taking everything into account, the literature study is an important component since it comprehends the present highlights of the framework and provides a clear image of how to execute the framework. The assessment and study will facilitate the mobility and comprehension of this work. In general, this writing survey gives subtleties of the entire task to ensure that the investigations had been done dependent on the theme and subtopic as referenced. Moreover, this section additionally to guarantee that the task to be created can give the commitment just as guarantee that the targets of the undertaking have been expressed effectively accomplished. Also, some past examinations have been utilized as references to this venture. It is to fortify the reasons why this venture ought to be actualized. All related or past research, references, contextual analysis, and different discoveries that identify with this undertaking title will be utilized with the end goal of effectively concentrating the task in time with no error.

In a nutshell, this chapter are discussing about the main concept of IDS and classes of IDS. It also discussing about the software and hardware that going to use in this project. To summarize, this literature research aids the project idea by providing an understanding of the system's current characteristics and providing a clear image of how to construct the system. The examination of literature facilitates the smooth growth of this project and provides greater insight.

CHAPTER 3: PROJECT METHODOLOGY

3.1 Introduction

This chapter center around the procedure utilized in finishing this undertaking. The chose strategy is quickly clarified in this subsection. Furthermore, there are two fundamental parts explained in this section which are the project methodology and project milestone. The project methodology will clarify the strategy and the technique stream utilized in a manner to finish this whole project. Then again, project milestone expounded on the calendar of the whole task which every one of the exercises that running all through finishing this project will be in the milestone. Moreover, to demonstrate the task of arranging inside the allocated time Gantt Chart is created.

اونيور سیتی تکنیکل ملیسیا ملاک

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

3.2 Project Methodology

This research will be done by implementing IDS snort on RPi and determine how the snort in the RPi can send the alert to the mobile phone or using Telegram environment. This research is focused on how to maximize ways to securing a small network by alerting user if there any threat that has been detected. IDS is one of the best ways to securing network by detect the suspicious traffic.

Flowchart is a formalized graphic representation of a logic sequence of the system. Based on figure 3.2, show that how this project will be developed. First, before conducting these system developments, research about the topic or main point which is IDS snort on RPi has been conducting to collect and understand all the previous studies. Then, the critical review on previous study has been done to get the clear information and data. Based on summary, it will be developing the comparison table to make sure this project has been improved from previous study. Thus, this project at least can be referred to past study. Present is to bridge the gap between prior research and this research, as well as to indicate how they performed their research and experimentations, the tools that has been used, the technique they used, and the outcomes of their thoughts and experiments. Also knowing about the limitation of their previous project and how to overcome it.

Next, after collecting all the information and data based on the previous study, it has going continue by designing the architecture of the project. In order to design this project, firstly, hardware and software has been identifying and list to use on this project. RPi is the main hardware that used in this project, so RPi 3 model B has been chosen to use in this research since this RPi price's is affordable. Hence, the operating system that going to use is Raspbian buster which has the latest update on 3rd April 2021. Moreover, memory 32GB has been used to store all the traffic. However, software that has been identified to be used are Snort, Barnyard2, MySQL, PhpMyAdmin, Apache (LAMP) and Telegram.

The more important is to install the operating system to the SD card to run the RPi. This process is using Raspberry Pi imager to write the OS into the SD card. After finishing the installing OS, check the OS either in a good environment or corrupted. Then install the snort software. Then check the snort are successfully installed or there

any failed to move to the next step. If snort is fully functioning that can receive the alert, it should be great start.

Furthermore, setting up the switch. By using Tenda switch assign which port that are functioning as a in or out port. This is important because to work out the port mirroring function. On RPi installing the snort and define the specific rule that need to use. Specific and detailing rules are important to make sure the traffic is not missing from give the alert. Lastly, make the connection from snort and switch so that snort can detect the traffic from switch. Switch is one of the best ways to filter the network from malicious activity.

Install and configure the Barnyard2, MySQL, Apache and phpMyAdmin. Log from the snort will be send and store in the MySQL database. This is the critical part to make sure all the snort logs have been stored without missing and can be differentiated based on type of alert. Then proceed to sketch and design the web page which function to user used to see the latest update about the current threat that incoming to their network. This webpage will show the pie chart and graph about the statistic of the threat. This webpage is optionally if user want to see the flow and analyze the packet that income and outgoing. Rather than they see from the Telegram notification. This webpage will be presented based on what are log from the snort that has been received and store in the database.

The final part and my contribution in this project are to send the alert to the telegram that can notify user if any incoming alert. In this phase, Telegram will generate unique identifier to send the alert to specific user. User will receive the notification synchronize with the log that has been received by snort. In this notification will justify the time and IP address of source and destination and type of alert depends on protocol that has been created in snort rules.

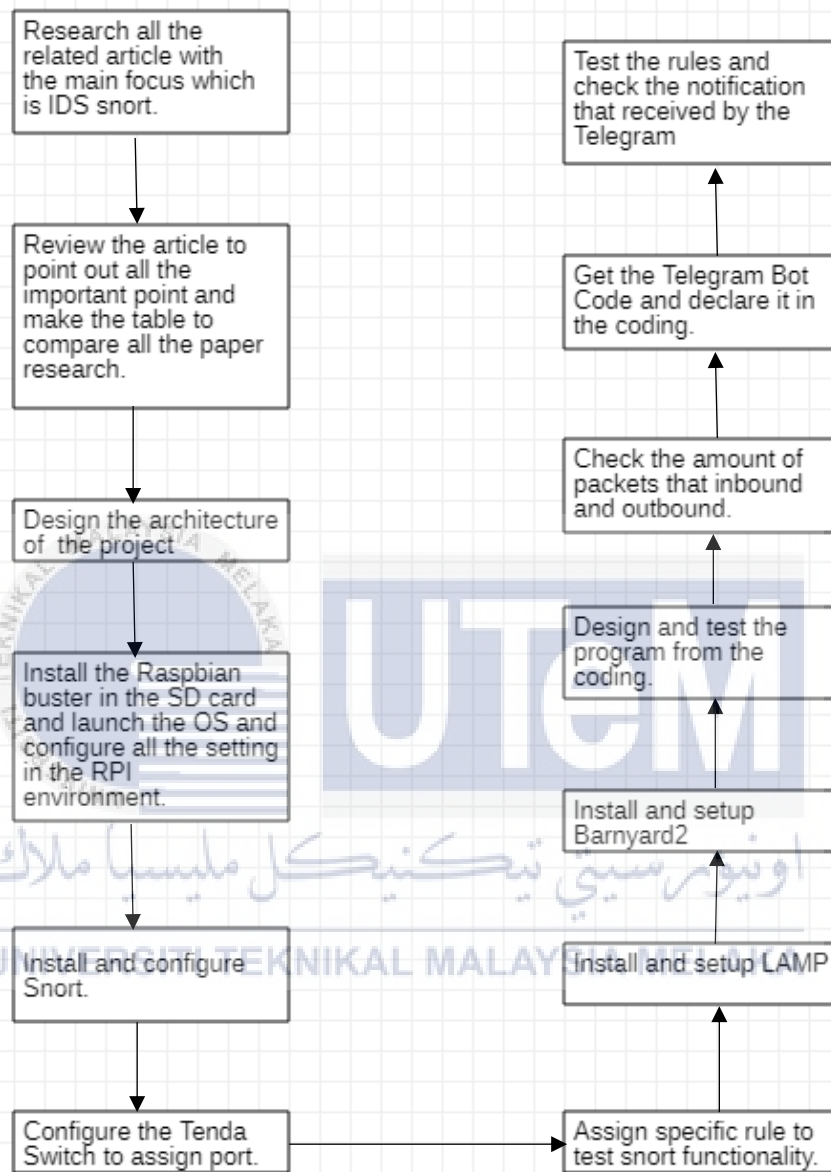


Figure 3.1 Flowchart of the project

3.3 Project Milestones

This project was created from the ground up, beginning with a concept that was produced and offered. The project specifics were thoroughly gathered based on earlier study, and material selection was made. The design was created using the materials chosen to test the project's resilience. The project will be developed further in PSM 2 according to the design. The Gantt chart for both PSM 1 and PSM 2 is shown in Table 3.1.



3.4 Conclusion

Taking everything into account, the project methodology is now done in this section. This chapter was discussed on the process and procedures on how the project would be develop. In the meantime, so as to complete the work on schedule and staying away from any deferred, project milestone is created. In conclusion, this chapter is highlighted to ensure the project in a perfectly timing manage and organized.



CHAPTER 4: DESIGN

4.1 Introduction

This segment mission is to find the theoretical design assessment and the consequence of thorough study. Designing a prototype is a vital step to make sure the product is smoothly running based on its planning. Thus, in this chapter, the detail prerequisite and design on the project will be finally depicted. All the software, hardware and specific design will be discussed in detail in this chapter.



4.2 Problem Analysis

Port scanning one of the famous ways in attacking missions. This if the first step that attacker will take to know either the host is alive or not and which port that attacker can attack. So that it can be use RPi to run the IDS snort to detect any port scanning that attacker has been made. This is because, RPi is cost-effective for every level of user. After snort detecting this threat, it will save in log file. Then, this log will be sent to Barnyard2 to process the log, so that it can be save in the database (MariaDB). It will be count number of packets that has been received to avoid any packet loss where it can lead to packet drop attack. All this log that has been store in database will be convert to the graphical user interface (GUI), so that it can easily understand by the user. Other than that, the log file will send to the Telegram, and it will notify the user that the snort rules have been trigger. Figure 4.1 showed the fundamental general flow of the system architecture of this project.

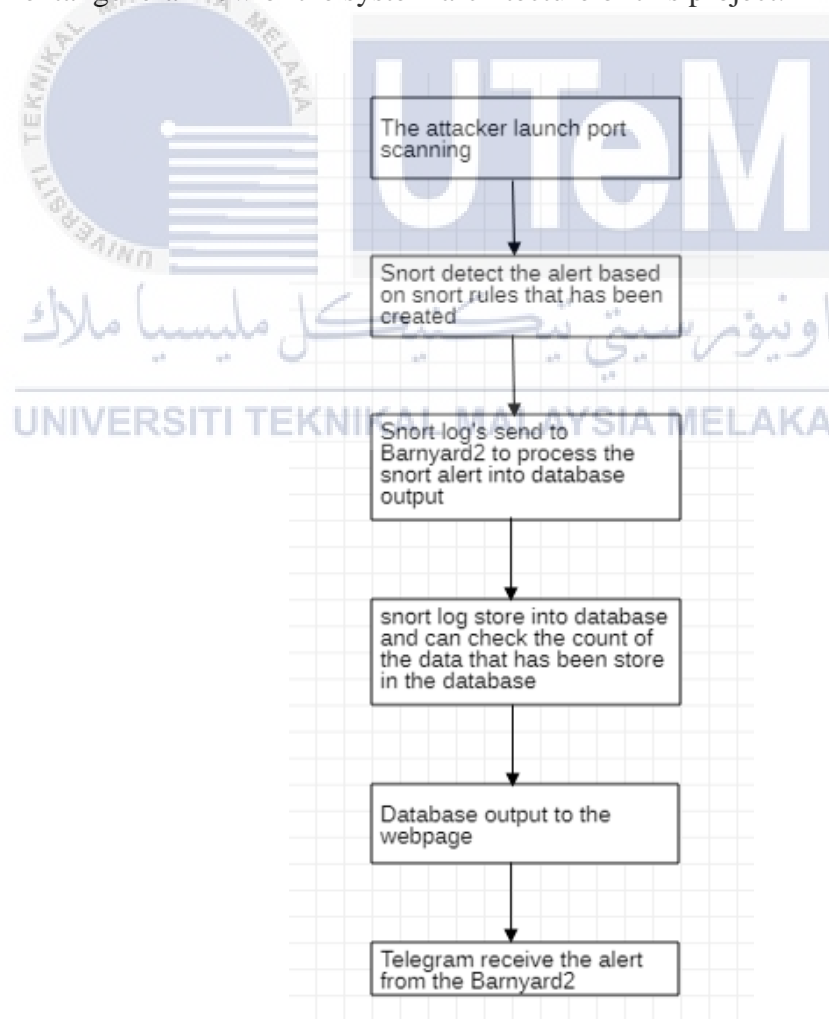


Figure 4.1 Flow System Architecture

4.3 Analysis Requirement

4.3.1 Data Requirement

Figure 4.2 shown the data flow of the project. The main data are from the snort log file that will be transfer to the Telegram as a message and to the database to store as backup and to display in the webpage.

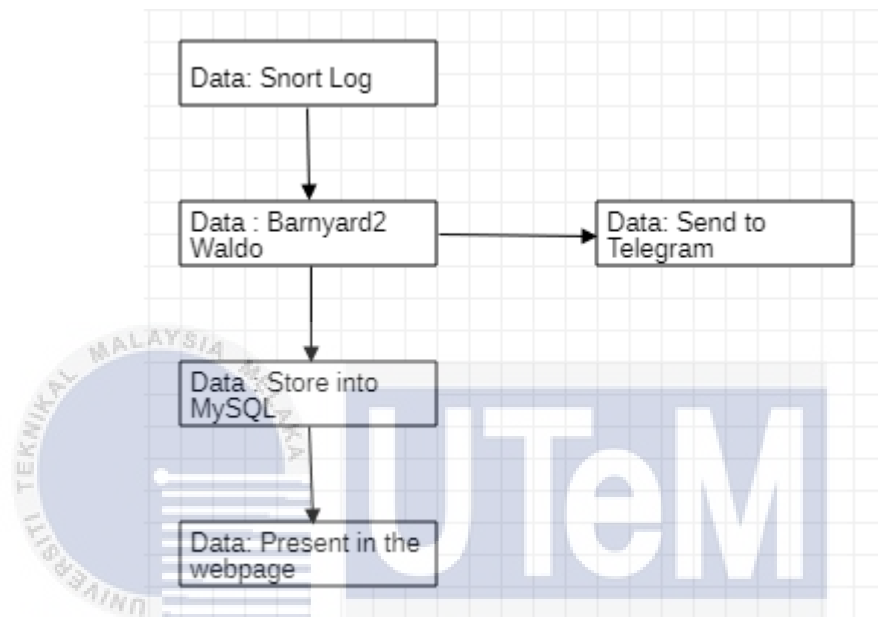


Figure 4.2 Data Flow of the Project

4.4 Software Requirement

4.4.1 Snort

Snort is one of powerful software that supported to be used in monitor network traffic using various operating systems. Snort can run for network IDS (NIDS). Snort is an open-source application that provide real-time network traffic analysis and data packet logging. It uses a sequence of rules that support to identify unusual network activity and uses those rules to find packets that match up against the rules and produces alert to the user. Snort will be deployed inline to stop that unusual packet. There are three primary uses of snort. Firstly, functions as a packet sniffer such as tcpdump. Second, function as a packet logger where useful for network traffic debugging. Lastly, function as a full-blown network intrusion detection system (Parag Vadher, 2020).



Figure 4.3 Snort Logo

4.4.2 Telegram

Telegram is a software that can use using mobile and desktop. This software is functioned to send message. This software is focusing on speed and security where famous as super-fast, simple, and free. There are a lot of functionality of this software but the more important that need to be highlighted is this software has bot functionality where this bot are made by third-party developers using Telegram Bot API. This bot used to receive the snort logs and send the notification to the telegram. In this project, Bot Father are used to create the bot and naming the bot as user likes and bot raw to get the ID for each phone number.



Figure 4.4 Bot Father Logo

4.4.3 Barnyard2

Barnyard2 is functioned to load the log file to the MySQL server that will be read by the Web-GUI server. This is an open-source interpreter for snort unified2 binary output files. Its major function is to allow Snort to write to disks efficiently while delegating the burden of parsing binary data into multiple forms to a separate process that will not cause Snort to miss network traffic.

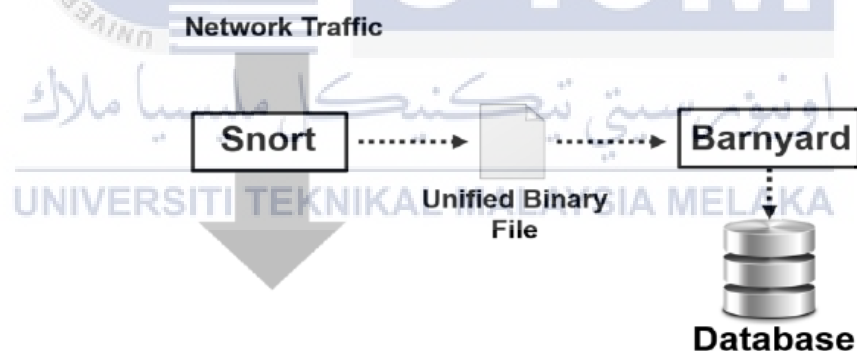


Figure 4.5 Illustration function of Barnyard2

4.5 Hardware Requirement

4.5.1 Raspberry Pi

RPi is the phrase for a range of single-panel computers. The RPi Model 3B has been introduced in February 2016 and has since published many versions and iterations. The earliest Pi had a 700MHz single-core CPU and just 256 MB RAM and the new design has a 1.2GHz quad-core Processor with 1 GB RAM. Moreover, people across the world are using Raspberry Pi to know and understand the software development skills, develop hardware projects, do smart homes, and even use it in commercial processes. Besides that, the RPi is a very economical computer running Linux, but it also offers a set of GPIO (general input / output) pins that allow user to manage physical modules and investigate the Internet of Things. Therefore, the RPi Foundation is working to put the authority of computing and digital making into people's hands around the world. It manages so by delivering high-performance, low-cost computers what users use to learn, resolve issues, and have leisure. Furthermore, RPi acts in the open-source environment. It has runs Linux which a diversity of distributions, and its core processor supported Raspbian operating system, is open source and runs an open-source software complement. The RPi Project is contributing to the Linux kernel and many other open-source tools as well as developing much of its own open-source software.



Figure 4.6 Raspberry Pi Model 3B

4.5.2 Switch

Switch function to connect other devices together such as computers, wireless access points and server where in the same local area network. Advantage of switch to increase level of security. In this project, Tenda TF1210P-8-150w switch has been used where has function to manage virtual LAN (VLAN), port security and port mirroring.



Figure 4.7 Tenda Switch

4.6 Model Design

Figure 4.8 shows the design for this project however Figure 4.9 will show actual hardware and software that are functioning in this project. In this project, the main important is the RPi where it will function to catch all the snort log and process it to alert to the user. Snort will function as NIDS to monitor the outbound and inbound traffic. The specific rules will function to make sure any suspicious activity and attempt to the network can be notify as an alert. In this project, rules port scanning for TCP will be applied. Log will be sent to Barnyard2, and it will store into the database. Website will show all the activity that from the snort log. So, the user will not read the log from the command anymore. Then, switch will function to connect all the device and do some filtering to who has enter the network. It also functions as port mirroring where it will forward the packet to the server which is the RPi. Modem will send the electrical signal to the other device; however, it will modulate analog signal carrier where it will carry the digital information, and it also demodulates an analog signal where it will decode the digital information from the analog carrier signal (Clare Edwards, 2018). Moreover, the computer act as a client where this client will try to access the server or as an attacker that want to port scanning the network. From that, the alert will be generated and send to mobile phone. Mobile phone will receive the notification in the Telegram if there any suspicious activity that triggered the rules.

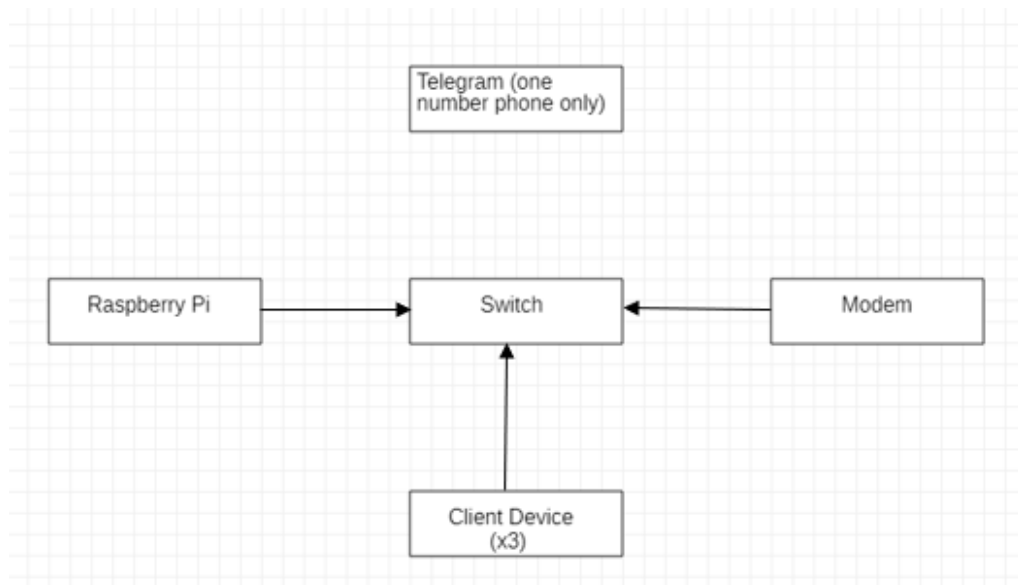


Figure 4.8 Design Architecture of the Project



Figure 4.9 Real Design

4.7 Conclusion

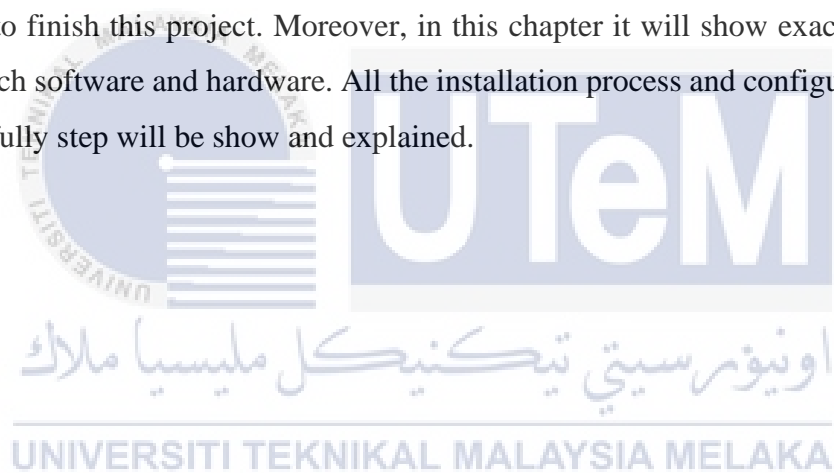
In a nutshell, the entire problem analysis, flow system architecture, and specification on hardware and software has been collected on this chapter. Analysis and design are one of the key elements before the implementation phase. This phase is the pre-preparation phase for the implementation and flow system. Thus, this chapter give better understanding before the process implementing it.



CHAPTER 5: IMPLEMENTATION

5.1 Introduction

In this chapter it will show how the process of this project will be going. From the installing snort until get the notification in the Telegram. It also will be focused on how for each of the software and hardware that has been identified in chapter 4 will be used to finish this project. Moreover, in this chapter it will show exact functionality for each software and hardware. All the installation process and configuration process with fully step will be show and explained.



5.2 Software Development Environment Setup

In this section, the development environment setup for IDS alerting system using RPi will involve hardware and software requirements. It will explain about the status of development for each component or module contains in this project. All the setups will be stated step by step and clearly shown. The hardware and software requirements are stated in the chapter 4 and will be explain further for the connection in the below section. Figure 5.1 shows the roughly execution and process for each of the software that will be explain further and which software that assign to which hardware.

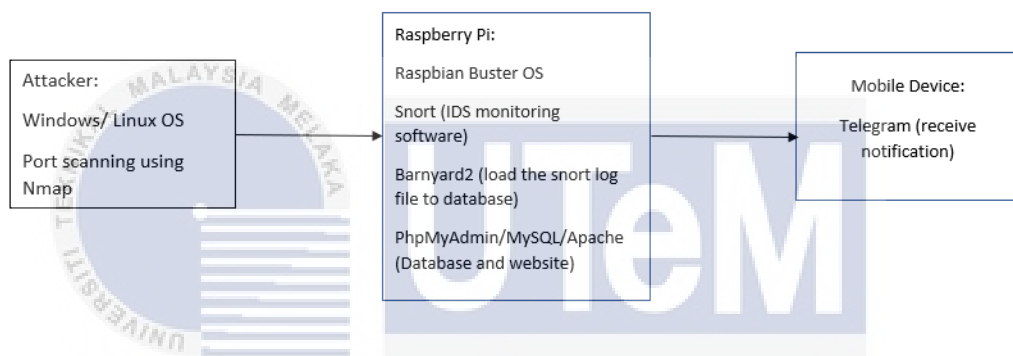


Figure 5.1 Software Flow

5.2.1 Package Dependent Libraries

Figure 5.2 until Figure 5.11 shows the installation for all the libraries dependency because RPi are not fully download the libraries. During the database installation, some packages are not included, so that it need to install manually.

```

root@raspberrypi:/home/pi# apt-get install default-mysql-server -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
  
```

Figure 5.2 Dependency for MySQL Server

```

root@raspberrypi:/home/pi# apt-get install default-libmysqlclient-dev
Reading package lists... Done
Building dependency tree
  
```

Figure 5.3 Dependency for library MySQL Client

```
root@raspberrypi:/home/pi# apt-get install autoconf
Reading package lists... Done
```

Figure 5.4 Install Auto configure

```
root@raspberrypi:/home/pi# apt-get install default-mysql-client
Reading package lists... Done
Building dependency tree
```

Figure 5.5 Install MySQL client

```
root@raspberrypi:/home/pi# apt-get install libtool -y
Reading package lists... Done
Building dependency tree
```

Figure 5.6 Install libraries

```
root@raspberrypi:/home/pi# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
```

Figure 5.7 Securing the Database

```
root@raspberrypi:/home/pi# apt-get install -y libapache2-mod-php
Reading package lists... Done
Building dependency tree
```

Figure 5.8 Install dependency for database

```
root@raspberrypi:/home/pi# apt-get install -y php php-common php-gd php-cli php-xml php-mysql
Reading package lists... Done
```

Figure 5.9 Install dependency for database

```
root@raspberrypi:/home/pi# apt-get install -y php-pear libphp-adodb
Reading package lists... Done
Building dependency tree
```

Figure 5.10 Install dependency for database

```
root@raspberrypi:/home/pi# apt-get install software-properties-common -y
Reading package lists... Done
Building dependency tree
```

Figure 5.11 Install dependency for database

5.2.2 Data Acquisition (DAQ)

Firstly, download the file DAQ version 2.0.6 from the internet. It is because RPi are not fully support DAQ latest version. Figure 5.12 shows the command for DAQ installation. In Figure 5.13 and 5.14 shows the package dependencies which missing that need to install the package manually.

```
root@raspberrypi:/home/pi/Downloads/daq-2.0.6# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
```

Figure 5.12 DAQ installation

```
root@raspberrypi:/home/pi/Downloads/daq-2.0.6# apt-get install flex bison -y
Reading package lists... Done
```

Figure 5.13 Missing Dependencies

```
ot@raspberrypi:/home/pi/Downloads/daq-2.0.6# sudo apt-get install libpcap-dev
ading package lists... Done
```

Figure 5.14 Missing Dependencies

5.2.3 Snort

Install some missing dependencies in snort file to make sure snort can support to function well otherwise it can be error. Install snort however need to download snort file first. You can search it in the google. Do the configuration to add user snort and directory for rules. In the snort configuration file, set the network address and path for the rules that has been created. Comment for all the rules that has been downloaded from the snort to avoid some error and uncomment for only one file that be file for rules. Validate the snort to make sure snort are successfully configured. Figure below shows command for installing snort dependencies

```
root@raspberrypi:/home/pi/Downloads/snort-2.9.9.0# apt-get install -y libdumbnet-dev liblua5.2-dev
libnghttp2-dev
```

Figure 5.15 Snort Missing Dependencies

```
root@raspberrypi:/home/pi/Downloads/snort-2.9.9.0# sudo apt-get install libpcre3-dev
Reading package lists... Done
```

Figure 5.16 Snort Missing Dependencies

```
root@raspberrypi:/home/pi/Downloads/snort-2.9.9.0# apt-get install -y libdumbnet-dev liblua5.2-dev
libnghttp2-dev
```

Figure 5.17 Snort Missing Dependencies

```
root@raspberrypi:/home/pi/Downloads/LuaJIT-2.0.5# make && make install
==== Building LuaJIT 2.0.5 ====
```

Figure 5.18 Dependencies LuaJit

```
root@raspberrypi:/home/pi/Downloads/snort-2.9.9.0# ./configure --enable-sourcefire
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
```

Figure 5.19 Snort configure by enabling sourcefire

```

root@raspberrypi:/home/pi/Downloads/snort-2.9.9.0# sudo make & make install
[2] 31059
Making install in src
make[1]: Entering directory '/home/pi/Downloads/snort-2.9.9.0/src'

```

Figure 5.20 Snort installation

5.2.4 Rules

Enter the rules directory that has been created and create rules. Create the sid message map for setup the alert output. Figure 5.21 shows the sid message map template.

```

GNU nano 3.2 /etc/snort/sid-msg.map Modified
1 || 10000001 || 001 || icmp-event || 0 || ICMP Test detected || url,tools.ietf.org/html/rfc792

```

Figure 5.21 sid message map

5.2.5 Barnyard2

Download and install the barnyard version 2-1.13 and configure it to link with the MySQL and snort. Figure 5.22 show the installation of Barnyard2 with the MySQL and Figure 5.23 shows installation with snort.

```

root@raspberrypi:/home/pi/Downloads/news/barnyard2-2-1.13# ./configure --with-mysql --with-mysql-libr
aries=/usr/lib/arm-linux-gnueabi/hf/

```

Figure 5.22 Barnyard with MySQL

```

root@raspberrypi:/home/pi/Downloads/news/barnyard2-2-1.13# cp etc/barnyard2.conf /etc/snort/
root@raspberrypi:/home/pi/Downloads/news/barnyard2-2-1.13# sudo mkdir /var/log/barnyard2
root@raspberrypi:/home/pi/Downloads/news/barnyard2-2-1.13# sudo chown snort.snort /var/log/barnyard2
root@raspberrypi:/home/pi/Downloads/news/barnyard2-2-1.13# sudo touch /var/log/snort/barnyard2.waldo
root@raspberrypi:/home/pi/Downloads/news/barnyard2-2-1.13# sudo chown snort.snort /var/log/snort/barn
yard2.waldo

```

Figure 5.23 Barnyard with snort

5.2.6 Database

Enter the database in the command prompt, create database for snort and assign the password and username for access. To check amount of the packet that has been received, you can check by count by event. Figure 5.24 shows the configuration that

need to do in the database configuration and Figure 5.25 show the command to check the number of logs that has been receive.

```

GNU nano 3.2 /etc/snort
#
# Examples:
# output database: log, mysql, user=root password=test dbname=db host=localhost
# output database: alert, postgresql, user=snort dbname=snort
# output database: log, odbc, user=snort dbname=snort
# output database: log, mssql, dbname=snort user=snort password=test
# output database: log, oracle, dbname=snort user=snort password=test
#
# alert_fwsm: allow blocking of IP's through remote services
# -----
# output alert_fwsm: <SnortSam Station>:<port>/<key>
#
# <FW Mgmt Station>: IP address or host name of the host running SnortSam.
# <port>:          Port the remote SnortSam service listens on (default 898).
# <key>:          Key used for authentication (encryption really)
#                of the communication to the remote service.
#
# Examples:
#
# output alert_fwsm: snortsambox/idspassword
# output alert_fwsm: fw1.domain.tld:898/mykey
# output alert_fwsm: 192.168.0.1/borderfw 192.168.1.254/wanfw
#
output database: log, mysql, user=snort password=123456 dbname=snort host=localhost sensor name=sensor01

```

Figure 5.24 Database configuration

```

root@raspberrypi:/etc/snort# mysql -u snort -p -D snort -e "select count(*) from event"
Enter password:
+-----+
| count(*) |
+-----+
| 0        |
+-----+

```

Figure 5.25 Log count

5.2.7 Telegram

Create the bot by using Father Bot and get the ID for bot. enter the token and ID. Edit the design how the alert would be. Figure 5.26 shows the configuration file of the Telegram that design how the message could be sent.

```

GNU nano 3.2 bot-tele.sh

#!/bin/bash

#init
initCount=0
logs=/home/pi/Desktop/alert-converting.txt

#File
msg_caption=/tmp/telegram_msg_caption.txt

#Chat ID dan bot token Telegram
chat_id="283680038"
token="1945640027:AAGshxka0yyWXqIERzkBNGlgxSTPJ09qc2o"

#kirim
function sendAlert
{
    curl -s -F chat_id=$chat_id -F text="$scaption" https://api.telegram.org/bot$token/sendMessage #> /dev/null 2&>1
}

#Monitoring Server
while true
do
    lastCount=$(wc -c $logs | awk '{print $1}') #getSizeFileLogs
    #DEBUG ONLY
    #echo before last $lastCount #ex 100 #after reset 0
    #echo before init $initCount #ex 0
    #echo "-----"
    if ((${lastCount}) > $initCount);
    then
        #DEBUG
        #echo "Sending Alert..."
        msg=$(tail -n 2 $logs) #GetLastLineLog
        echo -e "Alert Notification!!\n\nServer Time : $(date +%d %b %Y %T)\n\n" $msg > $msg_caption #set Caption
        caption=$(cat $msg_caption) #set Caption
        sendAlert #Panggil fungsi di function
        echo "Alert Sent"
        initCount=$lastCount
        rm -f $msg_caption
        sleep 1
    fi
    sleep 2 #delay if Not Indication
done

```

Figure 5.26 Telegram Identification

5.2.8 Webpage

Figure 5.27 shows the script for the webpage for display function and figure 5.28 shows the database scrip to connect the webpage with the database where the log file has been stored.

```

function display_ct() {
    var x = new Date();
    var ampm = x.getHours() >= 12 ? ' PM' : ' AM';
    hours = x.getHours() % 12;
    hours = hours ? hours : 12;
    var day = x.getDate();
    var months = x.getMonth() + 1;
    var hour = hours + ":" + x.getMinutes() + ":" + x.getSeconds() + ampm;
    var dates = day + "/" + months + "/" + x.getFullYear();
    var cur = hour + "<br>" + dates;
    document.getElementById('ct').innerHTML = cur;
    display_c();
}
</script>
<script>
$(document).ready(function() {
    setInterval(function() {
        $("#threat").load("docs/threat.php");
        $("#pingthreat").load("docs/pingsweep.php");
        $("#nmapthreat").load("docs/nmap.php");
        $("#table").load("docs/table.php");
        // refresh();
    }, 100)
});

```

Figure 5.27 Webpage Script

```

<?php
$dbServerName = "localhost";
$dbUserName = "root";
$dbPassword = "";
$dbName = "snort";

$conn = mysqli_connect($dbServerName, $dbUserName, $dbPassword, $dbName);

//check the error
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

```

Figure 5.28 Database Script

5.3 Implementation Status

Table 5.1 Implementation status

No.	Status	Descriptions	Duration for completed
1.	Assemble the hardware	Process of gathering all the required hardware that will be used for building the prototype device.	14 days
2.	Configuration of software environment	Process configuration of Raspbian Buster, snort, and database.	30 days
3.	Implementation of source code	Process of writing a source code to make the prototype website function as the project proposed.	60 days
4.	Monitor the network	Process to monitor traffic to make an alert.	14 days
5.	Develop website for system	Create webpage where user can understand the alert using GUI interface.	60 days
6.	Telegram Configuration	Configure the Telegram application to make sure that the system can sent an alert notification to the Telegram apps by installing Telegram Bot. The bot will send the message to desired channel/chat. In this project, I create a private channel which is IDS Alert for receiving the alert notification.	4 days

5.4 Conclusion

In conclusion, the critical part is installing the software and setup the environment for each software that need to be linked together this known as development stage. In this chapter most-likely are discussing about the installing and extra package that need to be installed because of the missing file. This chapter also show for each step during developing process to accomplish the purpose of this project.



CHAPTER 6: TESTING

6.1 Introduction

Test plans are divided into several part which are test organization, environment, schedule, strategy, and classes of test where it will be discussed more in this chapter. Test designs are divided into two which are test data and test description. This chapter provides the testing and analysis results from this project which is how the snort will be send the alert to the telegram. The alert will be sent if the rules has been fulfilled. In this chapter shown that the project that has been successfully execute and the objective has been accomplished.

6.2 Test Plan

Test plan is how the strategy to execute this project. Thus, in order to get the result, a testing phase need to be conducted to verify whether the developed system can be use or not. It will be briefly discussed about subject in this phase. The reason of documenting this is to expect the unexpected that came during execution of the proposed method.

6.2.1 Test Organization

In this part, it will briefly be described about the person or subject involved while testing phase is conducted. This project required three subjects in order to see the functional is success or not. The two subjects mentioned is as follow.

6.2.1.1 Attacker

Attacker will be responsible to launch the attack towards network environment. This person will be used the any OS that was focused only for attacking the environment with Wi-Fi connection. In this project the attacker will try to run port scanning to look which port that vulnerable port. To execute port scanning, in this case attacker might be use Nmap.

6.2.1.2 User and Administrator

This role is the important role in this project where it will be functioned to monitor the network. This role also will work to determine whether it is pass or fail result in order to detect, alert, and notifying the threat or attack towards network. All the result and consequences will be recorded by the following roles. Besides, this role also required to provide the feedback towards the proposed system in terms of flexibility of the solution.

6.2.1.3 Network Engineer & System Developer

As mentioned above, all the result and possible outcomes will be recorded and analyzed by these two roles according to their role and expertise. This is the primary role in this phase because all the possible outcomes will be analyzed and troubleshoot in order to produce the best solution for user.

6.2.2 Test Environment

Test environment will be describing and illustrates the testing environment for the proposed solution. The solution will have a Raspberry Pi 3 that act as server. This is the backbone of this solution. Besides, it also required the switch, wireless router and also an attacker machine. The switch will act as span mirroring devices that will filter all the incoming and outgoing packet towards the network according to the configured port. All the logs and packet than will be filtered by RPi as the IDS server. Lastly, an attacker machine will be needed, and it will launch the attack towards the network by using Wireless connection. Figure 6.1 illustrate the topology for the testing environment.

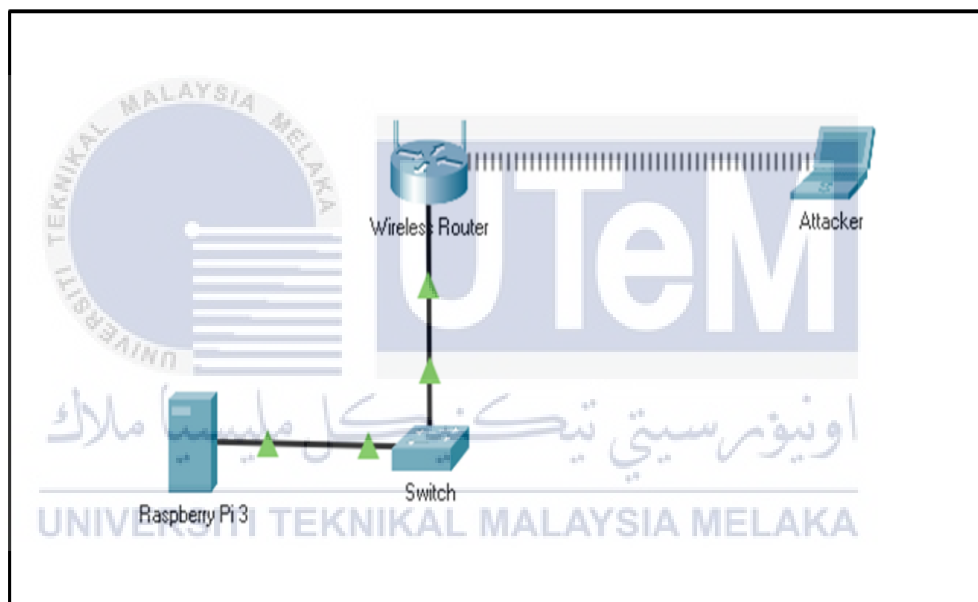


Figure 6.1 Illustrated the test environment

6.2.3 Test Schedule

This section describes how the testing phase will be conducted with standby subject according to their role. All the unexpected outcomes will make the cycle loop back to the previous implementation phase. It required the debug and troubleshot all the configuration in order to patch it.

6.3 Test Strategy

Test strategy will be describing about the approach that used in this phase. The approach techniques used in this phase is the black box test technique. Black box technique is the suitable approach in detecting the system or software failure. This technique does not cover in deep such as the development code script, method, and others.

6.3.1 Classes of Test

6.3.1.1 Functional Test

Main factor of doing functional test is to check whether the RPi can communicate with mirroring devices properly and correctly in order to trigger an alarm whenever there have anomalies activities in the network caused by attacker. The phase started with testing the communication between IDS RPi 3 and the mirroring span devices. This test became mandatory because all the next items will be depending on the connectivity within the span mirroring switch and IDS. This because all the incoming packets will be monitored and filtered by the port that have been set as mirroring port on the switch configuration.

6.4 Test Design

The outcome of this section will produce a form that illustrated in a table that contain Test Case ID, Test Functionality, precondition, steps, results, error and the conclude result whether it is success or fail.

6.4.1 Test Description

Table below will indicate as form that filled during the testing process.

Table 6.1 RPi 3 and Mirroring Switch Connectivity

Test Case ID	ND01
Test Functionality	Raspberry Pi 3 and mirroring switch connectivity
Precondition	Snort that installed in RPi 3 and the port mirroring configuration in switch
Execution Steps	<ol style="list-style-type: none"> i. Connect the RPi 3 and the wireless router to the port that have been set as mirroring port as in previous chapter. ii. Run the bash scripting execution file as coded in previous chapter iii. Ping testing to the raspberry pi 3 with the wireless connection
Expected Result	<ul style="list-style-type: none"> ❖ The barnyard2 panel will shows the alert if it detects any anomaly activities in the network referring to the rule that have been set in previous chapter
Error Message	None
Result	Pass

Table 6.2 System Login Testing

Test Case ID	ND02
Test Functionality	System login testing
Precondition	Apache2, MySQL, PhpMyAdmin, browser
Execution Steps	<ol style="list-style-type: none"> i. Open the browser and go to the raspberry pi IP address ii. The login prompt alert will be shown if the system does not detect any active login session. Besides, there have some tab that hidden when there no active session detected iii. Go to login tab iv. Enter username and password v. Click the login button
Expected Result	<ul style="list-style-type: none"> ❖ The system will redirect to the homepage index with all tabs shown when successful login ❖ The system will give the wrong username and password alert when wrong username or password entered
Error Message	None
Result	Pass

Table 6.3 Add New User Testing

Test Case ID	ND03
Test Functionality	Add new user testing
Precondition	Apache2, MySQL, PhpMyAdmin, browser
Execution Steps	<ol style="list-style-type: none"> i. Open browser and go to raspberry pi address then login to the system ii. Click to the add new user button located in the side navbar iii. Fill in the form required and choose the role for the new user iv. Click register button
Expected Result	<ul style="list-style-type: none"> ❖ The system will prompt the alert that inform the registration status whether it is success or failed. If the registration is success, the Ok button in the alert will redirect to the homepage. Meanwhile, if the error alert prompted, user will be noticed what type of error that they faced.
Error Message	None
Result	Pass

Table 6.4 Logout Testing

Test Case ID	ND04
Test Functionality	Logout testing
Precondition	Apache2, MySQL, PhpMyAdmin, browser
Execution Steps	<ol style="list-style-type: none"> i. Open browser and go to raspberry pi address then login to the system ii. Click the logout button that located in the side navbar, or user can click to the full name of user and there have a dropdown that shows the logout item button
Expected Result	❖ Alert prompt that informs the user successfully have logout to the system. This also will lead the system to kill or destroy the session that required relog in in future
Error Message	None
Result	Pass

Table 6.5 Web Panel

Test Case ID	ND05
Test Functionality	Web Panel that shows all threat according to the class and type of attack and the today threat detected
Precondition	Barnyard2, Apache2, MySQL, PhpMyAdmin, Browser as setup before.
Execution Steps	<ol style="list-style-type: none"> i. Start the apache2 on raspberry pi OS. Normally, this service will auto start when the OS is boot. ii. Run the bash scripting execution file as configured before iii. Open the browser and go to raspberry pi 3 address iv. Ping testing to the raspberry pi 3 with the wireless connection
Expected Result	❖ The panel will show the today threat, total threat, Ping Sweep threat, NMAP Port Scan Threat counted
Error Message	None
Result	Pass

Table 6.6 Telegram Alert

Test Case ID	ND06
Test Functionality	Test the telegram alert notification
Precondition	BotFather and telegram bash scripting code as setup before
Execution Steps	<ol style="list-style-type: none"> i. Run the bash scripting executed file ii. Ping testing to the raspberry pi 3 with the wireless connection
Expected Result	❖ The BotFather bot chat will give the alert notification if the server has detected the abnormal activities in the network
Error Message	None
Result	Pass

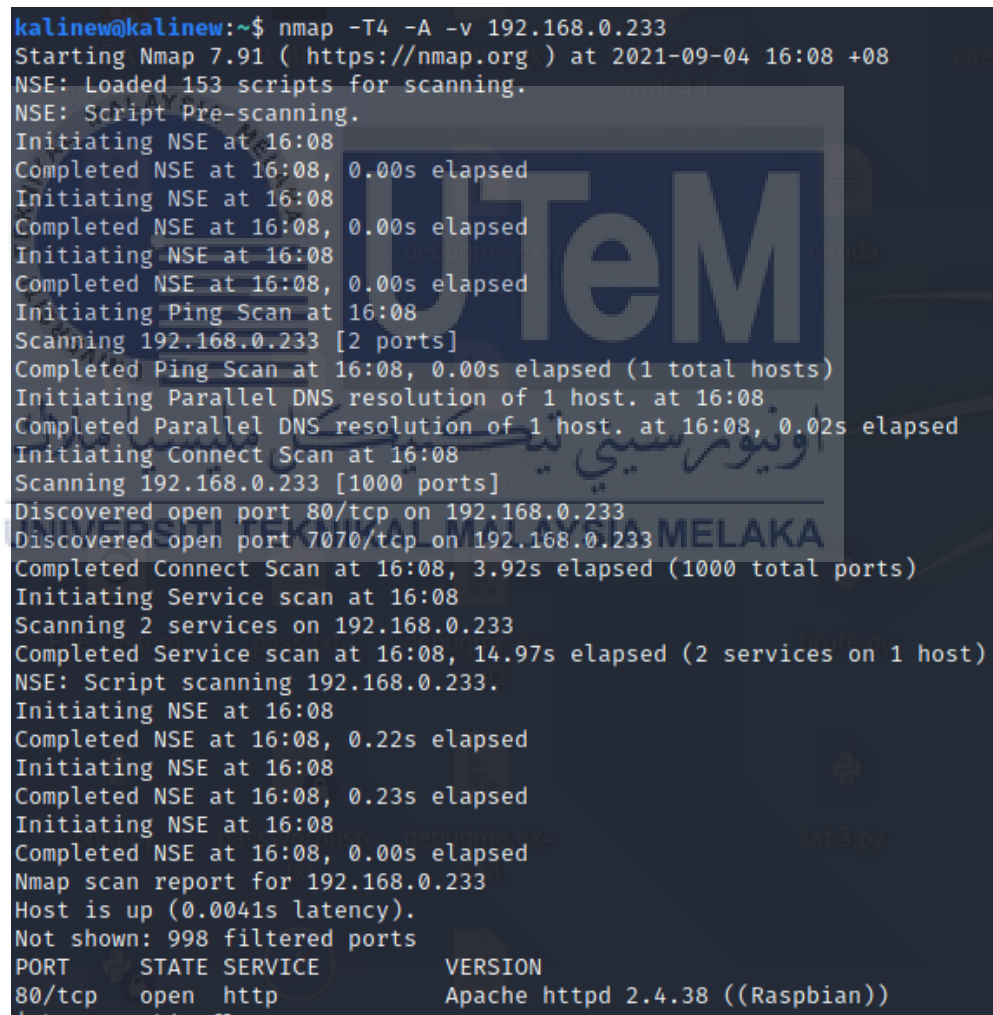
6.5 Test Result and Analysis

This section will be documenting the result and analysis of the carried test.

6.5.1 Raspberry Pi 3 And Mirroring Switch Connectivity

6.5.1.1 Attacker Side

The Attacker first need to connect to the wireless that provide by wireless router. Next, the testing is carried by using NMAP network scanning tool. Attackers need to enter the “nmap -T4 -A 192.168.0.233” in the terminal. The result of this will show the open port for the server. Based on Figure 6.2, it shows that the host has two open port which are port 80 and port 7070.



```

kalinew@kalinew:~$ nmap -T4 -A -v 192.168.0.233
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-04 16:08 +08
NSE: Loaded 153 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 16:08
Completed NSE at 16:08, 0.00s elapsed
Initiating NSE at 16:08
Completed NSE at 16:08, 0.00s elapsed
Initiating NSE at 16:08
Completed NSE at 16:08, 0.00s elapsed
Initiating Ping Scan at 16:08
Scanning 192.168.0.233 [2 ports]
Completed Ping Scan at 16:08, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 16:08
Completed Parallel DNS resolution of 1 host. at 16:08, 0.02s elapsed
Initiating Connect Scan at 16:08
Scanning 192.168.0.233 [1000 ports]
Discovered open port 80/tcp on 192.168.0.233
Discovered open port 7070/tcp on 192.168.0.233
Completed Connect Scan at 16:08, 3.92s elapsed (1000 total ports)
Initiating Service scan at 16:08
Scanning 2 services on 192.168.0.233
Completed Service scan at 16:08, 14.97s elapsed (2 services on 1 host)
NSE: Script scanning 192.168.0.233.
Initiating NSE at 16:08
Completed NSE at 16:08, 0.22s elapsed
Initiating NSE at 16:08
Completed NSE at 16:08, 0.23s elapsed
Initiating NSE at 16:08
Completed NSE at 16:08, 0.00s elapsed
Nmap scan report for 192.168.0.233
Host is up (0.0041s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE          VERSION
80/tcp    open  http             Apache httpd 2.4.38 ((Raspbian))

```

Figure 6.2 Port Scanning

6.5.1.2 Server Side

Once attacker has launched the code, server side will auto capture all the incoming and outgoing packet in the network. If there have any unusual traffic, the barnyard2 will be show the alert on its terminal. Figure 6.3 shows the alert that will be receive during the attacking phase. It shows the source and destination IP with the port number where the packet has been sent. The time and date of the attack also will be shown.

```

barnyard2.sh
File Edit Tabs Help
|o" |>| By Ian Firms (SecurixLive): http://www.securixlive.com/
+ '***' + (C) Copyright 2008-2013 Ian Firms <firnsy@securixlive.com>

Using waldo file '/var/log/snort/barns$':
  spool directory = /var/log/snort
  spool filebase  = snort.u2
  time_stamp     = 1630742338
  record_idx     = 0
Opened spool file '/var/log/snort/snort.u2.1630742338'
Closing spool file '/var/log/snort/snort.u2.1630742338'. Read 0 records
Opened spool file '/var/log/snort/snort.u2.1630742880'
Waiting for new data
09/04-16:08:14.552940  [**] [2:10000002:2] Snort Alert [2:10000002:2] [**] [Classification ID: 0] [Priority ID: 0] {TCP} 192.168.
0.100:13354 -> 192.168.0.233:22
09/04-16:08:14.652591  [**] [2:10000002:2] Snort Alert [2:10000002:2] [**] [Classification ID: 0] [Priority ID: 0] {TCP} 192.168.
0.100:13427 -> 192.168.0.233:22
09/04-16:08:15.057148  [**] [2:10000002:2] Snort Alert [2:10000002:2] [**] [Classification ID: 0] [Priority ID: 0] {TCP} 192.168.
0.100:13354 -> 192.168.0.233:22
09/04-16:08:15.155098  [**] [2:10000002:2] Snort Alert [2:10000002:2] [**] [Classification ID: 0] [Priority ID: 0] {TCP} 192.168.
0.100:13427 -> 192.168.0.233:22
09/04-16:08:15.565347  [**] [2:10000002:2] Snort Alert [2:10000002:2] [**] [Classification ID: 0] [Priority ID: 0] {TCP} 192.168.
0.100:13354 -> 192.168.0.233:22
09/04-16:08:15.660526  [**] [2:10000002:2] Snort Alert [2:10000002:2] [**] [Classification ID: 0] [Priority ID: 0] {TCP} 192.168.
0.100:13427 -> 192.168.0.233:22
09/04-16:08:16.072321  [**] [2:10000002:2] Snort Alert [2:10000002:2] [**] [Classification ID: 0] [Priority ID: 0] {TCP} 192.168.
0.100:13354 -> 192.168.0.233:22
09/04-16:08:16.168518  [**] [2:10000002:2] Snort Alert [2:10000002:2] [**] [Classification ID: 0] [Priority ID: 0] {TCP} 192.168.
0.100:13427 -> 192.168.0.233:22
09/04-16:08:16.577813  [**] [2:10000002:2] Snort Alert [2:10000002:2] [**] [Classification ID: 0] [Priority ID: 0] {TCP} 192.168.
0.100:13354 -> 192.168.0.233:22
09/04-16:08:16.676057  [**] [2:10000002:2] Snort Alert [2:10000002:2] [**] [Classification ID: 0] [Priority ID: 0] {TCP} 192.168.
0.100:13427 -> 192.168.0.233:22
  
```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
Figure 6.3 Snort Alert Testing

6.5.2 Alert To Database Testing

All the alert that shown on barnyard2 panel will auto insert into a database snort on table event. The class of the attack is classified using signature attribute. Figure 6.4 shows the signature that has been assign to the each of type alert. Signature 523 has been assigned to the TCP port scanning while 513 are assign for ICMP.

Showing rows 50 - 61 (62 total, Query took 0.0048 seconds.)

```
SELECT * FROM `event`
```

Options: Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Number of rows: 25 Filter rows: Search this table

	sid	cid	signature	timestamp
<input type="checkbox"/>	1	18428	513	2021-09-04 15:51:38
<input type="checkbox"/>	1	18429	513	2021-09-04 15:51:39
<input type="checkbox"/>	1	18431	523	2021-09-04 16:08:14
<input type="checkbox"/>	1	18432	523	2021-09-04 16:08:14
<input type="checkbox"/>	1	18433	523	2021-09-04 16:08:15
<input type="checkbox"/>	1	18434	523	2021-09-04 16:08:15
<input type="checkbox"/>	1	18435	523	2021-09-04 16:08:15
<input type="checkbox"/>	1	18436	523	2021-09-04 16:08:15
<input type="checkbox"/>	1	18437	523	2021-09-04 16:08:16
<input type="checkbox"/>	1	18438	523	2021-09-04 16:08:16
<input type="checkbox"/>	1	18439	523	2021-09-04 16:08:16
<input type="checkbox"/>	1	18440	523	2021-09-04 16:08:16

Console Check all With selected: Edit Copy Delete Export

Figure 6.4 Database Testing

6.5.3 System Testing

Once the user enters the address of the web, the system will show the alert that notified user to login to use the system. This alert will only show if system does not detect any active session. The interface of the first page of the website has been shown in Figure 6.5.

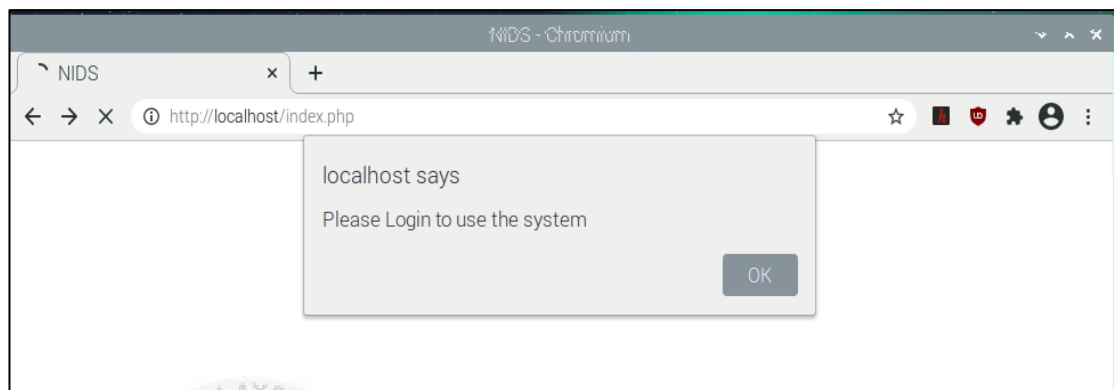


Figure 6.5 System Testing

6.5.3.1 Login Testing

Click the login menu on the side navigation bar. The user will be redirect to the login page. Enter username and password and click login. The login interface as shown in Figure 6.6.

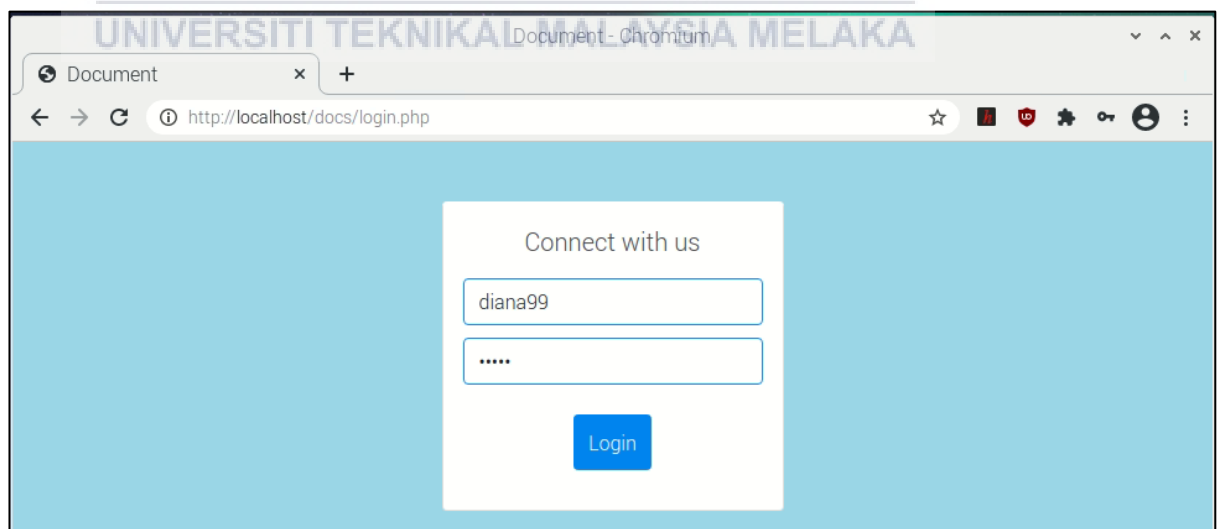


Figure 6.6 Login Testing

6.5.3.2 Dashboard Testing

Once the login button issued, the system will filter the entered username and password. If the combination is match with database. The system will redirect to the index with full menu shown in Figure 6.7. User can analyze the number of packets that has been receive by looking at the today threat. Alert with message either from TCP scan or ICMP scan will be shown in the real-time.

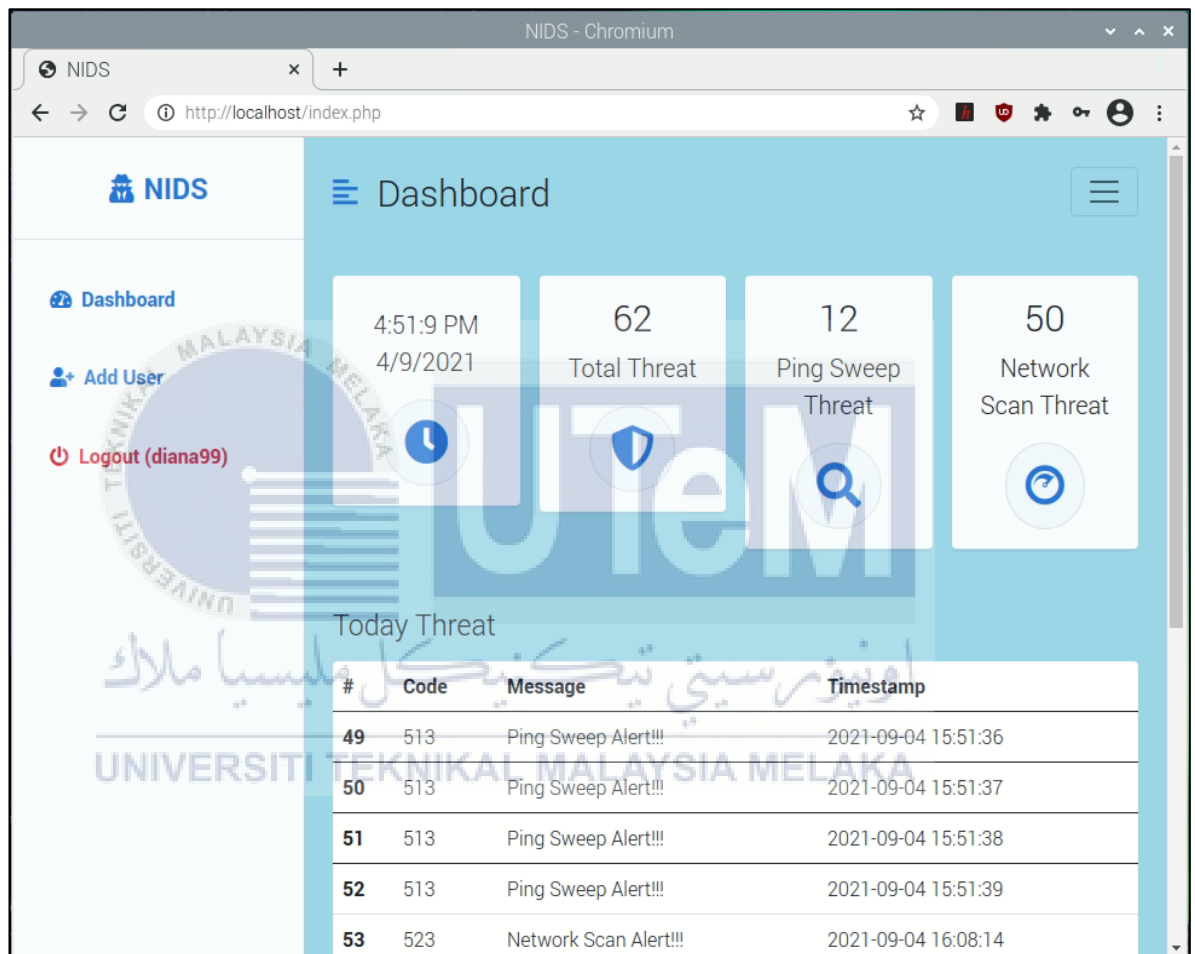
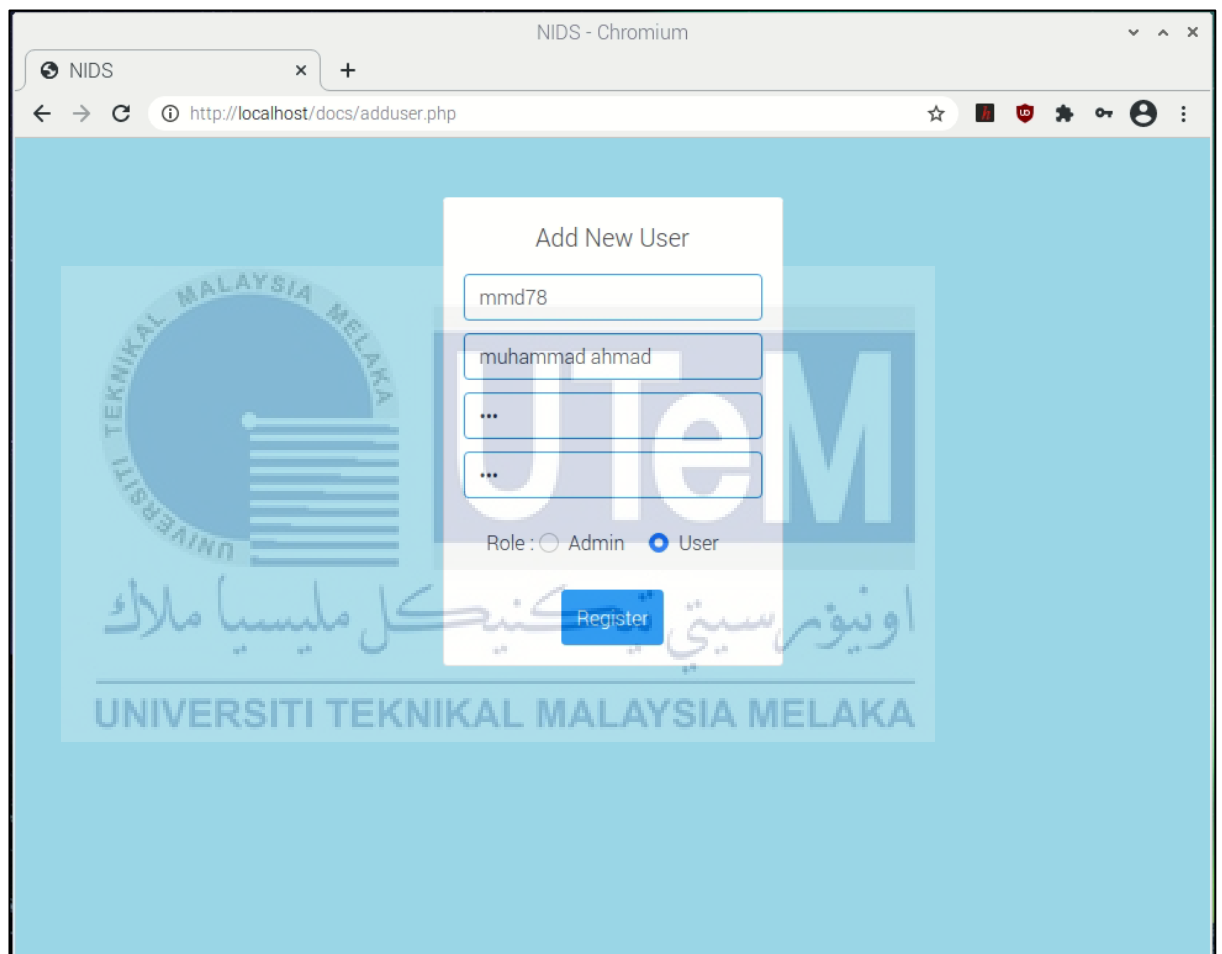


Figure 6.7 Dashboard

6.5.3.3 Add New User Testing

If the user login with the admin account, the additional menu will be shown in the side navigation bar. User will have privilege to add a new user in the system. Once menu have been clicked, user is required to fill the form and choose the role for new user. After all the form is filled, click the register button to submit the form request. Figure 6.8 show the requirement for adding new user which need the username, full name, password and role.



The screenshot displays a web browser window titled "NIDS - Chromium" with the address bar showing "http://localhost/docs/adduser.php". The main content area features a light blue background with the UTeM logo and the text "UNIVERSITI TEKNIKAL MALAYSIA MELAKA" and "اونيورسي تيكنيكل مليسيا ملاك". Overlaid on this is a white form titled "Add New User". The form contains the following fields and controls:

- Username: mmd78
- Full Name: muhammad ahmad
- Two password fields, each with a "..." indicator.
- Role selection: Admin and User.
- A blue "Register" button.

Figure 6.8 Add New User

If the form does not have any errors, the alert will be prompted to notified user regarding registration success. When OK button clicked, user will redirect to the index page. Successfully adding new user interface as shown in Figure 6.9.

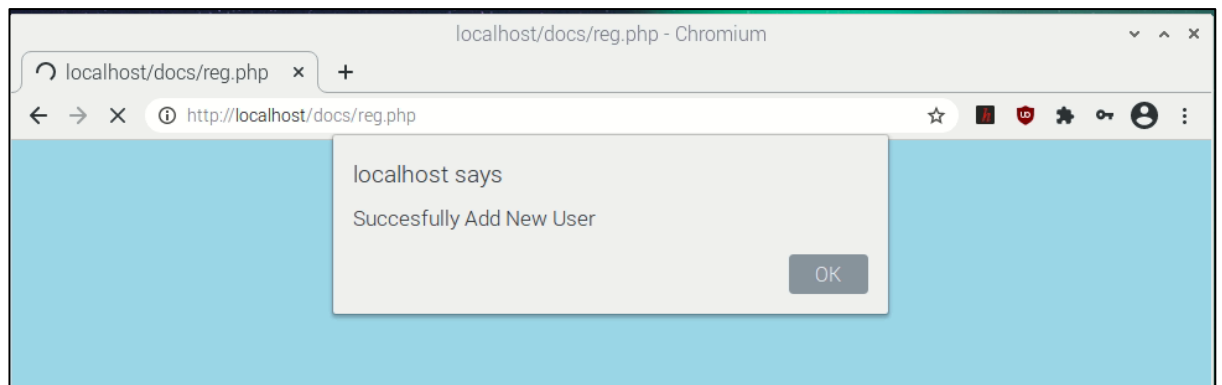


Figure 6.9 Successfully Adding New User

6.5.3.4 Threat Show Counter Testing

The system is built by using the real time concept. This means, all the threat and record will auto show if there have any changes in database. User no need to refresh the web page to see the latest data. Figure 6.10 has been shown the threat show counter testing.

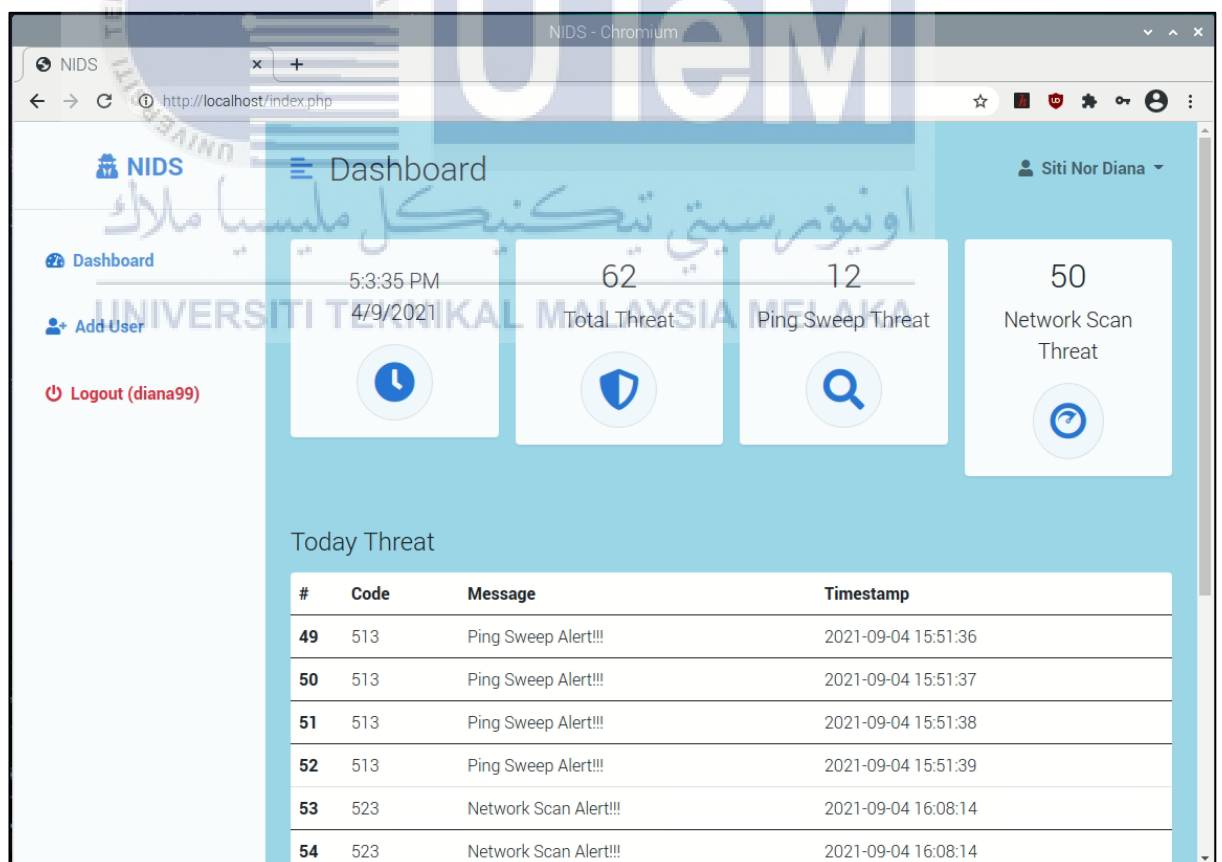


Figure 6.10 Threat Show Counter Testing

6.5.3.5 Logout Testing

When the logout button is submitted, the prompt will notify the user that they have success logout to the system. Together with this, the active session will be destroyed, and user is required to login the system in the future. Figure 6.11 show the interface that user would be see.

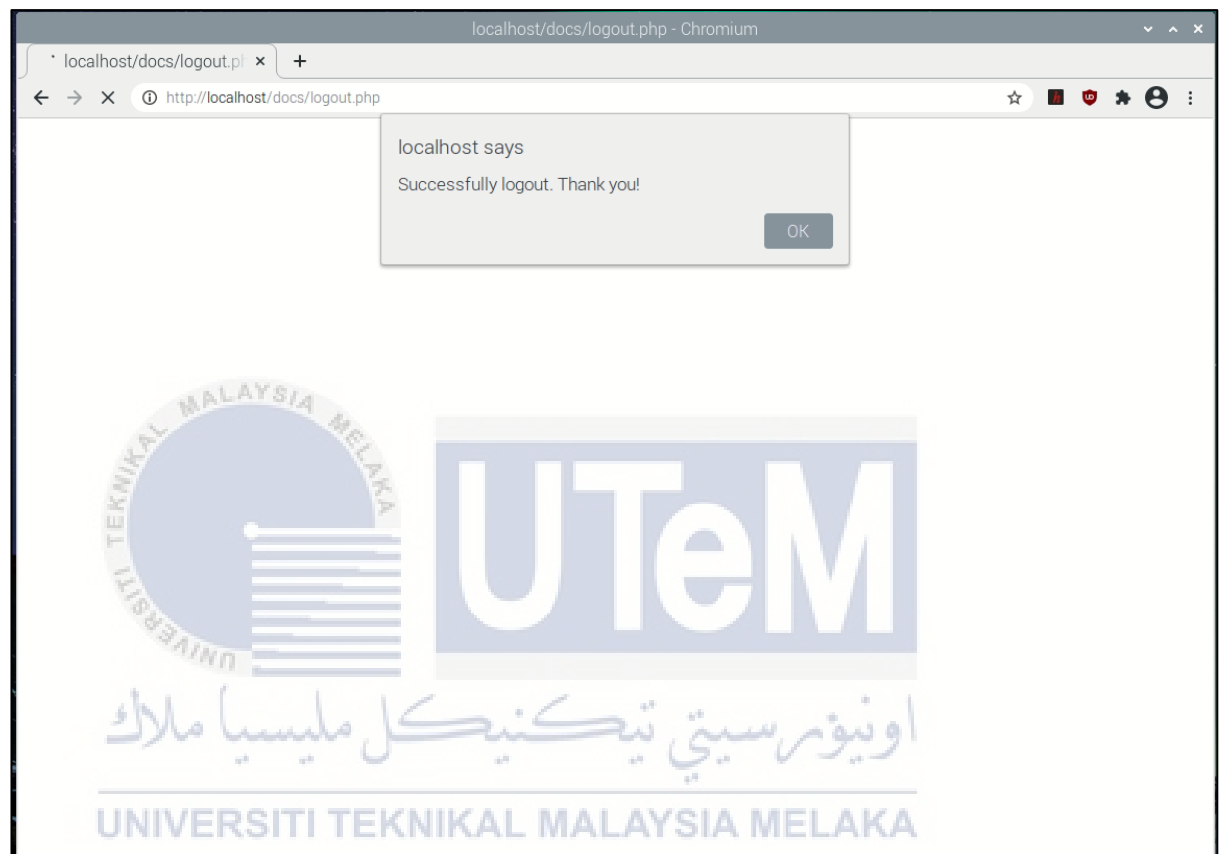


Figure 6.11 Logout Testing

6.5.4 Telegram Notification Testing

When the server has detected any unusual activities, all the alerts will also send to telegram private channel by using BotFather API that set on the script as in previous chapter. Figure 6.12 shows the alert that has been receive in the Telegram software where it shows the IP address of source and destination with port number. It also shows the alert are TCP or ICMP class with the time and date.

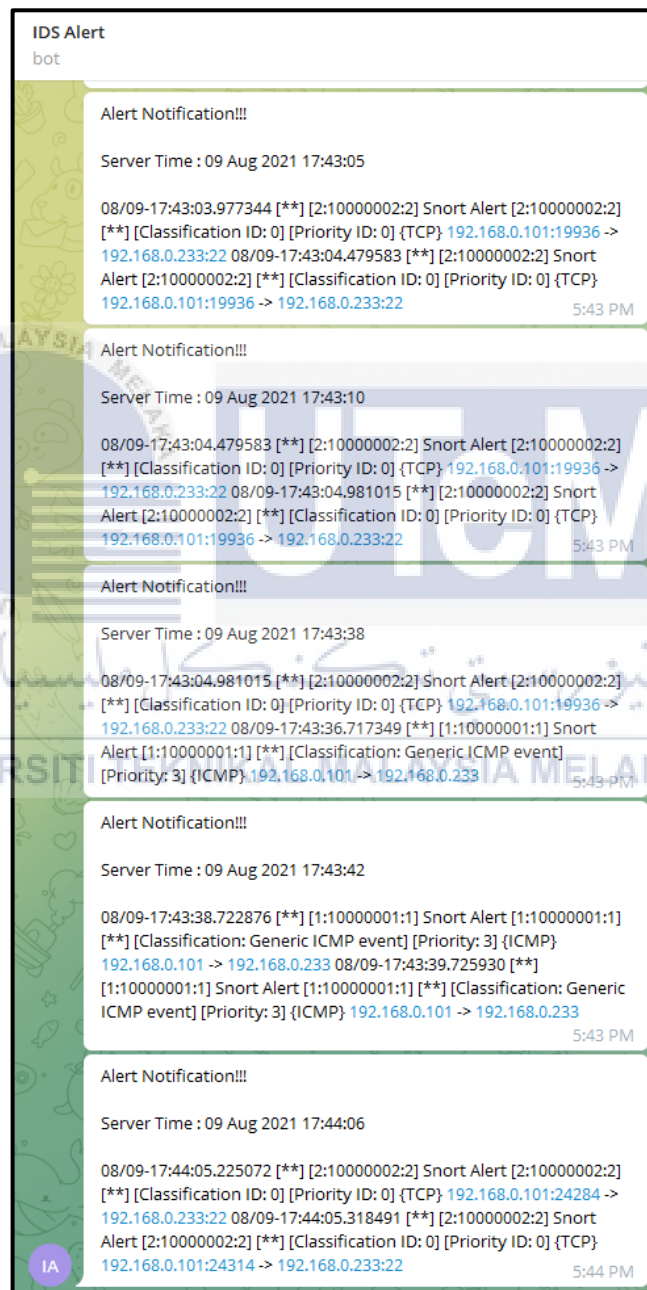


Figure 6.12 Telegram Alert

6.5.5 Analysis of Ram Usage When Execute All the Script Needed

Figure 6.13 shows the status when nothing is running in the background. It shows the free space is 173MB. After starting the necessary script like apache2, snort, barnyard2 and telegram scripting file. The ram usage is shown as Figure 6.13 where the free space left 89MB.

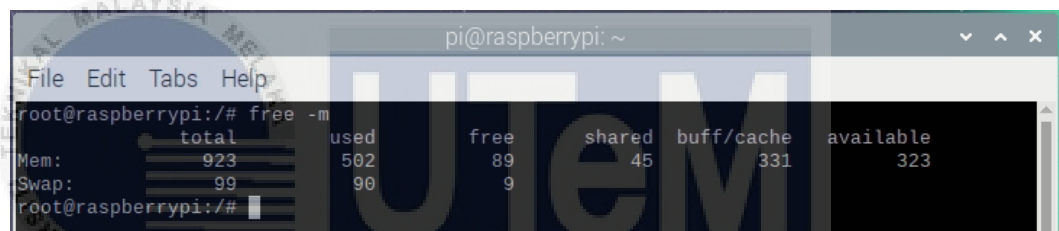


```

pi@raspberrypi: ~
File Edit Tabs Help
root@raspberrypi:/# free -m
      total        used          free      shared  buff/cache   available
Mem:    923         432          173           43         317         395
Swap:   99           94            5
root@raspberrypi:/#

```

Figure 6.13 Size of RAM Without Any Process



```

pi@raspberrypi: ~
File Edit Tabs Help
root@raspberrypi:/# free -m
      total        used          free      shared  buff/cache   available
Mem:    923         502            89           45         331         323
Swap:   99           90
root@raspberrypi:/#

```

Figure 6.14 Size of RAM With Process

It shows that the ram increase to 9% after running the script file. This is analyzed only with executed script. The comparison of the usage RAM is illustrated as Figure 6.15.

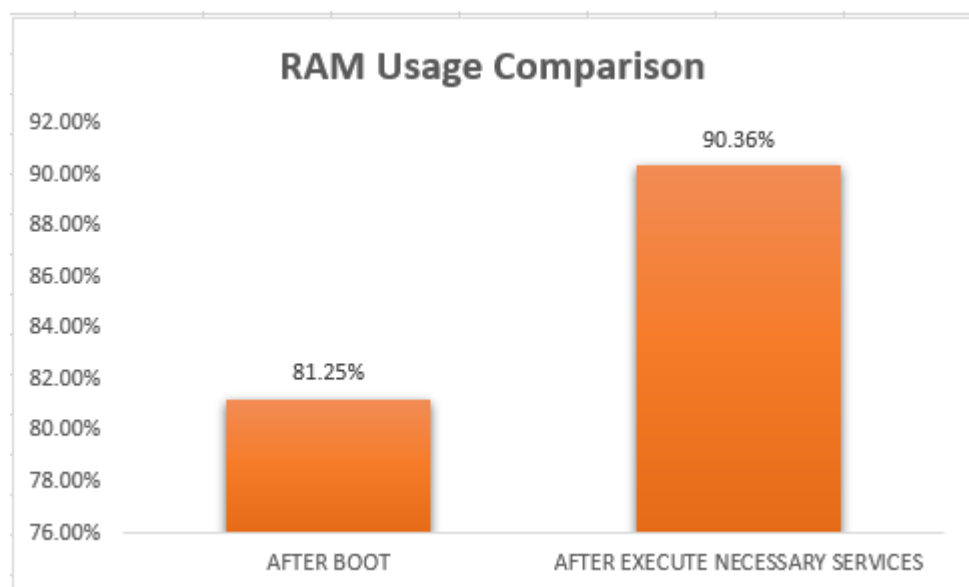


Figure 6.15 RAM Usage Comparison

6.6 Conclusion

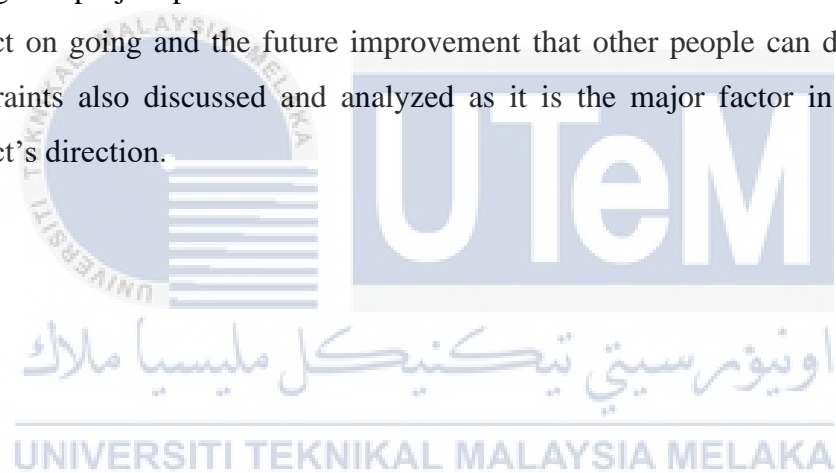
In conclusion, this chapter is required and its mandatory after the implementation phase is finish. In this chapter, it will give the result whether it is success or not. If the proposed system has some failure or bug, it needs to go back to the implementation phase to patch the bug. All the outcomes for this project will be used in the next chapter to conclude whether this proposed project is success and meets all the goals requirements that need to be achieve as stated in chapter 1. Besides, this chapter also will conclude that whether this proposed solution can be used or not in real life.



CHAPTER 7: PROJECT CONCLUSION

7.1 Introduction

This chapter will be the last chapter where it will conclude the entire projects result and finding. In this chapter will be deeply explain how the objective has been achieved during this project process. It also will be discussed about the limitations during this project on going and the future improvement that other people can do with it. The constraints also discussed and analyzed as it is the major factor in affecting this project's direction.



7.2 Project Summarization

This project is about intrusion detection system alerting system using raspberry pi 3. This project objective is to implement IDS in RPi. This objective has been achieved by installing snort in the RPi and the snort can get the alert by using simple rules. Second objective is to monitor and analyze the number of packets that coming to the network by using website. This objective is working on the attacker side, where the attacker function to launch the port scanning and see whether the snort can detect that threat or not. The main contribution in this project is to implement the alert function system in RPi. To achieve this objective, I have been used Telegram (BotFather) as a bot to received alert from the database and send to the telegram. Snort will communicate with the API server, then send the log file to send to the administrator. This port scanning known as abnormal since the attacker will know which port that has open. So, the attacker will try to attempt to attack regarding the type of port that has been open. So, the telegram will be sending the alert to show that someone has been trying to attack by checking the port vulnerabilities. It is because port scanning is the early indicator for a network. The significant of this project is user can more aware if there any threat incoming to their network. My project weaknesses are this project are set to localhost only, so that the attacker need to be in the same network in order to launch the attack. Moreover, user or administrator need to be in the same network to monitor and view the threat using web page. My project strength is the alert that has been received will be received in the real time. It just takes a minor delay time to received either in telegram or webpage, both are updating in the real time.

7.3 Project Contribution

This project contributes to the home user that requires security elements to protect their home network environment. This project can be greater contribution to the home where use high technology appliances at lower cost. This project will send the alert notification to the user that can notify user if has any unusual activities in their network. If the port scanning has been reached for more than So that user will be acknowledged that their network has been attack, and they can take any other precaution and prevention to avoid the attack. They can prepare their network by off

the network or block the IP that has been trying to breach their network. It is because the alert that has been sending are in real time, so the user can know on the spot.

7.4 Project Limitation

This project limitation was the memory and processor limitation. This is because RPi 3 using 1GB RAM and Quad Core 1.2GHz Broadcom BCM2837 64bit CPU. It causes to lag because RAM usage was huge. If there are multiple process that running in the background, the RPi RAM increase to 9% that lead to high usage. Second is about limitation where it set up to localhost only, so that the attacker needs to be in the same network in order to launch the attack. Moreover, user or administrator need to be in the same network to monitor and view the threat using web page. Furthermore, to get the alert from telegram only one user can get the alert. Lastly, the server does not setup domain name system so that it would be difficult to user surfing the webpage.

7.5 Future Works

Nowadays, IDS become more important for each home since all the workers need to Work from Home (WFH). The attacker will take advantage during this period. Then, this project could be used for any home user, but it can be improved by using larger memory and latest version of RPi where it has large RAM and more powerful processor with installation Ubuntu OS. This is because Raspbian OS has a limitation to install certain software that lead to the software cannot work properly. Next, to make more organizable it can setup a proper environment in network monitoring room with server rack. Furthermore, set domain to the website so that user can monitor it easily by entering URL and anywhere there are. Lastly, to make it more complete and more functioning, it can create other rules and more specific to detect the threat such as threat to detecting DDoS, malware, buffer overflow and others.

7.6 Conclusion

Although the project has many constraints and limitation, but this project managed to be fulfilled and achieve the objective successfully. However, some highlighted future improvements can still be done for a better use of the tool developed. Any proposed adjustment made should always at least maintain a high accuracy detection for network attacks. This is crucial to further secure the network by detecting incoming attacks beforehand. In this way, the general aim of any network setup which is to ensure secured data transmission are guaranteed and further loss can be avoided. Hopefully, this project will give a better future although it does not eliminate all the cybercrime cases, but it will minimize the risk getting threat in our network.



REFERENCES

- Ahmed, A. A., Kit, Y. W., & Sallam, A. A. (2018). *Raspberry Pi-Based Investigating Model for Identifying Intrusion Evidence*. February. <https://doi.org/10.19080/JFSCI.2018.07.555715>
- Jeremiah, J. (2019). Intrusion Detection System to Enhance Network Security Using Raspberry Pi Honeypot in Kali Linux. *2019 International Conference on Cybersecurity, ICoCSec 2019*, 91–95. <https://doi.org/10.1109/ICoCSec47621.2019.8971117>
- Jose, S., Malathi, D., Reddy, B., & Jayaseeli, D. (2018). A Survey on Anomaly Based Host Intrusion Detection System. *Journal of Physics: Conference Series*, 1000(1). <https://doi.org/10.1088/1742-6596/1000/1/012049>
- Karahan, O., & Kaya, B. (2020). Raspberry Pi Firewall and Intrusion Detection System. *Journal of Intelligent Systems: Theory and Applications*, 3(2), 21–24. <https://doi.org/10.38016/jista.653486>
- Kyaw, A. K., Chen, Y., & Joseph, J. (2016). Pi-IDS: Evaluation of open-source intrusion detection systems on Raspberry Pi 2. *2015 2nd International Conference on Information Security and Cyber Forensics, InfoSec 2015*, 165–170. <https://doi.org/10.1109/InfoSec.2015.7435523>
- Parag Vadher. (2020). Snort IDPS using Raspberry Pi 4. *International Journal of Engineering Research And*, V9(07), 151–154. <https://doi.org/10.17577/ijertv9is070099>
- Sabekti, M. A. (2018). *Pembuatan Web Interface Snort*.
- Tripathi, S., & Kumar, R. (2018). Raspberry Pi as an Intrusion Detection System, a Honeypot and a Packet Analyzer. *Proceedings of the International Conference on Computational Techniques, Electronics and Mechanical Systems, CTEMS 2018*, 80–85. <https://doi.org/10.1109/CTEMS.2018.8769135>

APPENDIX A

In this appendix A, it contains the script for the Telegram notification and snort rules.

A. Telegram script

```

GNU nano 3.2 bot-tele.sh

#!/bin/bash

#init
initCount=0
logs=/home/pi/Desktop/converting.txt

#File
msg_caption=/tmp/telegram_msg_captign.txt

#Chat ID dan bot token Telegram
chat_id="283686038"
token="1945640027:AAGshxka0yyWXqIERzkBNGlgxSTPJD9qc2o"

#kirim
function sendAlert
{
    curl -s -F chat_id=$chat_id -F text="$scaption" https://api.telegram.org/bot$token/sendMessage #> /dev/null 2>&1
}

#Monitoring Server
while true
do
    lastCount=$(wc -c $logs | awk '{print $1}') #getSizeFileLogs
    #DEBUG ONLY
    #echo before_last $lastCount #ex 100 #after reset 0
    #echo before_init $initCount #ex 0
    #echo "-----"

    if((${lastCount}) > ${initCount});
    then
        #DEBUG
        #echo "Sending Alert..."
        msg=$(tail -n 2 $logs) #GetLastLineLog
        echo -e "Alert Notification!!!\n\nServer Time : $(date +"%d %b %Y %T")\n\n" $msg > $msg_caption #set Caption / Pesan
        caption=$(cat $msg_caption) #set Caption
        sendAlert #Panggil Fungsi di function
        echo "Alert Sent"
        initCount=$lastCount
        rm -f $msg_caption
        sleep 1
    fi
    sleep 2 #delay if Not Indication
done

```

B. Snort rules

```

GNU nano 3.2 local.rules

alert icmp any any -> 192.168.0.233 any (msg:"ICMP PING SWEEP ALERT!!!"; classtype:icmp-event; sid:10000001; rev:001; GID:1; )
alert tcp any any -> 192.168.0.233 22 (msg:"NMAP TCP SCAN ALERT!!!"; sid:10000002; rev: 002; GID:2;)
#alert tcp any any -> 192.168.0.233 80 (msg:"NMAP TCP SCAN ALERT PORT 80!!!"; sid:10000003; rev: 003; GID:3;)

```