ML Based Intrusion Detection System

A Project Report

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

MASTERS OF SCIENCE (INFORMATION TECHNOLOGY)

By

Deepak Varma

Roll Number - 306

Under the esteemed guidance of

Prof. Pinky Panda

Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY

GURU NANAK KHALSA COLLEGE

OF

ARTS, SCIENCE & COMMERCE

(Autonomous)

MATUNGA, MUMBAI – 400 019

MAHARASHTRA

YEAR 2021– 2023

GURU NANAK

KHALSA COLLEGE

OF

ARTS, SCIENCE & COMMERCE

(Autonomous)

MATUNGA, MUMBAI, MAHARASHTRA – 400 019

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled," **ML Based Intrusion Detection System**", is bonafied work of **Deepak Ashok Varma** bearing Seat No: **306** submitted in partial fulfillment of the requirements for the award of degree of MASTERS OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Internal Guide **Coordinator**

External Examiner

Date: **College Seal**

SYNOPSIS

ML Based Intrusion Detection System

INTRODUCTION:

Internet services have become essential to business commerce as well as to individuals. With the increasing reliance on network services, the availability, confidentiality, and integrity of critical information have become increasingly compromised by remote intrusions. Enterprises are forced to fortify their networks against malicious activities and network threats. Therefore, a network system must use one or more security tools such as a firewall or an intrusion detection system to protect important data or services from intruders.

Relying on a firewall system alone is not sufficient because a firewall cannot defend the network against intrusion attempts on open ports required for network services. Hence, an intrusion detection system (IDS) is usually installed to complement the firewall. An IDS collects information from a network or computer system and analyses the information for symptoms of system breaches. A network IDS monitors network data and gives an alarm signal to the computer user or network administrator when it detects antagonistic activity on an open port. This signal allows the recipient to inspect the system for more symptoms of unauthorized network activities.

In this project, I have focused on real-time network-based intrusion detection where the incoming network data is captured on-line and the detection result is reported instantaneously or within a fraction of a minute, so that the network administrator is notified and can stop the on-going attack. Our approach also could be applied as host-based detection. There are many possible features of network data that could serve as input to an IDS, we propose to consider only 12 features of network traffic data extracted from the headers of data packets. We show that these 12 features are effective in identifying normal network activity and classifying main Denial of Service (DoS). Using a small number of features reduces the complexity of data analysis and thus can increase the detection speed and reduce computer resource (CPU and memory) consumption.

Software Requirements:

- Linux
- Python, SQL
- VM WARE

Hardware Components:

- **Operating System (**Microsoft® Windows® 7/8/10 (64-bit)**):**

  For using an Android emulator, you'll most likely need the 64-bit version of Windows 10. Most android emulators require this as a minimum and don't work with 32-bit Windows.
- **Processor:**

  Most VMware require an Intel processor, but some also have support for AMD processors. The Intel processor should have support for Intel ® VT- x Intel ® EM64T(Intel ® 64),and execute Disable (XD) Bit functionality
- **Memory:**

  For using VMware, you'll need 8 GB RAM minimum, 16 GB RAM recommended. 25 GB of available disk space minimum.

- **Storage:**

  Every emulator has different storage requirements, but you should keep at least 25 GB in mind. You'll need more of it once you start using the emulator and download all your games. Games will also get stored on the hard drive, so make sure you have sufficient space
- **Screen Resolution**:

  Assuming you're downloading an Android emulator for playing games, you would want a high-definition screen resolution. However, for most emulators, the minimum screen resolution is 1280 x 900 pixels. This is one of those things that can enhance the quality of graphics.
- **Graphics Driver**:

  Make sure to update your graphics driver on Windows 10, as that's recommended for using VMware. An old graphics driver may not fully support or optimize graphics for Android games.

Advantages:

- Verifies success or failure of an attack: Since a host based IDS uses system logs containing events that have actually occurred, they can determine whether an attack occurred or not.
- Monitors System Activities: A host-based IDS sensor monitors user and file access activity including file accesses, changes to file permissions, attempts to install new executables etc.
- Detects attacks that a network based IDS fail to detect: Host based systems can detect attacks that network based IDS sensors fail to

detect.

- Near real time detection and response: Although host based IDS does not offer true real-time response, it can come very close if implemented correctly.

Disadvantages:

- Host based IDSs are harder to manage, as information must be configured and managed for every host.
- The information sources for host based IDSs reside on the host targeted by attacks, the IDSs may be attacked and disabled as part of the attack.
- Host based IDSs are not well suited for detecting network scans or other such surveillance that targets an entire network.
- Host-based IDSs can be disabled by certain denial-of- service attacks.

DECLARATION

I hereby declare that the project entitled, “**ML Based Intrusion Detection System**” done at **Guru Nanak Khalsa College**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfilment of the requirements for the award of degree of **MASTERS OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Deepak Varma

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.
I am thankful to our institution for giving me the opportunity to do this project which came up with a lot of fruitful lessons.

I am highly indebted to my Professor **Mrs. Pinky Panda** Mam as well as our principal **Mr. H.S Kalsi** sir for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. I am also thankful to HOD of our department **Mrs. Jasbir kaur** ma'am who invested her valuable time into inspiring us to do a creative project.

I would like to express my gratitude towards my parents and friends for their kind co-operation and encouragement which helped me in completion of this project.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

Last but not least, I would like to thank God who made me able to do this project with patience and determination.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Background

1.2 Objectives

1.3 Purpose, Scope, and Applicability

1.3.1 Purpose

1.3.2 Scope

1.3.3 Applicability

1.4 Organization of Report

CHAPTER 2: SURVEY OF TECHNOLOGIES

2.1 Introduction

2.2 Existing System

2.3 Existing Solutions in the Market

2.4 Proposed System

2.5 Justification of Proposed System

CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1 Problem Definition

3.2 Requirements Specification

3.3 Planning and Scheduling

3.3.1 GANTT Chart

3.3.2 WBS Chart

3.4 Software and Hardware Requirements

CHAPTER 4: SYSTEM DESIGN

4.1 Basic modules

4.2 Process Model

4.3 Procedural Design

- 4.3.1 Use Case Diagram
- 4.3.2 Data Flow Diagram
- 4.3.3 Sequence Diagram
- 4.3.4 Activity Diagram

4.4 User interface design

4.5 Test Cases

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Implementation

5.2 Coding Details and Details

5.3 Testing

CHAPTER 6: CONCLUSION

6.1 Conclusion

6.2 Future Scope

CHAPTER 7: REFERENCES

CHAPTER-1

INTRODUCTION

1.1 Background

In any organization value of data may range from miniscule to invaluable. Sensitive Data leaks cause massive loss to an organization. To prevent such scenarios, various organization has started researching in improving network security to protect itsvaluable or sensitive data. This can be achieved by using an Intrusion Detection System (IDS) that helps to detect malicious attacks and network intrusions. An Intrusion detection system is deployed at a network junction or on a host system, an IDS can either be a host or a network-based IDS which are used according to an organization's requirement. Once a malicious activity or an intrusion is detected, it is flagged, or an alarm is raised so that the attack can be isolate.

Machine Learning techniques are widely used to improve existing systems or environments, to achieve the best possible result. Similarly using Machine Learning technique in Networking is an emerging field. Various work has been done in thisfield using Machine Learning techniques to achieve better result than using traditionaltechniques. Exponential increase in information has attracted a lot of interest in CyberSecurity. Various on-going research are focused on improving existing network tools to achieve better network security. Many studies are conducted to improve Intrusion Detection System (IDS) which helps to monitor a network or a host-based system anddetects malicious activities. The existing IDS have low detection rate, high false alarm rate and are unable to detect novel attacks. To overcome these problems, many researches are focused on utilizing Machine Leaning methods to discover abnormal patterns to improve Detection rate.

1.2 Objectives

- Study and analyze various ML and DL Algorithms:

To get a clear idea about different available technique and which technique we can use in our project we study various machine learning technique and implement some using KDD and our own generated dataset

- Study working of IDS and Network:

Studies the traditional IDS system identified shortcomings in the system and think about ways to overcome them such as Detection rate and False alarm rate.

- Building network monitoring tool (IDS):

Using Python to building a network monitoring program capable of sniffing packets.

- Improving Security with integrating Machine Learning in IDS:

Implementing ML technique on IDS to improve Detection rate and Reduce false positive

- Design a GUI for end user:

For end user build a GUI which provide a user-friendly way to operate the whole operation and which helps in easy monitoring of network.

1.3 Purpose, Scope and Applicability

1.3.1 Purpose

The purpose of this system is to create a platform where users can seek and provide help to others. Due to the excessive increase in volume of data, protection and security of such a sensitive information is major concern of organization. IDS being an of the important security tools to fight against such malicious activity. But the traditional IDS alarm report of intrusion to network and detection accuracy gets reduced with such a large network monitoring chuck data. This is one of the major issues when the system encounters unknown attacks or zero-day attack. Due to this, IDS has decreased accuracy rate and to Increased false alarm rate.

1.3.2 Scope

The goal of this project is to analyze and study various Machine Learning Algorithms and traditional IDS. And design IDS using Machine Learning to improve detection rate, reduce false alarm rate and a system with a better capability of distinguishing attacks from normal packets. The model will be integrated with a User Interface (UI) which will provide a more user-friendly way to monitor the network activity

1.3.3 Applicability

This application could be used by any general person, or any person who wants to prevent malicious attack or any expert who themselves want to guide people. Research for improving different network security tool to provide better security is the need of time. So, an Intrusion Detection System (IDS) using different Machine Learning algorithm which can automatically discover the essential differences between normal data and abnormal data with high accuracy. In addition, machine learning methods have strong generalizability, so they are also able to detect unknown attacks. And thereby increases detections rate of attacks and reduces false alarm rate in IDS which issues is being faced by the traditional IDS. This system can help us achieve better network security.

1.4 Organization of Report

- Survey Of Technologies
- Requirements and Analysis
- System Design

CHAPTER-2

SURVEY OF TECHNOLOGIES

2.1 Introduction

The artificial neural network approach is one of the most popular techniques for the design of IDS. Jirapummin et al. [1] proposed a hybrid neural network using a combination of Self-Organizing Map (SOM) and Resilient Back-Propagation Neural Network (BPNN). To evaluate their approach, they used an available well known pre-processed dataset which is KDD99 [2]. The KDD99 dataset is a network packet dataset consisting of normal network activity as well as many network attack types. The dataset is based on the DARPA98 dataset from MIT Lincoln laboratory, which provides answer class (labelled data) for evaluation of intrusion detection. Pan et al. [3] designed a hybrid system by using a BPNN and a C4.5 Decision Tree considering the KDD99 dataset. The results showed that using only a BPNN without C4.5 Decision Tree, their system could not detect the network attack types such as User to Root (U2R) and Root to Local (R2L) at all. Moradi and Zulkernine [4] used a Multi-Layer Perceptron (MLP) artificial neural network in off-line mode to classify normal network activity, Satan (Probe) attacks and Neptune attacks using the KDD99 dataset. Ngamwitthayanon et al. [5] designed a multi-state IDS system to classify normal data and each attack type using the KDD99 dataset. Their results showed a higher detection rate in each classification category than when only a single state was used to classify all categories.

Labib and Vemuri [6] developed a real-time IDS using SelfOrganizing Maps (SOM) to detect normal network activity and DoS attack. They pre-processed their dataset to have 10 features for each data record. Each record contained information of 50 packets. Their IDS was evaluated by human visualization for different Fig. 1. Network intrusion detection system environment. 2228 P. Sangkatsanee et al. / Computer Communications 34 (2011) 2227–2235 characteristics of normal data and DoS attack. No detection rate was reported. Puttini et al. [7] used a Bayesian classification model for anomaly detection to classify normal network activity and attack using a 3-month training dataset and a 1-month test dataset. They evaluated their approach by adjusting a penalty value to see how it affected the classification results. They also needed human expert to visualize the normal and abnormal network behaviors. No detection rate was reported. Amini et al. [8] designed a real-time IDS using two unsupervised neural network algorithms which are Adaptive Resonance

2.2 Existing Systems

Currently, If someone needs any kind of help they have to manually contact through various trustable organizations who provide a platform for these but people are sometimes unaware of their existence and very often don't come across them. Intrusion Detection System (IDS) is a security system that serves the infrastructure as a protective layer. The IDS technology has grown tremendously over the years to keep up with the progress of computer crime. Research has been conducted since the advent of the technology in the mid-80s to improve the ability to detect attacks without losing network performance.

2.3 Existing Solutions in the Market

Organizations can select from a variety of reasonably-priced and powerful IDS and IPS solutions that fit a variety of needs- from startups on a tight budget to global enterprises. Some will be standalone solutions and others will be features added to other security products.

Our guide to selecting the best solution consists of:

- Important Factors in Choosing an IDS or IPS Solution
- Leading IDS and IPS Solutions
- Comparing IDS and IPS Solutions

The approach should be simple but efficient in detecting network intrusion in an actual real-time environment. We describe the concept of information gain and machine learning methods which are applied to our real-time IDS approach in the following section

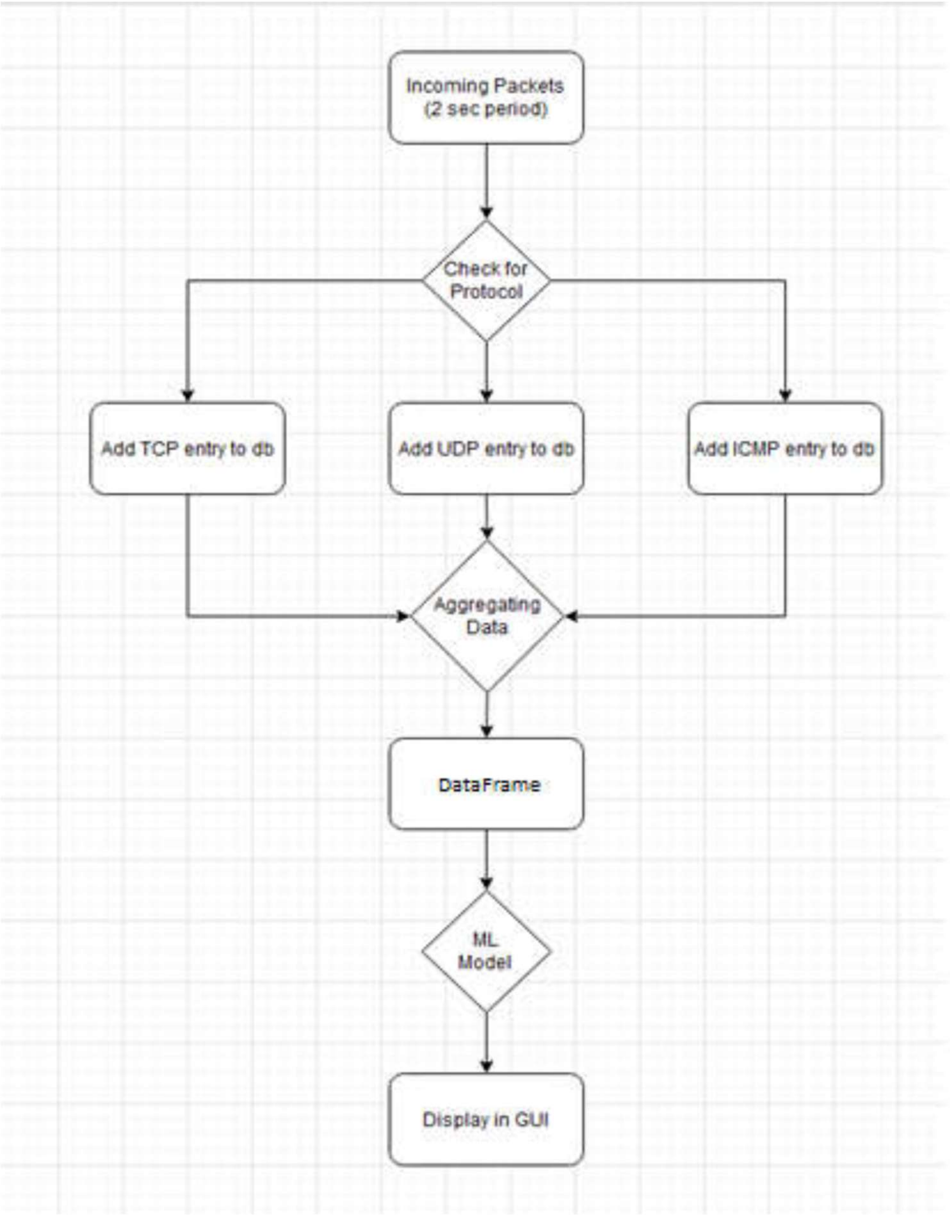
2.4 Proposed System

Our Proposed system has 5 phases. 1st Phase (Packet sniffing) starts with network packet sniffing program here we used python programming language to build the network packets sniffer which scan network port on host machine and packets are sniffed. These packets are formatted respective with source IP addresses, destination IP address, protocol name and network features. The network packets are aggregated based on Source IP address and Destination IP address and these packet data will be captured and logged into a csv file in a designated folder this completes the 2nd face (DB Storage). This aggregated data in CSV will represent the network traffic flowing through the host's end. This CSV file becomes the input for the Machine Learning model which is the 3rd phase (ML model).

The 3rd phase actually has two sub process

a) Training: Here Machine Learning model which is trained on previously captured data by sniffer program. The csv individually produce by are collaborated and labeled together as Attack or normal (1 or 0) respectively and the ML model i.e., SVM is trained on this data set.
b) Implementing: The trained SVM model is used to predict whether its and attack or not and a new column is attached.

The 4th Phase (GUI): The model results are displayed on GUI which is built on python tkinter . Were the attack packets are classified as red and normal packets are shown colorless.



Proposed System

2.5 Justification of Proposed System

Advantages of Proposed System

- Verifies success or failure of an attack: Since a host based IDS uses system logs containing events that have actually occurred, they can determine whether an attack occurred or not.

- Monitors System Activities: A host-based IDS sensor monitors user and file access activity including file accesses, changes to file permissions, attempts to install new executables etc.

- Detects attacks that a network based IDS fail to detect: Host based systems can detect attacks that network based IDS sensors fail to detect.

- Near real time detection and response: Although host based IDS does not offer true real-time response, it can come very close if implemented correctly.

Disadvantages of Proposed System

- Host based IDSs are harder to manage, as information must be configured and managed for every host.
- The information sources for host based IDSs reside on the host targeted by attacks, the IDSs may be attacked and disabled as part of the attack.
- Host based IDSs are not well suited for detecting network scans or other such surveillance that targets an entire network.
- Host-based IDSs can be disabled by certain denial-of- service attacks.

CHAPTER-3

REQUIREMENTS AND ANALYSIS

3.2 Problem Definition

Development of a software for social welfare, An intrusion detection system (IDS) is a device or software application that monitors a network or systems for malicious activity or policy violations. Any malicious activity or violation is typically reported either to an administrator or collected centrally using a security information and event management (SIEM) system. A SIEM system combines outputs from multiple sources, and uses alarm filtering techniques to distinguish malicious activity from false alarms. So, in this we will design a new intrusion portfolio that will addresses the challenges you face every day. Intelligent detectors with false alarm immunity, keypads that are easier to use, panels with more power, and communications solutions that compensate and provide us the tools to create high-quality, reliable solutions for our customers. Support all major communication formats plus internet & StarLink Wireless Radios, universal primary/backup communicator (hi-speed up/downloads from Gemini Panels). Up/downloading, including unique PC-preset unattended

method.

3.3 Requirement Specification

The requirement analysis lists all the required features of the project and characterizes their individual requirements. After performing all the analysis we point out the requirements for designing this software.

A] Hardware

Requirement

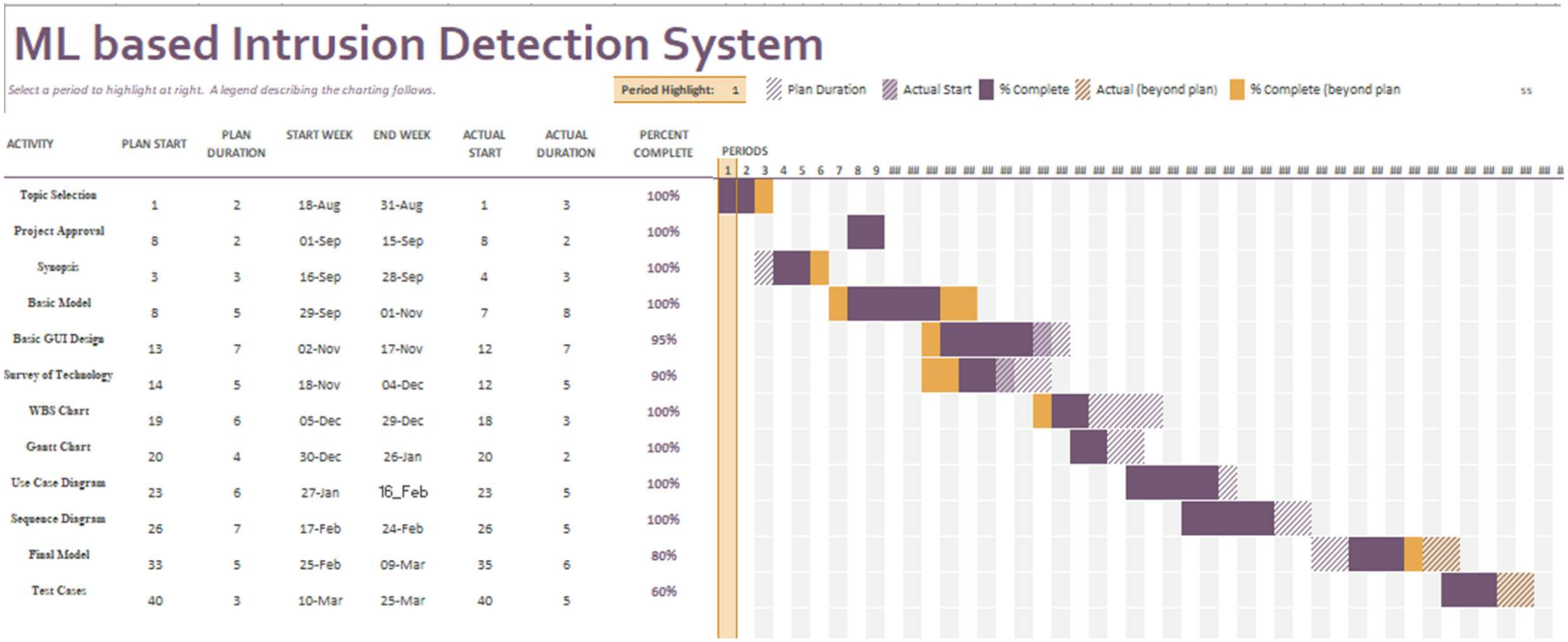
- Computers /Laptops
- Network Card
- Minimum 8 GB RAM.
- Quad core 2 Ghz

B] Software Requirement

- Linux OS
- Python 3.10
- Tkinter
- Vmware

3.4 Planning and Scheduling

3.4.1 GANTT Chart



ML based Intrusion Detection System

Select a period to highlight at right. A legend describing the charting follows.

Period Highlight: 1

▨ Plan Duration ▨ Actual Start ■ % Complete ▨ Actual (beyond plan) ■ % Complete (beyond plan

ACTIVITY	PLAN START	PLAN DURATION	START WEEK	END WEEK	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Topic Selection	1	2	18-Aug	31-Aug	1	3	100%
Project Approval	8	2	01-Sep	15-Sep	8	2	100%
Synopsis	3	3	16-Sep	28-Sep	4	3	100%
Basic Model	8	5	29-Sep	01-Nov	7	8	100%
Basic GUI Design	13	7	02-Nov	17-Nov	12	7	95%
Survey of Technology	14	5	18-Nov	04-Dec	12	5	90%
WBS Chart	19	6	05-Dec	29-Dec	18	3	100%
Gantt Chart	20	4	30-Dec	26-Jan	20	2	100%
Use Case Diagram	23	6	27-Jan	16_Feb	23	5	100%
Sequence Diagram	26	7	17-Feb	24-Feb	26	5	100%
Final Model	33	5	25-Feb	09-Mar	35	6	80%
Test Cases	40	3	10-Mar	25-Mar	40	5	60%

Description of the GANTT Chart

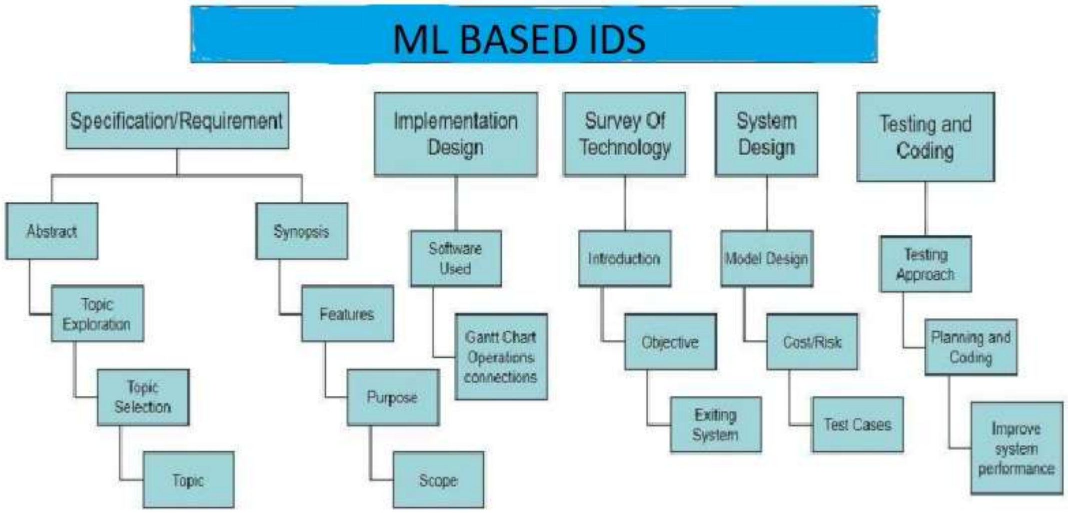
A Gantt chart is a project management tool assisting in the planning and scheduling of projects of all sizes, although they are particularly useful for simplifying complex projects. Project management timelines and tasks are converted into a horizontal bar chart, showing start and end dates, as well as dependencies, scheduling and deadlines, including how much of the task is completed per stage and who is the task owner. This is useful to keep tasks on track when there is a large team and multiple stakeholders when the scope changes.

As it's in a bar chart format it is possible to check on progress with a quick glance. You can easily see:

- a visual display of the whole project,
- timelines and deadlines of all tasks,
- relationships and dependencies between the various activities,
- project phases

Project management solutions that integrate Gantt charts give managers visibility into team workloads, as well as current and future availability, which allows for more accurate scheduling. Gantt charts have been around for nearly a century, having been invented by Henry Gantt, an American mechanical engineer, around 1910.When you set up a Gantt chart, you need to think through all of the tasks involved in your project and divide them into manageable components. Then decide who will be responsible for each task and delegate to the team.

3.4.2 WBS Chart



Description Of WBS Chart

- A Work BreakDown Structure is a deliverable-oriented hierarchical decomposition of the work to be executed by the project team to accomplish the project objectives and create the required deliverables.
- The Work Breakdown Structure (WBS) is developed to establish a common understanding of project scope. It is a hierarchical description of the work that must be done to complete the deliverables of a project. Each descending level in the WBS represents an increasingly detailed description of the project deliverables.
- The first two levels of the WBS (the root node and Level 2) define a set of planned outcomes that collectively and exclusively represent 100% of the project scope. At each subsequent level, the children of a parent node collectively and exclusively represent 100% of the scope of their parent node.

SOFTWARE REQUIREMENTS:

- **VMware Tools**

VMware Tools is a set of services and modules that enable several features in VMware products for better management of, and seamless user interactions with, guest operating systems. It includes a number of feature enhancements, driver-related enhancements, and support for new guest operating systems.

VMware Workstation includes the ability to group multiple virtual machines in an inventory folder. The machines in such a folder can then be powered on and powered off as a single object, useful for testing complex client-server environments.

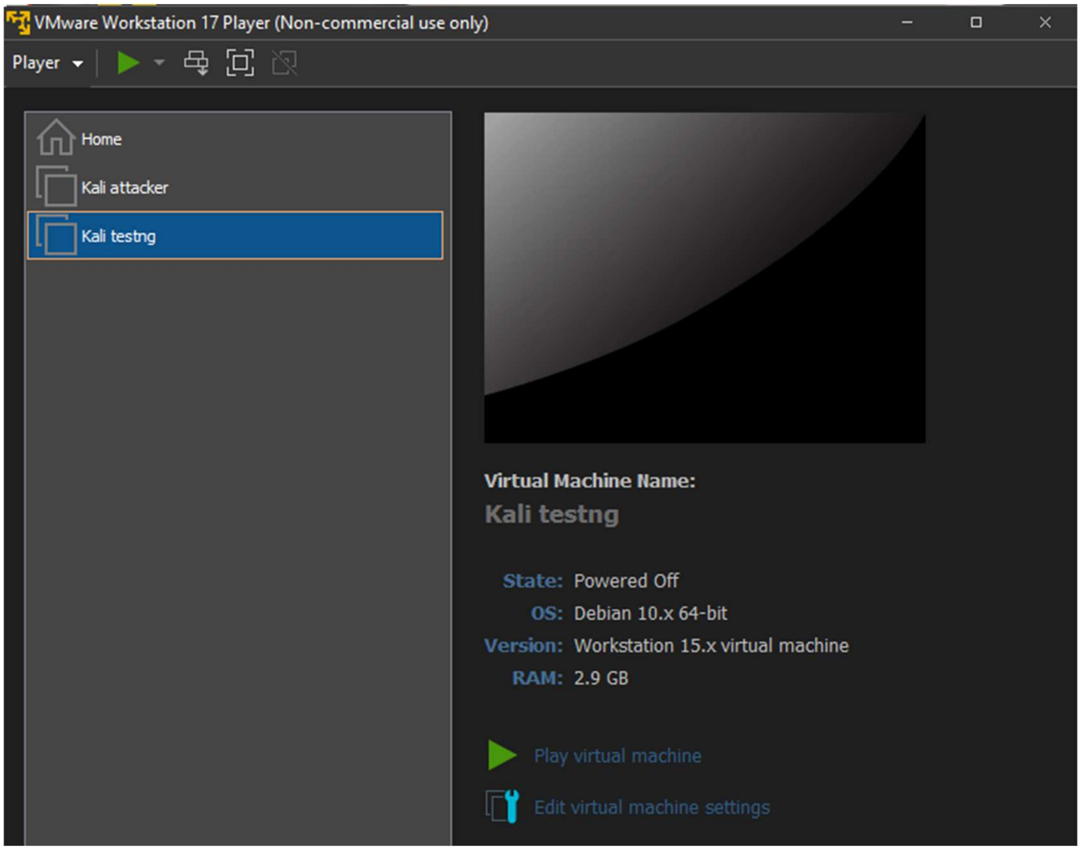
- **Kali Linux VMware:**

Installing “Guest Tools”, gives a better user experience with VMware VMs. This is why since Kali Linux 2022.1, during the setup process it should detect if Kali Linux is inside a VM. If it is, then automatically install any additional tools (in VMware case, open-vmtool).

Kali Linux is a Debian-derived Linux distribution designed for digital forensics and penetration testing. It is maintained and funded by Offensive Security.

It was developed by Mati Aharoni and Devon Kearns of Offensive Security through the rewrite of BackTrack, their previous information security testing Linux distribution based on Knoppix.

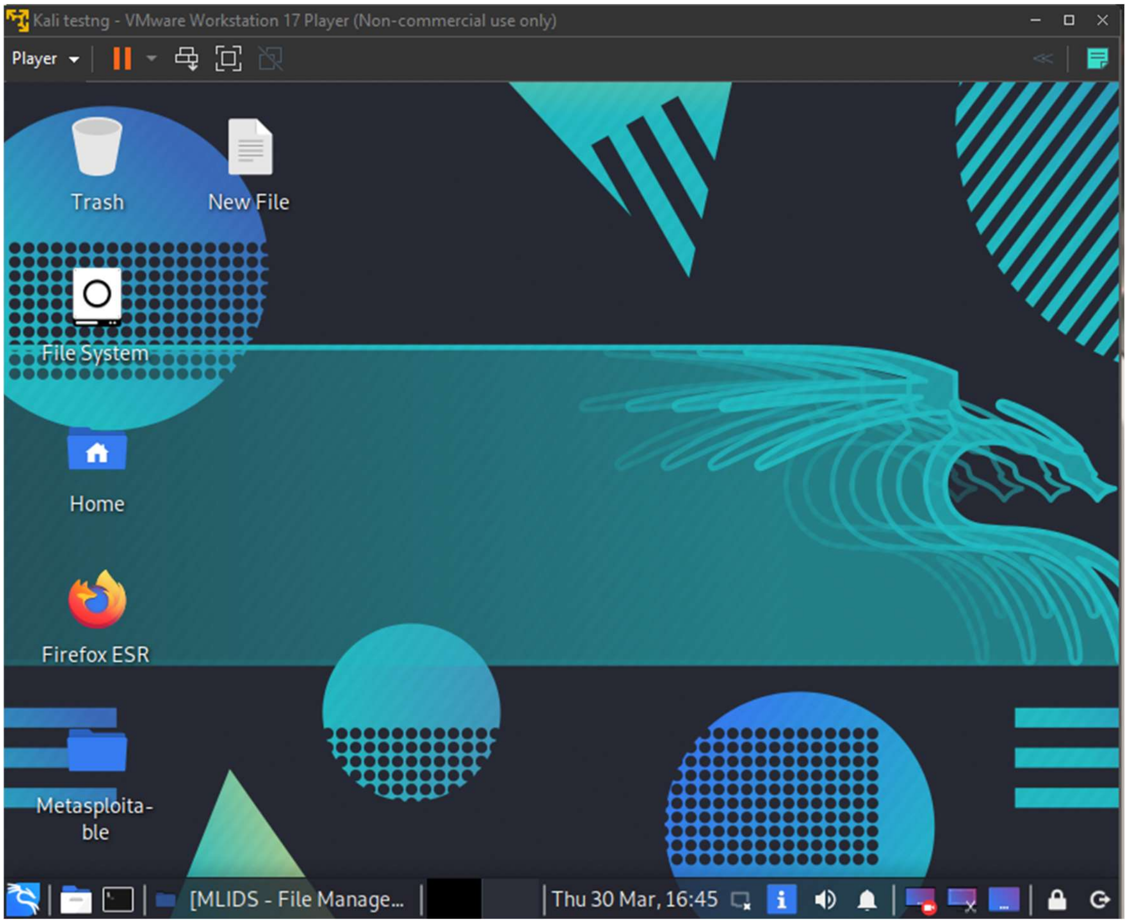
Kali Linux is based on the Debian *Testing* branch. Most packages Kali uses are imported from the Debian repositories.



VMware Workstation 17 Player (Non-commercial use only)
Player
Home
Kali attacker
Kali testng
Play virtual machine
Edit virtual machine settings

Virtual Machine Name:
Kali testng

State: Powered Off
OS: Debian 10.x 64-bit
Version: Workstation 15.x virtual machine
RAM: 2.9 GB



HARDWARE REQUIREMENTS:

- **Operating System (**Microsoft® Windows® 7/8/10 (64-bit)**):**

  For using an Android emulator, you'll most likely need the 64-bit version of Windows 10. Most android emulators require this as a minimum and don't work with 32-bit Windows.

- **Processor:**

  Most VMware require an Intel processor, but some also have support for AMD processors. The Intel processor should have support for Intel ® VT- x Intel ® EM64T(Intel ® 64),and execute Disable (XD) Bit functionality

- **Memory:**

  For using VMware, you'll need 8 GB RAM minimum, 16 GB RAM recommended. 25 GB of available disk space minimum.

- **Storage:**

  Every emulator has different storage requirements, but you should keep at least 25 GB in mind. You'll need more of it once you start using the emulator and download all your games. Games will also get stored on the hard drive, so make sure you have sufficient space

- **Screen Resolution**:

  Assuming you're downloading an Android emulator for playing games, you would want a high-definition screen resolution. However, for most emulators, the minimum screen resolution is 1280 x 900 pixels. This is one of those things that can enhance the quality of graphics.

- **Graphics Driver**:

  Make sure to update your graphics driver on Windows 10, as that's recommended for using VMware. An old graphics driver may not fully support or optimize graphics for Android games.

CHAPTER-4

SYSTEM DESIGN

4.1 Basic Modules

- **Python:**
  1. Pandas: The pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD.
  2. Scikit-learn: Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy

- **Tkinter:**
  The tkinter package (“Tk interface”) is the standard Python interface to the Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, as well as on Windows systems. (Tk itself is not part of Python; it is maintained at Active State.) Running python -m tkinter from the command line should open a window demonstrating a simple Tk interface, letting you know that tkinter is properly installed on your system, and also showing what version of Tcl/Tk is installed, so you can read the Tcl/Tk documentation specific to that version.
  1. Frames: he Frame widget is very important for the process of grouping and organizing other widgets in a somehow friendly way. It works like a container, which is responsible for arranging the

position of other widgets. It uses rectangular areas in the screen to organize the layout and to provide padding of these widgets. A frame can also be used as a foundation class to implement complex widgets

Syntax: w = Frame (master, option, ...)

2. Entry: The Entry widget is used to accept single-line text strings from a user. If you want to display multiple lines of text that can be edited, then you should use the Text widget. If you want to display one or more lines of text that cannot be modified by the user, then you should use the Label widget.

   Syntax: w = Entry (master, option, ...)

3. The Button widget is used to add buttons in a Python application. These buttons can display text or images that convey the purpose of the buttons. You can attach a function or a method to a button which is called automatically when you click the button.

   ```
   Syntax: w = Button ( master, option=value, ... )
   ```

- **VMware Tools**

VMware Tools is a set of services and modules that enable several features in VMware products for better management of, and seamless user interactions with, guest operating systems. It includes a number of feature enhancements, driver-related enhancements, and support for new guest operating systems.

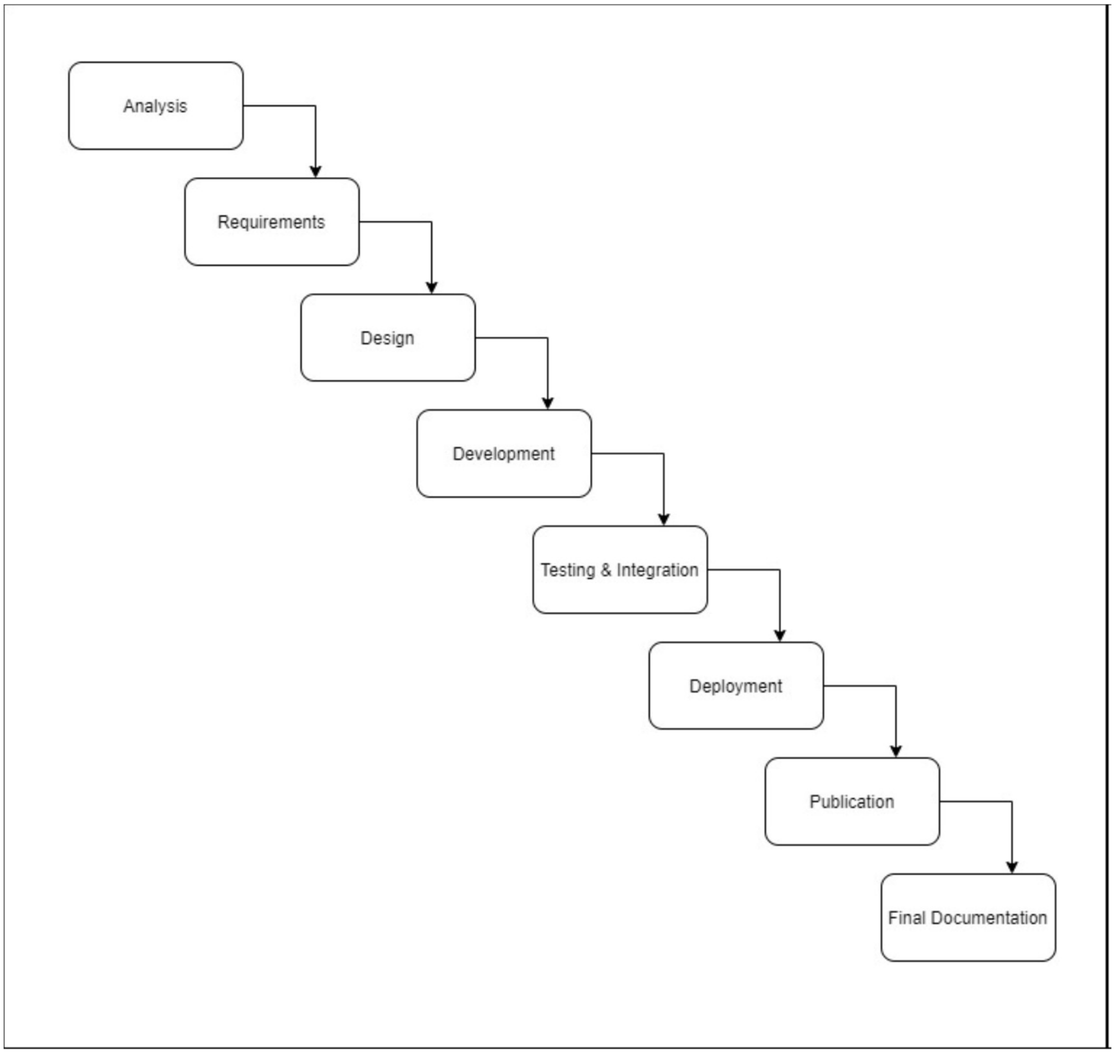
1. Installing “Guest Tools”, gives a better user experience with VMware VMs. This is why since Kali Linux 2022.1, during the setup process it should detect if Kali Linux is inside a VM. If it is, then automatically install any additional tools (in VMware case,

open-vmtool).

4.2 Process Model

Process Model Used for the Project: **Waterfall Model**

The cascading effect from one phase to the other as is illustrated in figure. In this model each phase well defined starting and ending point, with identifiable deliveries to the next phase. This model is sometimes referred to as the linear sequential model or the software life cycle.



4.3 Procedural Design

4.3.1 Use Case Diagram:

Description of Use Case Diagram

- A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.
- The UML diagram help us to understand how the software would be implemented and how it will be able to classify attacks and the product.
- In our case the UML diagram has 4 character namely user, sniffer and converter, ML model working inside system. GUI is there for user which help him see how ML classify the attack and normal packets.
- A Use Case diagram is a type of behavioural diagram defined by the UML Created From a use case analysis. Its purpose is to prevent a graphical overview of the functionality provided by a system in terms of actors, and their goal represented as use case.

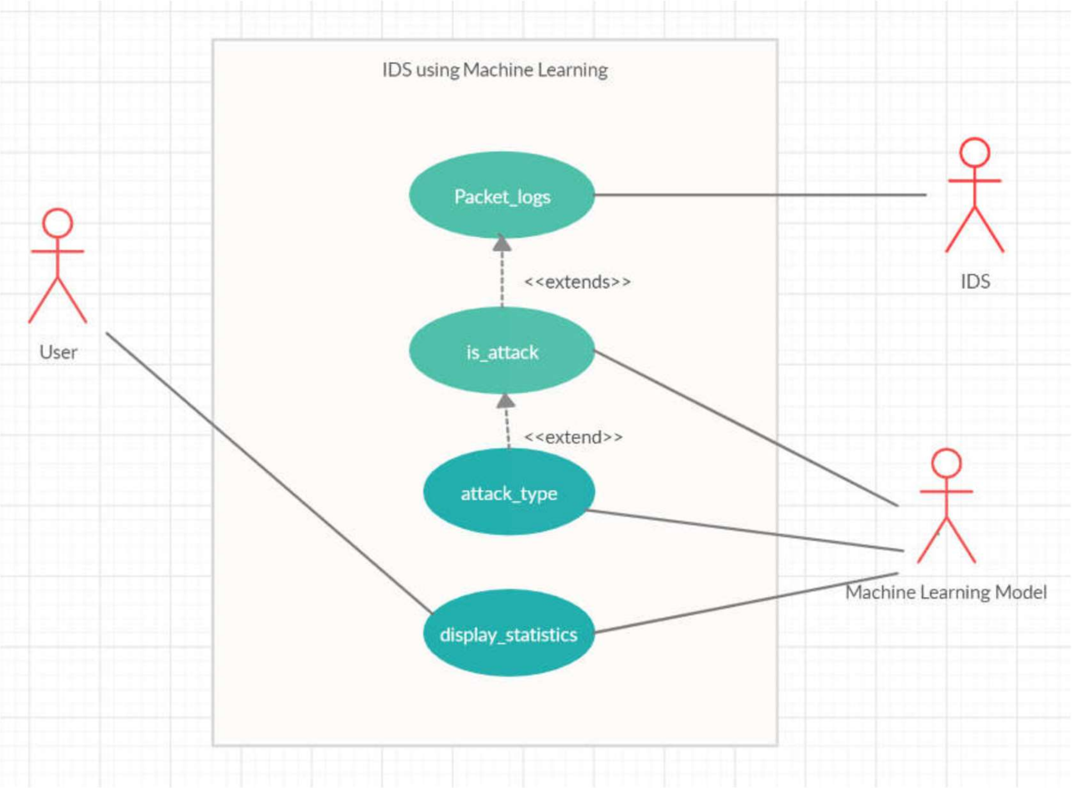
From a use case analysis. Its purpose is to prevent a graphical overview of the functionality provided by a system in terms of actors, and their goal represented as use case.

It is a type of diagram that shows a set of use cases, actors and their relationship. It should have a distinct name.

It commonly contains:

1. Use Cases
2. Actors (Primary and Secondary)

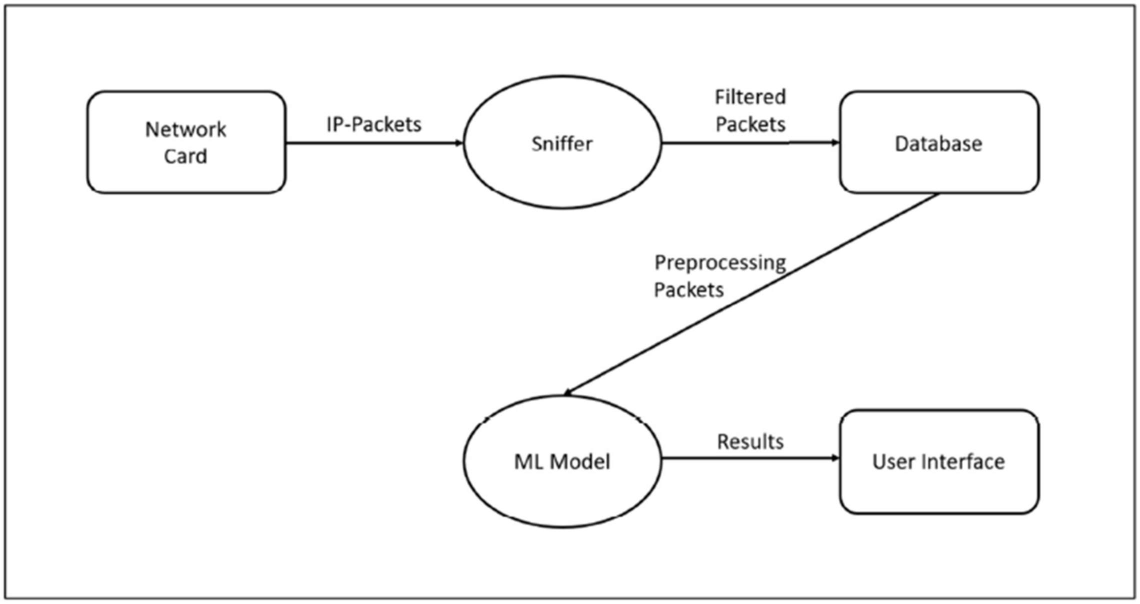
3. Dependency and Generalization



Use Case Diagram

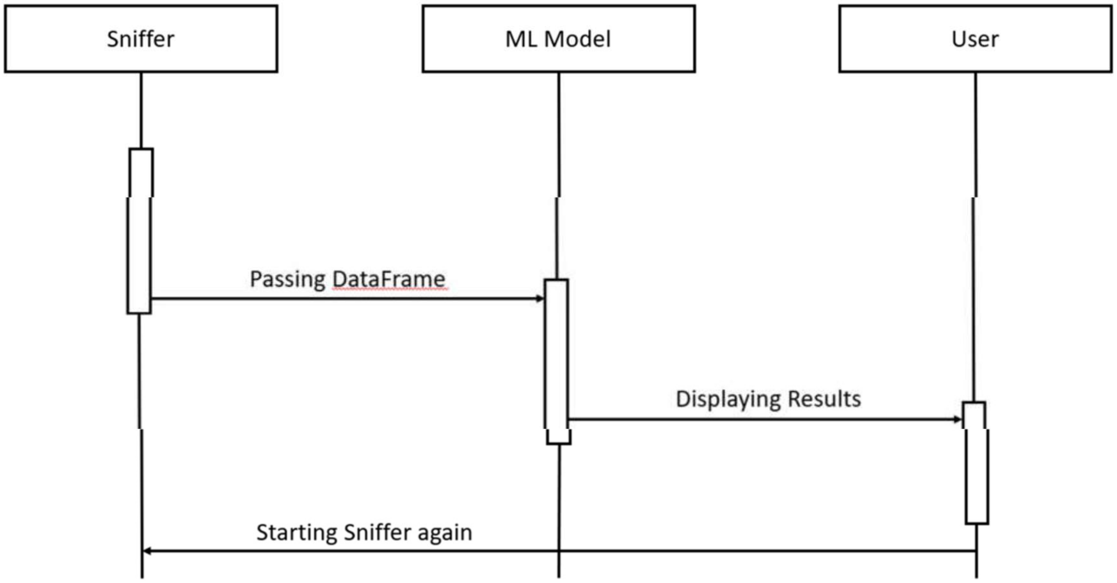
4.3.2 Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.



4.3.3 Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. Sequence diagram is an interaction diagram that emphasizes the time ordering of messages. A sequence diagram is a structured representation of behaviour as a series of sequential steps over time. It is used primarily to show the interactions between objects in the sequential order. The sequence diagram is also called as Message Sequence Chart.



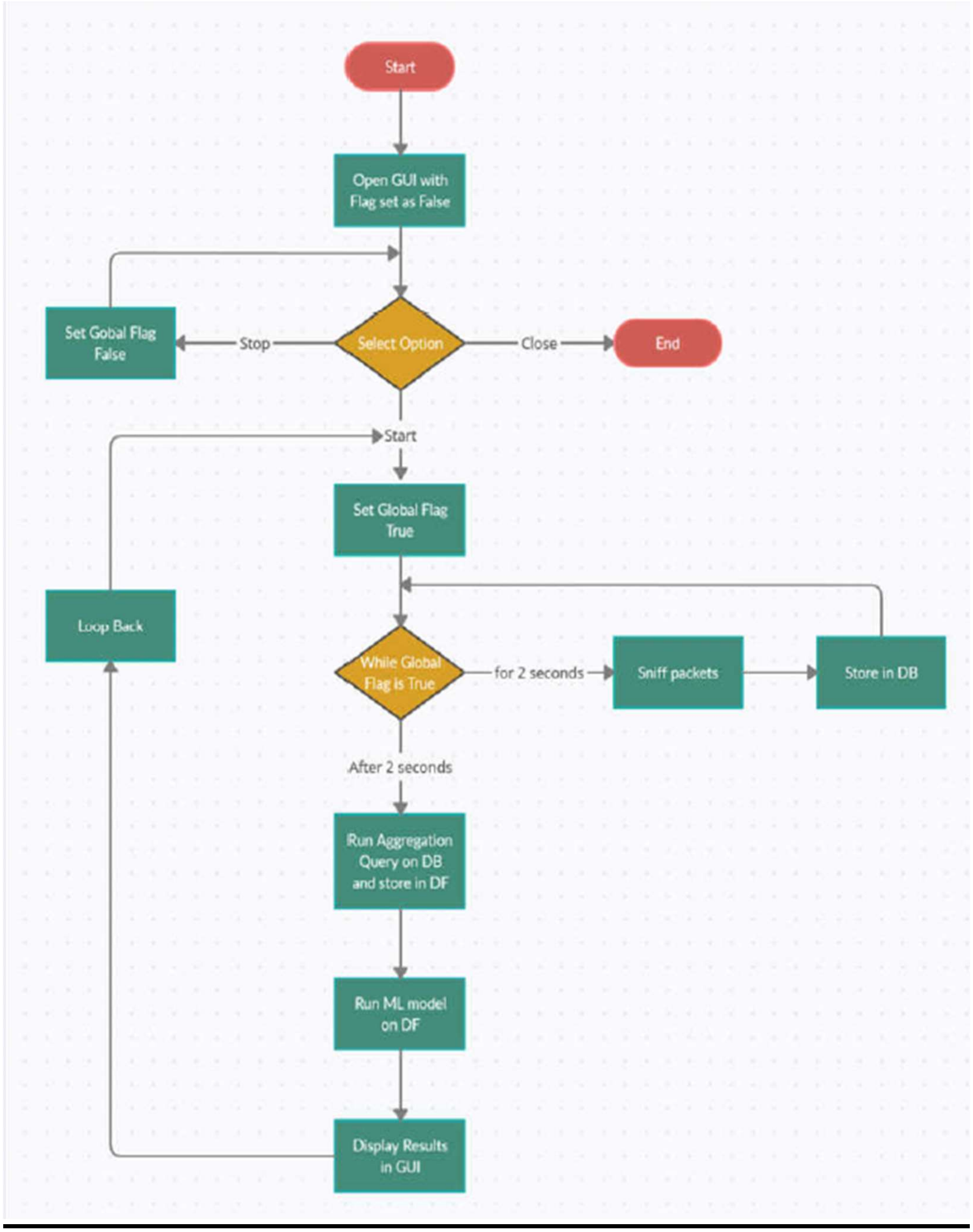
Sequence Diagram

4.3.4 Activity Diagram:

Description of Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise ac- tivities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows).

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational process.



Activity Diagram

4.4 User Interface Design

4.4.1 Results

Home Page

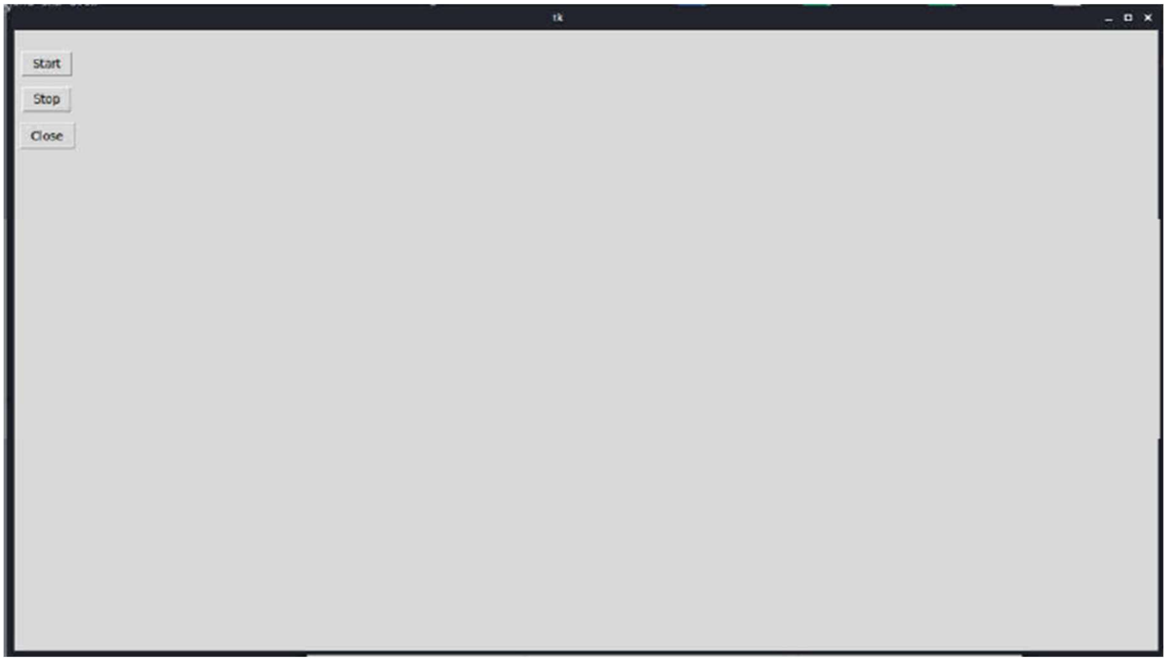
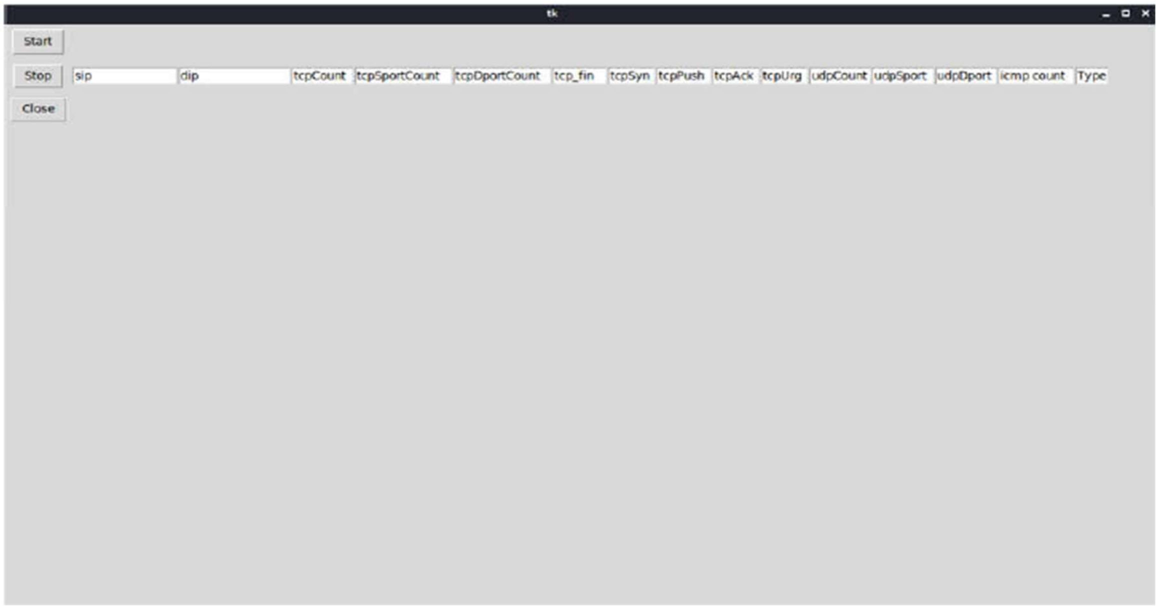
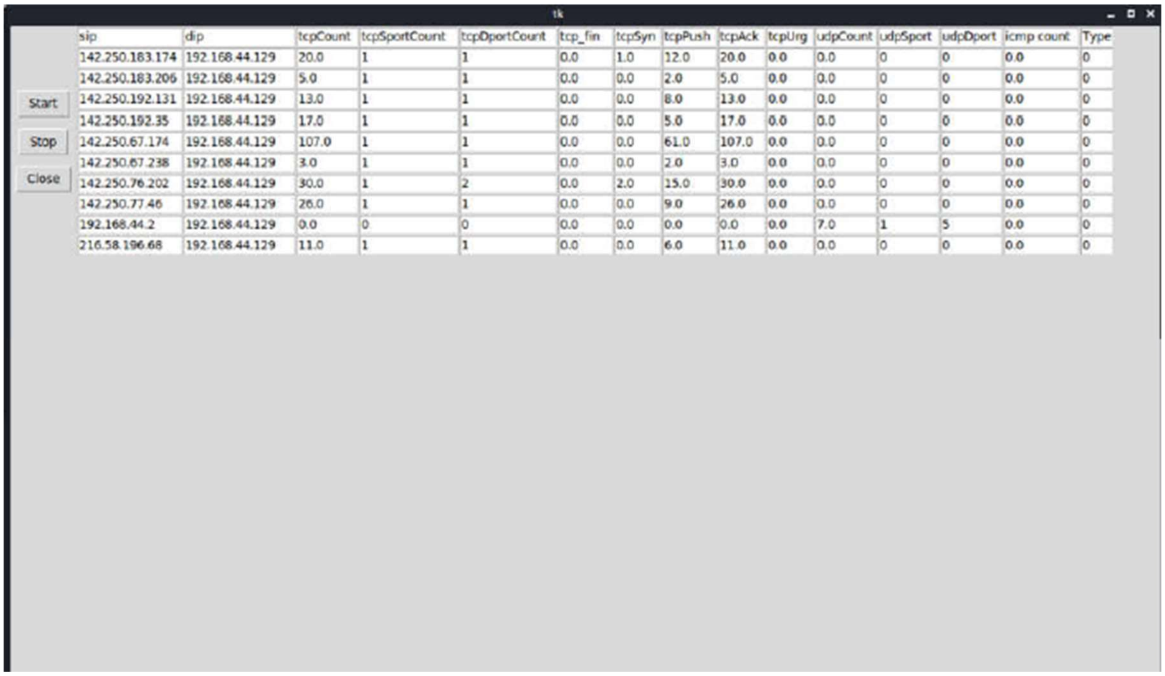


Table Creation



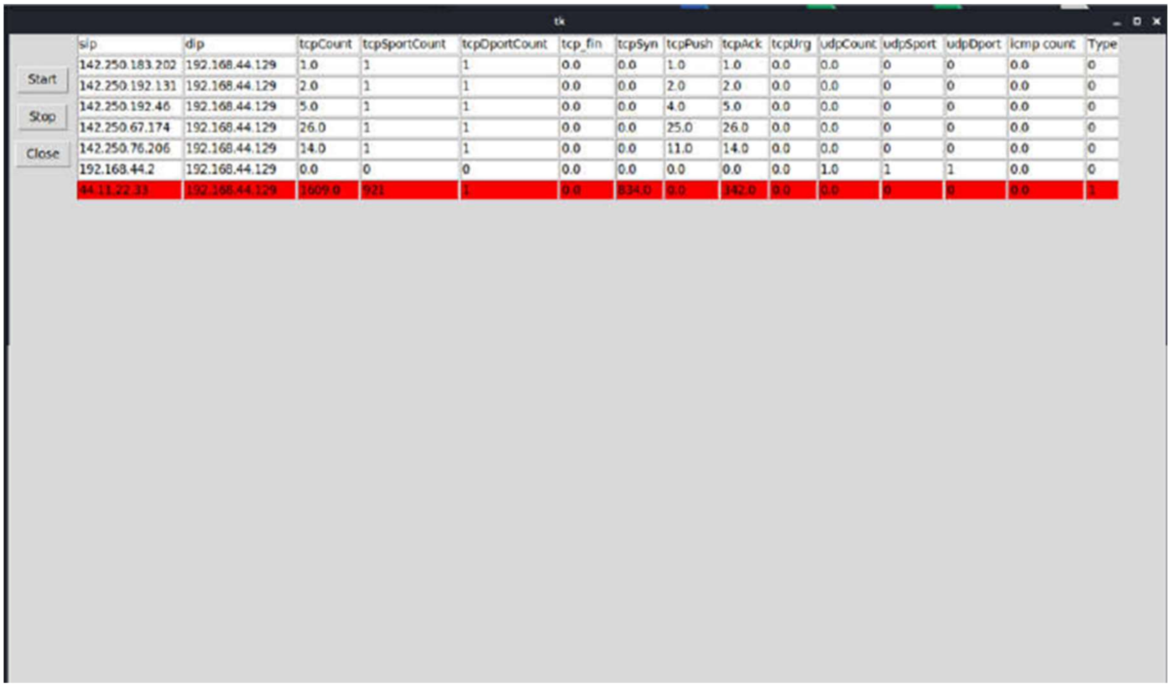
sip	dip	tcpCount	tcpSportCount	tcpDportCount	tcp_fin	tcpSyn	tcpPush	tcpAck	tcpUrg	udpCount	udpSport	udpDport	icmp count	Type

Connection Display



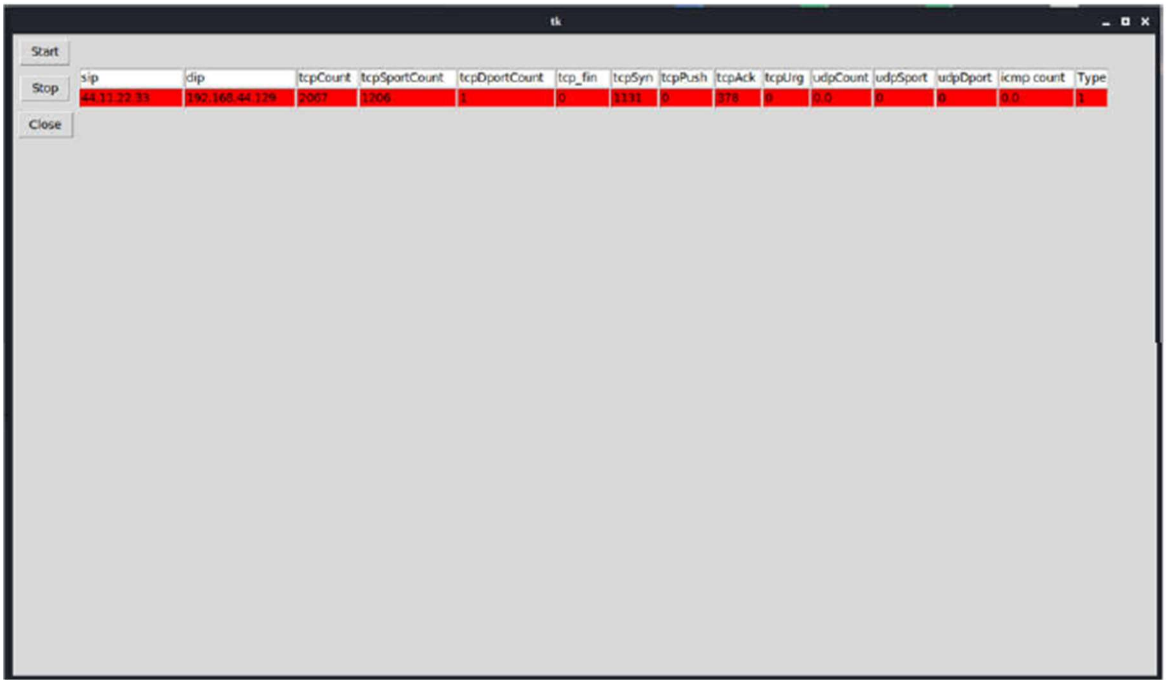
sip	dip	tcpCount	tcpSportCount	tcpDportCount	tcp_fin	tcpSyn	tcpPush	tcpAck	tcpUrg	udpCount	udpSport	udpDport	icmp count	Type
142.250.183.174	192.168.44.129	20.0	1	1	0.0	1.0	12.0	20.0	0.0	0.0	0	0	0.0	0
142.250.183.206	192.168.44.129	5.0	1	1	0.0	0.0	2.0	5.0	0.0	0.0	0	0	0.0	0
142.250.192.131	192.168.44.129	13.0	1	1	0.0	0.0	8.0	13.0	0.0	0.0	0	0	0.0	0
142.250.192.35	192.168.44.129	17.0	1	1	0.0	0.0	5.0	17.0	0.0	0.0	0	0	0.0	0
142.250.67.174	192.168.44.129	107.0	1	1	0.0	0.0	61.0	107.0	0.0	0.0	0	0	0.0	0
142.250.67.238	192.168.44.129	3.0	1	1	0.0	0.0	2.0	3.0	0.0	0.0	0	0	0.0	0
142.250.76.202	192.168.44.129	30.0	1	2	0.0	2.0	15.0	30.0	0.0	0.0	0	0	0.0	0
142.250.77.46	192.168.44.129	26.0	1	1	0.0	0.0	9.0	26.0	0.0	0.0	0	0	0.0	0
192.168.44.2	192.168.44.129	0.0	0	0	0.0	0.0	0.0	0.0	0.0	7.0	1	5	0.0	0
216.58.196.68	192.168.44.129	11.0	1	1	0.0	0.0	6.0	11.0	0.0	0.0	0	0	0.0	0

Prediction of Attack



sip	dip	tcpCount	tcpSportCount	tcpDportCount	tcp_fin	tcpSyn	tcpPush	tcpAck	tcpUrg	udpCount	udpSport	udpDport	icmp count	Type
142.250.183.202	192.168.44.129	1.0	1	1	0.0	0.0	1.0	1.0	0.0	0.0	0	0	0.0	0
142.250.192.131	192.168.44.129	2.0	1	1	0.0	0.0	2.0	2.0	0.0	0.0	0	0	0.0	0
142.250.192.46	192.168.44.129	5.0	1	1	0.0	0.0	4.0	5.0	0.0	0.0	0	0	0.0	0
142.250.67.174	192.168.44.129	26.0	1	1	0.0	0.0	25.0	26.0	0.0	0.0	0	0	0.0	0
142.250.76.206	192.168.44.129	14.0	1	1	0.0	0.0	11.0	14.0	0.0	0.0	0	0	0.0	0
192.168.44.2	192.168.44.129	0.0	0	0	0.0	0.0	0.0	0.0	0.0	1.0	1	1	0.0	0
44.11.22.33	192.168.44.129	1609.0	921	1	0.0	834.0	0.0	342.0	0.0	0.0	0	0	0.0	1

Attack Packets Display



sip	dip	tcpCount	tcpSportCount	tcpDportCount	tcp_fin	tcpSyn	tcpPush	tcpAck	tcpUrg	udpCount	udpSport	udpDport	icmp count	Type
44.11.22.33	192.168.44.129	2067	1206	1	0	1131	0	378	0	0.0	0	0	0.0	1

4.5 Test Cases

Test Case No.	**Test Case Description**	**Expected Result**	**Result**
1.	Sniffer | Reading TCP Packets	Expected TCP packet header .	PASS
2.	Sniffer | Reading UDP Packets	Expected UDP packet header	PASS
3	Sniffer | Reading ICMP Packets	Expected ICMP packet header	PASS
4	DB | Connection test	A Cursor Variable to store Unorganized data into the data base	PASS
5	DB | Data Storage	Query to Store data according to TCP, UDP, ICMP header values	PASS
6	Aggregate Data | Data Modifying Aggregating and fetching query	All TCP, UDP, ICMP packet converted into one dataFrame	PASS
7	Two Second Time Loop | Add timer function to the Loop	Import time and add time variable and integrate in while loop	PASS

8	ML | Known attack	Test model on Known attack tool	Pass
9	ML | UnKnown attack	Test model on UnKnown attack tool	PASS
10	GUI | Start Button	Change global flag to True	PASS
11	GUI | Stop Button	Change global flag to False	PASS
12	GUI | Close Button	Destroy root process	PASS
13	Table | Table without Data	Print Table Heads	Pass
14	Table | Table with Data	Add data after prediction to the table	PASS
15	GUI | Check Flag	Checks Flag after every 3 seconds	PASS

CHAPTER-5

IMPLEMENTATION AND TESTING

5.1 IMPLEMENTATION

After you have carefully planned your project, you will be ready to start the project implementation phase, the third phase of the project management life cycle. The implementation phase involves putting the project plan into action. It’s here that the project manager will coordinate and direct project resources to meet the objectives of the project plan. As the project unfolds, it’s the project manager’s job to direct and manage each activity, every step of the way. That’s what happens in the implementation phase of the project life cycle: you follow the plan you’ve put together and handle any problems that come up.

The implementation phase is where you and your project team do the project work to produce the deliverables. The word “deliverable” means anything your project delivers. The deliverables for your project include all the products or services that you and your team are performing for the client, customer, or sponsor, including all the project management documents that you put together.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the software development methodology adopted.

5.2 CODING DETAILS AND CODE DETAILS

MLIDS.py

```
import sniffer
import modifier
import pickle
import sklearn
import pandas as pd
from sklearn import svm
from sklearn.model_selection import train_test_split
import numpy as np
from sklearn import metrics
from tkinter import *
# ML Model training code
'''
df = pd.read_csv(r'allAttack0.csv',header=None)
df[0:5]
df[14]=df[13]
df[14]=1
#df[14].unique()

df1 = pd.read_csv(r'normal3.csv',header=None)
df1[0:5]
df1[14]=df1[13]
df1[14]=0
#df1[14].unique()



frames=[df,df1]
dataset=pd.concat(frames)

dataset.replace('?',-9999,inplace=True)
X=dataset.drop([0,1,14],axis="columns")
y=np.array(dataset[14])

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25)
clf=svm.SVC()
clf.fit(X_train,y_train)
```

```
y_pred=clf.predict(X_test)

print("Accuracy",metrics.accuracy_score(y_test,y_pred))
#print(results)

#Saving model
filename="microdatasetmodel.sav"
pickle.dump(clf,open(filename,'wb'))
'''

# list of table heads
lst = ['sip            ', 'dip            ', 'tcpCount',
'tcpSportCount', 'tcpDportCount', 'tcp_fin', 'tcpSyn', 'tcpPush',
'tcpAck',
        'tcpUrg', 'udpCount', 'udpSport', 'udpDport', 'icmp count',
'Type']

# Global Status Flag
flag = 0

# Globally Loading Machine Learning Model
filename = "microdatasetmodel.sav"
load_picklemodel = pickle.load(open(filename, 'rb'))

# Tkinter Start Button code
def start():
    print("Giving Flag to IDS")
    global flag
    flag = True


# Tkinter recursive loop code
def StartIDS():

    #Delete previous Entries before new loop
    print(len(tableFrame.winfo_children()))
    if len(tableFrame.winfo_children()):
        for i in tableFrame.winfo_children():
            i.destroy()
    #Displaying Table Heads
    for heads in range(len(lst)):
        e = Entry(tableFrame,width=len(lst[heads]))
        e.grid(row=0, column=heads, padx=0, pady=0)
        e.insert(END,lst[heads])

    # Run when Start is pressed i.e flag is True
    if flag:
        # Starting sniffer
        df = sniffer.sniff()
        # modifier.modify()

        # Pickle programn
        if len(df) > 0:
            df = df.fillna(value=np.nan)
            df = df.replace(np.nan, int(0))
            temp = df
            temp = temp.drop(['sip', 'dip'], axis=1)
            result = load_picklemodel.predict(temp)

            # output dataframe for GUI
            df[14] = result  # appending ML results to dataframe

            # table creation
            # total number of rows and columns in list
            total_columns = 15
            total_rows = len(df.index)
            print(total_rows)
            for i in range(total_rows):
                if df.iat[i,14]==1:
                    for j in range(total_columns):
                        e =
Entry(tableFrame,bg="Red",width=len(lst[j]))
                        e.grid(row=i + 2, column=j, padx=0, pady=0)
                        e.insert(END, df.iat[i, j])
                else:
                    for j in range(total_columns):
                        e = Entry(tableFrame, width=len(lst[j]))
                        e.grid(row=i + 2, column=j, padx=0, pady=0)
                        e.insert(END, df.iat[i, j])
    root.after(3000,StartIDS)

# Tkinter Stop Button Code
def stop():
    print("Stopping IDS")
    global flag
    flag = 0

# Close button program
def close():
    print("Closing program")
    root.destroy()


root = Tk()
root.geometry("1280x720")
root.title("MLIDS")

controlFrame = Frame(root)
controlFrame.grid(row=0,column=0, padx=5, pady=0)

startButton = Button(controlFrame, text="Start", command=start)
startButton.grid(row=0, column=0, padx=0, pady=5)

stopButton = Button(controlFrame, text="Stop", command=stop)
stopButton.grid(row=1, column=0, padx=0, pady=5)

closeButton = Button(controlFrame, text="Close", command=close)
closeButton.grid(row=2, column=0, padx=0, pady=5)

tableFrame = Frame(root)
tableFrame.grid(row=0,column=1)

root.after(3000,StartIDS)

root.mainloop()
```

Sniffer.py

```
import time
import sqlite3
import socket
import csv
from general import *
from networking.ethernet import Ethernet
from networking.ipv4 import IPv4
from networking.icmp import ICMP
from networking.tcp import TCP
from networking.udp import UDP
from networking.pcap import Pcap
from networking.http import HTTP
import pandas as pd


db = sqlite3.connect('packet.db')
c = db.cursor()


def main():
    pcap = Pcap('capture.pcap')
    conn = socket.socket(socket.AF_PACKET, socket.SOCK_RAW,
socket.ntohs(3))


    t_end = time.time() + 2
    c.execute('''CREATE TABLE IF NOT EXISTS packets
(sip,dip,tcp,tcp_sport,tcp_dport,tcp_fin,tcp_syn,tcp_push,tcp_ack,tcp
_urg,udp,udp_sport,udp_dport,icmp)''')
    while time.time() < t_end:
        raw_data, addr = conn.recvfrom(65535)
        pcap.write(raw_data)
        eth = Ethernet(raw_data)
        if eth.proto == 8:
            ipv4 = IPv4(eth.data)

            if ipv4.target == "192.168.91.132":

                if ipv4.proto == 1:
                    icmp = ICMP(ipv4.data)
                    c.execute('''INSERT INTO packets (sip,dip,icmp)
VALUES (?,?,?)''',(ipv4.src,ipv4.target,1))
                    db.commit()

                # TCP
                elif ipv4.proto == 6:
                    tcp = TCP(ipv4.data)
                    c.execute('''INSERT INTO packets
(sip,dip,tcp,tcp_sport,tcp_dport,tcp_fin,tcp_syn,tcp_push,tcp_ack,tcp
_urg) VALUES
(?,?,?,?,?,?,?,?,?,?)''',(ipv4.src,ipv4.target,1,tcp.src_port,tcp.des
t_port,tcp.flag_fin,tcp.flag_syn,tcp.flag_psh,tcp.flag_ack,tcp.flag_u
rg))
                    db.commit()

                # UDP
                elif ipv4.proto == 17:
                    udp = UDP(ipv4.data)
                    c.execute('''INSERT INTO packets
(sip,dip,udp,udp_sport,udp_dport) VALUES
(?,?,?,?,?)''',(ipv4.src,ipv4.target,1,udp.src_port, udp.dest_port))
                    db.commit()

    pcap.close()

def sniff():
    main()
    print('Staring sniffer')
    """
    c.execute('SELECT sip,dip,sum(tcp),count(DISTINCT
tcp_sport),count(DISTINCT tcp_dport), sum(tcp_fin), sum(tcp_syn),
sum(tcp_push), sum(tcp_ack), sum(tcp_urg), sum(udp), count(DISTINCT
udp_sport), count(DISTINCT udp_dport), sum(icmp) FROM packets GROUP
BY sip, dip;')
    r = c.fetchall()
    for i in r:
        j = list(i)
        cfile = "current.csv"
        with open(cfile, 'a+', newline='') as file:
            writer = csv.writer(file)
            writer.writerow(j)
    print("completed csv")
    """

    query=pd.read_sql_query('''SELECT sip,dip,sum(tcp),count(DISTINCT
tcp_sport),count(DISTINCT tcp_dport), sum(tcp_fin), sum(tcp_syn),
sum(tcp_push), sum(tcp_ack), sum(tcp_urg), sum(udp), count(DISTINCT
udp_sport), count(DISTINCT udp_dport), sum(icmp) FROM packets GROUP
BY sip, dip;''',db)

    df=pd.DataFrame(query)
    #print(df)

    c.execute('DROP TABLE packets')
    db.commit()

    return df

#sniff()
```

5.3 TESTING

Black box Testing:

Black box is a method of software testing that examines the functionality of the application without peering into its internal structure or working. This method of testing can be applied virtually to every level of testing. This test can be functional or non-functional. A black box tester is unaware of the internal structure of the application.

Unit Testing:

It is a software testing method by which a person unit of source code sets of one or more computer programs modules together with associated control data usage procedures and operating procedures are tested to determine whether they are fit for use.

Integration Testing:

Integration testing is a systematic technique for constructing the program structure while at the same time to uncover the errors associated with interfacing. The objective is to take unit- tested module and build a program structure that has been detected by designing. It also tests to find the discrepancies between the system and its original objectives. Subordinate stubs are replaced one at time actual module. Tests were conducted at each module was integrated. On completion of each set another stub was replaced with the real module.

Grey box Testing:

It is a combination of black box testing and white box testing. The aim of this testing is to search for the defects if any due to improper structure or improper usage of application. A grey box tester partially knows the internal structure which includes access to the documentation of internal data structure as well as algorithm used. Grey box tester requires both high level and detailed documents describing the application.

Black-Box testing helps to find errors such as-

- Incorrect or missing functions
- Interface errors
- Errors in data structures

Functional Testing:

Functional testing is a technique in which all the functionalities of the program are tested to check whether all the functions that where proposed during the planning phase are full filled. This is also to check that if all the functions proposed are working properly.

This is further done in two phases:

- One before the integration to see if all the unit components work properly
- Second to see if they still work properly after they have been integrated to check some functional compatibility issues arise.

Regression Testing:

It is a type of software testing that ensures that previously developed and tested software still performs the same way after it is changed or interfaced with other software. During regression testing, new software bugs or regression may be uncovered. The purpose of regression testing is to ensure that changes have not introduced new faults. The main reason of this testing is to check that change in one part of the software affects the other part or not.

CHAPTER-6

CONCLUSION

6.1 Conclusion

We presented a practical and efficient real-time network intrusion detection system (RT-IDS) model which can be used with existing well-known machine learning algorithms. Our RT-IDS model consists of three phases: the pre-processing phase, the classification phase, and the post-processing phase. We also presented how we preprocess the network packet header data into records of 12 essential features. We present an uncomplicated IDS model which can be easily applied with existing machine learning technique. We propose 12 essential features which are relevant to DoS. This small number of features can significantly improve the on-line (real-time) IDS detection speed and consumption of computer resources. We present a practical real time, network-based IDS that not only can efficiently detect but also can classify network data into two categories which are normal and Denial of Service (DoS)

6.2 Future Scope of the Project

This project can be really beneficial to the IT industry but it needs to be upgrade more with large number of datasets. Larger the data then more accurate will be the model.

REFERENCES

- **https://stackoverflow.com/**
- **https://www.geeksforgeeks.org/**
- **https://www.wikipedia.org/**
- **https://www.youtube.com/**
- **https://www.flaticon.com/**

Documents referred:

Department of Computer Engineering, Faculty of Engineering, King Mongkut's University of Technology Thonburi, 126 Pracha-utid Road, Toongkru, Bangkok 10140, Thailand

National Electronics and Computer Technology Center, 112 Phahonyothin Road, Klong Luang, Pathumthani 12120, Thailand