**Chapter 2 PROJECT PLANNING AND MANAGEMENT**

* 1. **Detail System Requirement Specification**
     1. **System Overview Product Perspective**

The system has been inspired by the already existing intrusion detection systems which attempt to protect the home-network against attacks and intrusions. We are taking these pre-existing models and combining them to get additional advantages and reduce the downsides as much as possible. Till date, there have been very few attempts at making an IDS for IOT devices. Most of the IDS present in the market cater to non-IDS networks. We are taking the principles of these IDS systems and modifying them as per requirement of an IOT network.

**Product Functions**

1. User authentication
2. Data connection establishment
3. Intrusion detection analysis
4. Attack Notification
5. Dashboard
6. Network log analysis
7. Visualizations

**User Classes and Characteristics**

Factories and companies employing a network of IOT devices. Usually the IOT net- works for these users are larger and require higher level security systems as economic factors are involved. Residential habitants employing network of IOT devices. In- cludes home-IOT networks (smart homes), or residential society IOT networks like CCTVs, smoke detectors, fire alarms etc. These networks are usually smaller than the previous.

**Operating Environment**

The system will be implemented on Linux. Python is used for programming the backend using Django Framework. Languages such as Python and Sqlite and platform like apache server are used for database and connection. Libraries like sklearn, pandas etc. will be used. For real life implementation, IOT data will be connected from Tshark.

**Design and Implementation Constraints**

* The user will be notified through a web app, which should be installed on his system.
* Node of the IOT network should be connected to the apache server where the processing is to take place.
* The device to be notified also has to be part of the given network.
* Processing time should be as low as possible.
* Backend should be connected to database.
* Wifi supported on the IOT device.

**User Documentation**

* User manual.

**Assumptions and Dependencies**

* We are assuming that the machine has the required resources (memory and pro- cessing power etc.) and capabilities to run the system.
* The user has updated system.
* We are assuming that the system has the required packages and

dependencies(such as Django and apache server) to run the system.

**2.1.2 Functional Requirements(System Features)**

**Sensing of the data using ultrasonic sensor**

**Description and Priority**

First, sensors or devices collect data from their environment. The benefit of this feature is much larger and the cost of this feature is reasonably low. Only risk of this feature is that it sometimes may not work as intended.

**Stimulus/Response Sequences**

Once the IOT device is started, distance data is processed and sent from the node to the apache server. The network traffic is monitored using Tshark and dumped into a csv file. The python script in the backend reads this csv file, preprocesses it and feeds it to the ML model. If an intrusion is detected, it is notified to the user on the front end.

**Functional Requirements**

IOT sensor node equipped with Arduino and nodemcu.

**Connectivity Description and Priority**

The sensors/devices is be connected to the network through Zigbee protocol. This feature is of high priority because the sensed data needs to be processed further. The data will be processed only if it gets proper way to reach to the server. Only risk of this feature is that it can sometimes not send the data to the server.

**Stimulus/Response Sequences**

Data sensed is sent to the server.

**Functional Requirements**

Apache server of the network along with wifi module.

**Data Processing Description and Priority**

Once the data gets to the server, software performs processing on it. Here IDS software is used to determine the category of the attack. This feature is of high priority because this is the main objective behind selecting this project. Only risk of this feature is that it can sometimes not categorize the attack perfectly.

**Stimulus/Response Sequences**

In response, it will determine whether the attack was normal or malicious.

**Functional Requirements**

IDS software.

**User interface Description and Priority**

Next, the information is made useful to the end-user in some way. This is an alert to the user (text, notification, etc.). This feature is of medium priority. Only risk of this feature is that it can sometimes not work as intended.

**Stimulus/Response Sequences**

In response it will give notification of the attack on the GUI.

**Functional Requirements**

GUI

* + 1. **Non- Functional Requirements Performance Requirements**

The data transfer from the IOT node to the apache server should take as little time as possible for proper real time functioning of the system. Also, the processing of the

data should be fast enough so as to give proper result as soon as the attack happens,

i.e. the time between the occurrence and detection of the attack should be minimum.

**Safety Requirements**

The backend could crash resulting in failure of the whole system. If the system takes up all the processing power, that machine could crash. Regular maintenance of the networking components, such as checking the proper functioning of the Arduino ,ultrasonic sensor and nodemcu should be done.

**Security Requirements**

User need to signup first and then login to get access to the system, in this case users data should be protected. Systems should be secure against unauthorized access to any of their data, unauthorized use of them or any of their components. Details regarding IOT devices and their data should also be protected.

**Software Quality Attributes**

Our software has many quality attribute that are given below: -

* Availability: This software is freely available to all users. The availability of the software is easy for everyone.
* Maintainability: After the deployment of the project if any error occurs then it can be easily maintained by the software developer.
* Reliability: The performance of the software is better which will increase the reliability of the Software.
* User Friendly: Since, the software is a GUI application, the output generated is much user friendly in its behavior.
* Integrity: Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.
* Security: Users are authenticated using many security phases so reliable security is provided.

data should be fast enough so as to give proper result as soon as the attack happens,

i.e. the time between the occurrence and detection of the attack should be minimum.

**Safety Requirements**

The backend could crash resulting in failure of the whole system. If the system takes up all the processing power, that machine could crash. Regular maintenance of the networking components, such as checking the proper functioning of the Arduino ,ultrasonic sensor and nodemcu should be done.

**Security Requirements**

User need to signup first and then login to get access to the system, in this case users data should be protected. Systems should be secure against unauthorized access to any of their data, unauthorized use of them or any of their components. Details regarding IOT devices and their data should also be protected.

**Software Quality Attributes**

Our software has many quality attribute that are given below: -

* Availability: This software is freely available to all users. The availability of the software is easy for everyone.
* Maintainability: After the deployment of the project if any error occurs then it can be easily maintained by the software developer.
* Reliability: The performance of the software is better which will increase the reliability of the Software.
* User Friendly: Since, the software is a GUI application, the output generated is much user friendly in its behavior.
* Integrity: Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.
* Security: Users are authenticated using many security phases so reliable security is provided.
  + 1. **Deployment Environment**

**Software Requirements-**

1. Python 3 –Coding language used in our project
2. Django Framework-Used for backend work
3. Node- Used for executing javascript code outside of a web browser
4. Npm-Used as package manager for the Node JavaScript platform
5. React js-Used for frontend work
6. Sqlite-Used for database
7. Xampp server-Used for storing network data
8. Wireshark-Used for network data analysis

**Hardware Requirements-**

1. Arduino-For reading inputs of ultrasonic sensor and turn it into an output as distance data
2. Nodemcu-Node Microcontroller Unit used to build IOT network
3. Ultrasonic sensor-Used for collecting sensor data
4. Connecting wires-Used for connecting nodemcu to Arduino and ultrasonic sensor
   * 1. **External Interface Requirements**

## User Interfaces

1)User Signup page

* Username
* Email id
* Password
* Signup(Button)

2)User login page

* Username
* Password
* Login(Button)

3)User Notification page

Notifies when the network is under attack or some anomalous activity is detected.

3)User dashboard page

Displays ultrasonic data from php script in animated format on web page.

3)User Network Logs page

Displays the real time Tshark data in table format.

4)User Visualisations page

Displays graphs and confusion matrix for various types of data.

**Hardware Interfaces**

Hardware interfaces for a web application will run on Linux,Windows or Mac. For connection of IOT node to the apache server will require a network. For a larger model, the network and servers will increase. Here IOT network consists of ultrasonic sensor, Arduino and Nodemcu which sends data to the apache server using wifi module.

**Software Interfaces**

1. Operating system-We have chosen Linux operating system because of its user friendly features, benefits of security and control without complexity and unrealistic costs.
2. Database-Sqlite using apache server
3. Python-Django is used to program and implement the backend and React js used to create web pages for frontend.IDS is designed to run in the background of an IoT network, where distance data is processed and sent from the node to the apache server. The network traffic is monitored using Tshark and dumped into a csv file. The python script in the backend reads this csv file, preprocesses it and feeds it to the ML model. If an intrusion is detected, it is notified to the user on the front end.

**Communications Interfaces**

1)Web application runs on a web browser 2)HTTP is used for notification system

3)Wifi module is used for data transfer from IOT node to the apache server.

* + 1. **Other Requirements**

1. Configuration- -Systems should be configurable for detection and reaction, as fea- sible for alerts, updates, protocol and port coverage, and detection-threshold levels

-Systems should be configurable treat identified IP or other network addresses excep- tionally; for example, configurable to never block or shun network activity from one or more IP addresses.

1. System Updates-These capabilities should be engineered in such a manner as to allow updates during operational use of the products without disruption.
2. Ease of use-Operator console should not require undue expertise to operate; experts may reside external to operator staff.

## Project Process Modeling

Iterative waterfall model is the best suited for this project. Iterative waterfall model can be thought of as incorporating the necessary changes to the classical waterfall model to make it usable in practical software development projects. It is almost same as the classical waterfall model except some changes are made to increase the efficiency of the software development.

The iterative waterfall model provides feedback paths from every phase to its preceding phases, which is the main difference from the classical waterfall model.In our project, every phase is well defined and to be executed one after the other sequentially, like the waterfall model. But if need be, there is space for going back to previous stages making changes. Hence, iterative waterfall is to be used for this project.

## 2.3 Cost Efforts Estimates

As per the basic COCOMO cost estimation formula projected cost for our product, Development Effort = a1 x (KLOC)a2PM

= 2.4 \* (4)1.05 = 10 Person Months(approx)

Nominal development time=b1 x (Effort)b2 Months

= 2.5 \* (10)0.38 = 6 months(approx)

Cost required to develop the product = NDT x Average salary per month x mem- bers

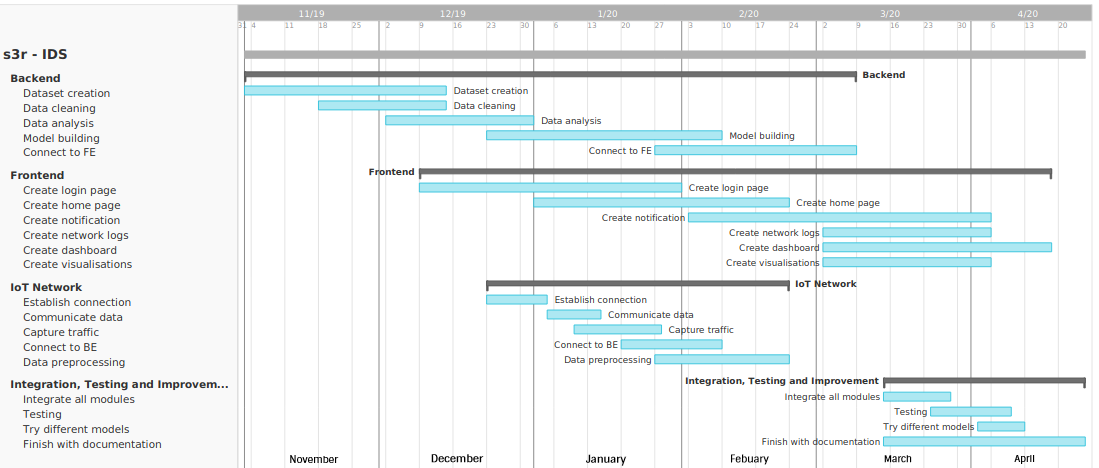
=6 \* 5000 \* 4

=Rs 120,000/-

Where, KLOC is the estimated size of the software product expressed in Kilo Lines of Code

a1, a2, b1, b2 are constants for each category of software products

* 1. **Project Scheduling Time Line Chart**

****

**Figure 2.1:** Time Line Chart