Malicious Node Detection for various Heterogenous IoT Communication Protocols

by

Somya
201915004

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
ELECTRONICS AND COMMUNICATION

with specialization in
Wireless Communication and Embedded Systems
to

Dhirubhai Ambani Institute of Information and Communication Technology

A program jointly offered with
C.R.Rao Advanced Institute of Mathematics, Statistics and Computer Science

May, 2021

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Master of Technology in Electronics and Communications at Dhirubhai Ambani Institute of Information and Communication Technology & C.R.Rao Advanced Institute of Applied Mathematics, Statistics and Computer Science, and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

Somya

Somya

Certificate

This is to certify that the thesis work entitled Malicious Node Detection in Internet of Things has been carried out by Somya for the degree of Master of Technology in Electronics and Communications at *Dhirubhai Ambani Institute of Information and Communication Technology & C.R.Rao Advanced Institute of Applied Mathematics, Statistics and Computer Science* under my/our supervision.

PriyankaMekala

Prof. Priyanka Mekala
Thesis Supervisor

Supriya

Prof. Supriya Goel
Thesis Supervisor

Acknowledgments

I would like to express my special thanks of gratitude to my supervisors Prof. Supriya Goel and Prof. Priyanka Mekala, as well as both the directors of CR Rao AIMSCS and DA-IICT who allowed me to do this wonderful project on the topic Malicious Node Detection in Heterogeneous Internet of Things, which also helped me in doing a lot of research. I came to know about so many new things I am thankful to them.

Secondly, I would like to thank my parents and friends who helped me complete this project within the limited time frame.

Contents

Abstract

In recent years security is an increased concern for IoT devices. Due to limited capabilities compared to traditional computer systems, these tiny devices cannot run the heavy encryption algorithms required for preventing attacks. Nowadays, IoT comprises several communication protocols like Bluetooth Low energy, WiFi, and Zigbee for different applications including home automation, smart city etc. With such a heterogeneous system, it becomes complex to provide security as with every different protocol comes more vulnerabilities in the network.

Anomaly-based detection methods have received increasing interest from the scientific community in the last few years. It acts as a second layer to the system's security. With deep packet inspection, it evaluates the network traffic and forms a set of informative features formalizing the normal and anomalous behavior of the system. We classify among a normal or abnormal activity using machine learning algorithms and present the results of our detection system implemented on a heterogeneous IoT testbed. This system is applicable for companies, offices, government organization or secret agencies who want to increase their network security to protect their systems.

Keywords: IoT, Security, Anomaly Detecion, BLE, WiFi, Communication Protocols

List of Principal Symbols and Acronyms

BLE	Bluetooth Low Energy
DoS	Denial of Service
IDS	Intrusion Detection System
IoT	Internet of Things
LoF	Local Outlier Factor
MAC	Media Access Control
MITM	Man in the Middle
RSSI	Received Signal Strength Indicator

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1 Background

IoT or Internet of Things is a concept referring to the billions of everyday physical devices around the world embedded with sensors, software, and other technologies, all exchanging data via the internet. The number of these IoT devices has reached 9.5 billion in 2019 [1] from 1.7 billion in 2015 [2] and is further expected to grow at a much faster rate. With the growing number of devices, there are increasing numbers of challenges in IoT like security, powering billions of sensors, e-waste, [1] and among that security has turned out to be a significant concern. According to SonicWall Research labs, there's a 30% increase in IoT malware attacks (i.e. a total of 32.4 million) worldwide in the third Quarter of 2020 [1]. 2016 Dyn cyberattack is an example of increased concern in which Mirai malware used IoT endpoints as botnets and used them to launch DDoS attacks against well-known websites. Today, IoT devices are embedded all around us in seemingly everything, e.g. wearables, cars, homes, and some of that IoT communication will contain data that must be protected or otherwise shielded. Compromising a single component or communication channel in IoT-based systems can paralyze the part or complete Internet network.

1.2 Architecture

Although there is no standard architecture for IoT, it can be divided into four layers, first is physical or perception consisting of sensors interacting with the physical world, second is the network layer (gateways and routers), it aggregates data from various sensors, and third is processing layer which has data processing algorithms and stores and processes the data sent from gateways to the cloud, and fourth is the application layer that acts as an interface and provides a view of analytics [4]. The below figure 1.1 represents a three-layer architecture combining the cloud layer and the application layer. For example, in healthcare, body sensors will form the physical layer, gateways will include the network layer, and the application layer will display readings such as medical intakes, physical movements. The given architecture makes it easy for us to identify the vulnerabilities.

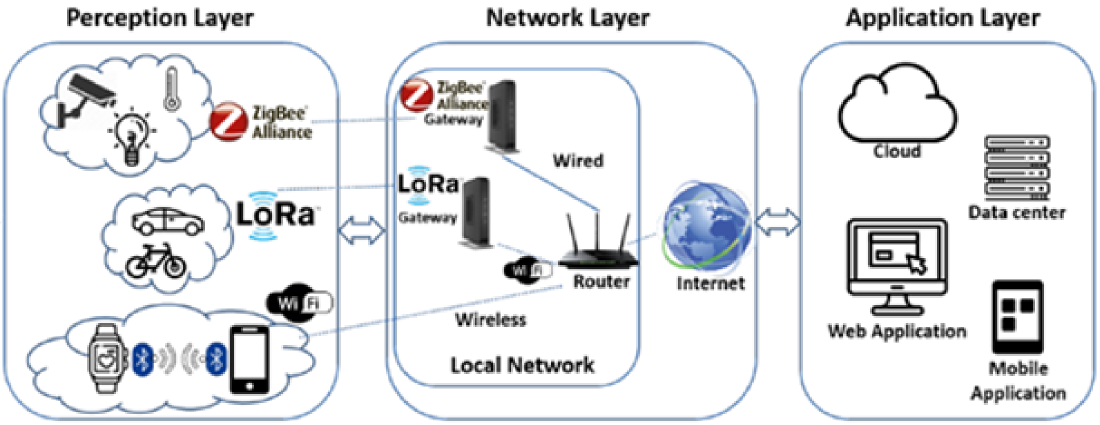


Figure 1.1: Architecture of IoT [4]

1.3 Limitations and Threats

IoT devices are small in size and constrained in power, memory, computational capability, network bandwidth and all these factors pose several limitations in the areas of security and privacy. IoT cannot adopt the same security system as used till now because of the constraints mentioned before and due to IoT's heterogeneous nature in terms of protocols Like BLE and IEEE 802.11 employed in the single system.

IoT deals with vast amounts of data which makes it more prone to flooding attacks. As the IoT system expands, attack methods have also become more complex. Today's networks are vulnerable to:

- Denial of Service - In DoS, an attacker makes a machine or network resource unavailable for users.
- Spoofing - In a spoofing attack, attacker impersonates the identity of legitimate node and sends malicious traffic to disturb the normal operation of the network.
- MITM (Man in the Middle) - Spoofing attack can lead the attacker in the "middle" of the conversation between two parties where this illegitimate node can either eavesdrop or insert malicious data.
- Botnet attacks - by forming a network of hacker controlled IoT nodes which are further used to carry out mass attacks.
- Physical attack - Such attacks risk is also there as sensors are deployed in open spaces.

The traditional security approach to cope up with attacks includes encryption, authentication, firewall, etc. Encryption is an effective countermeasure but due to the lack of computing power it's not so easy to encrypt all the data and hence we require a second layer of security. Thus we propose a novel and next-generation anomaly detection system that works with IoT devices and can detect behavioral anomalies in the heterogeneous network. Such an intrusion detection system is independent of the type of IoT devices, communication protocols, and network structure.

1.4 Objective

The goal of our thesis is to make an anomaly detection system for heterogeneous IoT infrastructure. Most of the work done is concentrated on specific communication protocol or has not been implemented and little goes for spoofing detection in heterogeneous IoT networks along with the implementation. Spoofing attacks can lead to other type of attacks, via this, an attacker can disturb the normal operation of the network, gain control of devices or remove some devices from the network. The main goals of this report are to develop the system with the following features:

- Raspberry Pi based system that detects anomalies in the network (that points to an attack) irrespective of IoT Communication protocols
- Acceptable detection rate, lower false positives and negatives without adding any overhead to the IoT devices and network.
- Our system should be applicable for small companies or offices who want to increase their network security to protect their systems, but cannot afford to invest in an enterprise-grade solution.

While achieving this certain assumptions are made, such as the adversary cannot fully control the target device physically or remotely, detection infrastructure is secure and can’t be compromised by the attacker. The attacker doesn’t use jamming, and it is an indoor environment with static sensors.

Remaining thesis is organized as follows- Chapter 2 covers the Background work which gives an insight into the related work and malicious node detection. Chapter 3 discusses the methodology and framework used. The experiment and hardware details are presented in Chapter 4. Finally, Chapter 5 contains a discussion of the results and their importance and uses. The thesis ends with a conclusion in Chapter 6 which summarizes and concludes the thesis.

CHAPTER 2

Related Work

2.1 Malicious Node Detection

Any node which follows the abnormal activity to attack other nodes in network, such node is said to exhibit malicious behavior. It can disturb the operation of the entire network. We will discuss several security solutions to detect such malicious behavior in this section. Firewalls and anti-malware software alone is not enough to protect an entire network from attack. The identity of a malicious node can be verified through cryptographic algorithms but authentication is not always possible because of additional infrastructural overhead due to key handling and management. Many papers have applied an Intrusion Detection System (IDS) for increased security. Intrusion detection is a monitoring system that refers to the detection of malicious activity and gives immediate alerts. An intrusion is different from the normal behavior of the system, and hence anomaly detection techniques are applied in the intrusion detection domain. An intrusion detection system works by the theory that there is a difference in the network flow between legitimate users and malicious users. It can be designed in various ways, some are shown in the block diagram of Fig 2.1. An IDS uses two different main methods to classify malicious activity: Statistical Anomaly Detection and Rule Based Detection

- Rule-based detection system in which a set of pre-defined rules (requires ex-

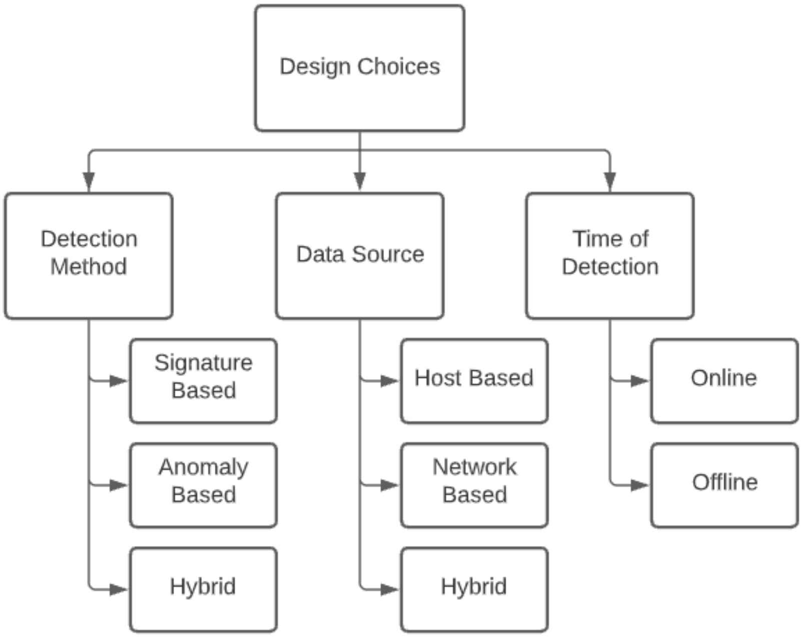


Figure 2.1: Design Choices [5]

pert knowledge) are used to detect any malicious activity. In this, network traffic is examined for known attack patterns (known as signatures) and matches the patterns which point to an attack. It fails to detect unknown or modified attacks and those attacks that exploit the recently discovered vulnerabilities i.e. Zero-day attacks. Since they require to save huge amounts of rules (expert knowledge), they can be resource demanding and therefore, it is not possible to use a pattern based approach for IoT networks.

- Anomaly based detection systems compare a normal recorded behavior with current input and use a type of statistical calculation to determine anomalies in the network traffic. The advantage is that a new attack for which a signature doesn't exist can be detected. Moreover, the IoT technologies do not remain the same forever. Instead, they continue to evolve and implement new functions e.g. BLE, Wi-Fi, and hence new vulnerabilities with new attack signatures. The problem with these systems is that they generate a vast amount of inappropriate false alarms whenever abnormal activities are detected and are not too flexible for a complex environment.

- Hybrid detection- They start with anomaly detection, then try to relate it with the corresponding signature e.g. Kalis.

An IDS can be positioned in different locations within the network. When placed in the physical network it is known as a network-based IDS and if on a host system, it is known as a host based IDS. In classical settings detection system is deployed on hosts directly on a network device such as a router, a switch, a host system or it can be placed as its dedicated device but such host-based approaches cause issues, due to resource constraints of devices processing and deep inspection analysis is only possible for selective hosts in a system. The optimal placement for an IDS depends on the environment it is going to protect. And instead of capturing packets at the network layer and inspecting their payloads, deep inspection of header data is done as packet payloads are often encrypted. If it finds any data packets that are suspicious it will send an alert. The alert is written to a log file that can be read by administrators.

2.2 Literature Survey

A lot of methods are used for the objective of malicious activity detection. To secure the network [6] presents lightweight encryption algorithms for low-resource devices for IoT environments. [7],[8] use an n-gram based approach to characterize the normal behavior of the Bluetooth and HTTP protocol respectively. Several machine learning algorithms [4],[9],[10] as well as deep learning are being explored to improve the accuracy, reduce the false positives and the overhead to the system. [11] Follows a method to distribute an anomaly detection scheme across several end-devices. In [12], the authors focused on the security of IoT devices in the smart city and proposed a Random Forest ML-based architecture called Anomaly Detection-loT (AD-IoT) system. The proposed technique can efficiently identify any sort of suspicious activity happening at the distributed fog nodes using machine learning-based dataset evaluation. But experiments on datasets are challenging as labeled samples of attacks are difficult to obtain, and this data also usually becomes quickly outdated by new attacks. The most commonly used datasets by researchers to design new IDS are the NSL-KDD and DARPA. Both datasets were created more than a decade ago and hence problem is that neither can reflect the network behaviors of current IoT networks.

Zarpelão et al. [13] surveyed intrusion detection research efforts for IoT and classified them based on detection method, placement strategy, security threat, and validation strategy. The main observation of the authors is that intrusion detection schemes for IoT are still emerging. In particular, they noted that the proposed solutions do not cover a broad range of attacks and IoT technologies. Moreover, many of the currently offered schemes have never been thoroughly evaluated and validated.

SNORT [14] is the most widely used Signature-based Intrusion Detection/ Prevention system. SNORT detects intrusion attempts by analyzing network traffic in real-time. In SNORT, the signature compromises of the header that consists of the source address, destination address, and ports and its options that include payload and metadata which are used to determine whether the network traffic corresponds to a known signature or not. But Snort becomes too resource-demanding for IoT hence Alessandro et al. propose RPiDS [15], an IDS architecture customized for IoT environments that includes Raspberry Pi equipped with Snort. Results show that the Raspberry Pi can host Snort, hence providing portability and security on demand (i.e., everywhere, anytime).

Gajewski et al. focuses on attacks on Home automation systems and has proposed two step method to improve attack detection accuracy. First, packet traffic data is collected on the Home gateway and with the help of ML algorithm (NB model) classifies the data records as normal or anomalous. Second, Across ISP data center behavioral similarities are found between observations from Home gateway's (this correlation process requires more computational resources), if any similar anomalies are found that might point to an attack.

Kalis is one of the few methods that do not target individual protocols. It follows a Hybrid approach and proposes that processing network events and traffic through all the detection techniques requires a high amount of system resources and can cause delays in attack reaction but we can restrict our set of attack detection techniques by knowing what attacks are possible in that network. From collected observations, Kalis finds out characteristics of the network (whether mobile or static, powerful or constrained) and then find out possible selective attacks and hence narrows down the detection set.

Due to various vulnerabilities in BLE, spoofing attacks are a threat to BLE devices. All spoofing attacks will result in certain anomalous features in the advertisement packets that will contain BLE's identity. Jianliang et al. has proposed BlueShield to obtain certain features like advertising interval, RF signal frequency offset Received Signal Strength Indicator (RSSI) and determine those anomalies.

Summerville et al. have proposed a lightweight packet anomaly detection approach that is feasible to run on resource constrained IoT end nodes. using pattern matching to detect signs of undesirable activities. The authors propose a technique called bit pattern matching for small IoT devices to perform feature selection. As these devices use few and relatively simple protocols and hence are highly similar in network payloads.

2.3 IoT Communication Protocols

Out of several communication protocols in IoT like Bluetooth, Wifi, Zigbee, Lora, Nb-IoT, we have considered the first three due to their popularity and compatibility with the upcoming devices. The main difference among the three is shown in table 2.2 [15].

	Bluetooth Low Energy (BLE)	**Wi-Fi**	**Zigbee**
IEEE Spec	802.15.1	802.11	802.15.4
Max. Data Rate	1Mbps	54 Mbps	250kbps
Power Consumption	0.01-0.5W	High	30mW
Frequency	2.4GHz	2.4Ghz and 5GHz	2.4GHz, 900MHz and 868MHz
Applications	Smart Wearables	Broadband Internet Access	Home Automation (Monitoring and Control)

2.3.1 Bluetooth Low Energy

Bluetooth Low Energy (BLE) [19] was started as Bluetooth 4.0 with the goal of low power consumption. The modulation technique used is GFSK and the theoretical upper limit for the data rate is 1Mbps. Its throughput is around hence other technologies like Wi-Fi still have maintained their position. The standard uses a technique called FHSS frequency hopping spread spectrum, in which the radio hops between channels on each connection event. Since Wi-Fi and classic Bluetooth also work in the 2.4 GHz band, this technique avoids interference. BLE has two types of packets

1. Advertising - It broadcast data for applications that do not require full connection establishment or to discover slaves. This type of packet consists of up to 31 bytes of payload along with the basic header information such as Bluetooth device address. These packets are broadcasted blindly without scanning any nearby device and the state of the device is said to be in advertising mode.

2. Data packets - These packets carry general user data between master and slave and the device is said to be in a connected state (transitions between states are represented in Fig 2.3).

Around 80 % of BLE devices do not include secure pairing and communicate data without using any secure authentication mechanism. As the attacks are getting more and more complex there comes the need to upgrade the BLE devices with sophisticated security features, and in the meantime, adopt security patches to fix the vulnerabilities. This may not be preferably adopted by different vendors at a largescale, or for limited or no I/O capabilities or ensure backward compatibility with the legacy BLE devices.

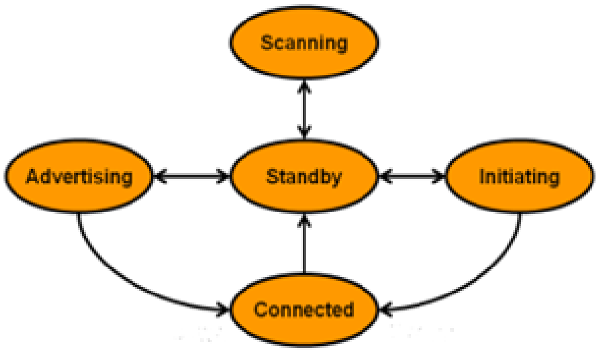


Figure 2.2: State Diagram of BLE [27]

2.3.2 Wi-Fi

Wi-Fi (Wireless Fidelity) [30] is used for in-building network connectivity. IEEE 802.11 specifies the MAC and Physical layer protocols for implementing Wi-Fi communication. The Wi-Fi protocol operates in 2.4 GHz and 5 GHz bands. Wi-Fi networks are still vulnerable to attacks on the data link layer. Data link layer attacks like the deauthentication attack can successfully attack devices on all Wi-Fi networks. These attacks can be used to execute man-in-the-middle attacks (MITM), and denial of service (DoS) attacks. There is a need for an intrusion detection system that can detect attacks on the Wi-Fi protocol, with low false positives and negatives alerts.

2.3.3 Zigbee

Zigbee [30] popular for low-power mesh network speed of 250 kbit/s, 128-bit AES encryption, and numerous power-saving features such as sleep with scheduled wakeups. IoT works with different protocols on different levels and Zigbee is one among them It is based on IEEE 802.15.4 that defines its Physical and MAC layers. Zigbee's main objective is to save energy such that sensors can survive years with an AA battery. Applications include a home automation system, smart lighting system.

Zigbee consists of three main components: Coordinator, Router, End Device. a Zigbee coordinator is the only device type that can start a network, each Zigbee network must have only one coordinator. Zigbee routers act as intermediary devices that permit data to pass to and fro through them to other devices. A Zigbee End Device, used in our experiment, provides only basic functionality and cannot send or receive directly with other devices.

The protocol features both network-wide and pairwise encryption and authentication still have various vulnerabilities that have been identified and fixed in the past. One of Zigbee's weak points lies in the network coordinator's initial handshake with a joining device, which is unencrypted. Zigbee security is necessary as our door lock has the same security as our lights.

CHAPTER 3

Methodology

IoT devices are prone to spoofing attacks where an attacker can impersonate a benign BLE device and feed malicious data to its users. One important type of attack that leads to several others (e.g DoS, MITM, energy depletion, jamming, and various protocol-level exploits) and is easy to launch is spoofing. We aim to detect these spoofing attacks and detect that malicious node with the experiment setup of figure 3.1. Broadly our work phase can be said to be in three steps. First, node identfication, then, analyze and extract the features and lastly classify the abnormal data (also shown in Fig 3.2). Our focus will be on the parameters that are hard to predict by the attacker. Each advertising packet contains a unique identifier of the device and the information about services provided by it.

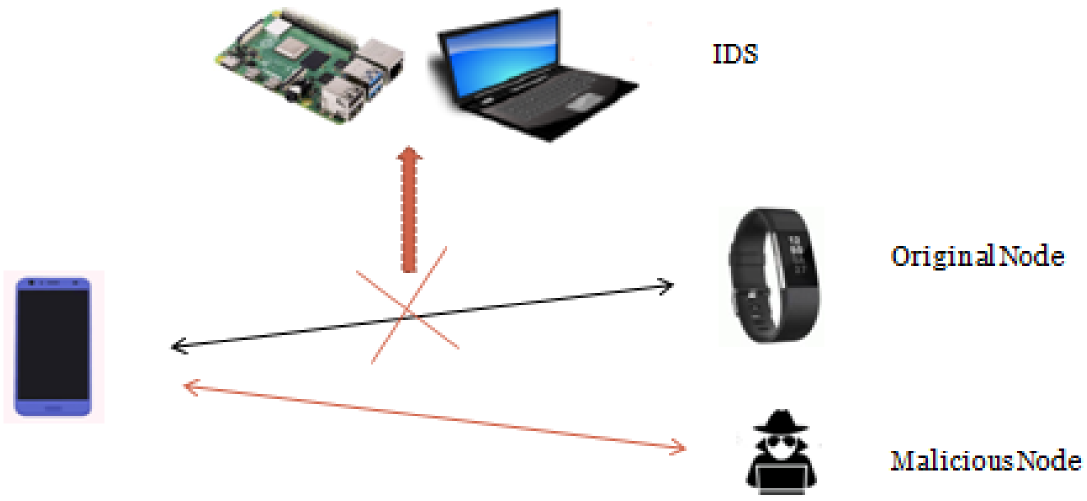


Figure 3.1: Experiment setup

3.1 Features Extraction

Feature selection is the most critical step in building threat detection models. During this step, the set of attributes or features deemed to be the most effective attributes is extracted to construct suitable detection algorithms (detectors). Defining an extensive feature space can potentially enable the system to detect diverse attack types but redundant features that only introduce system overhead and delay in the process of attack detection and prevention must be avoided. Our focus will be on the parameters that are hard to predict by the attacker.

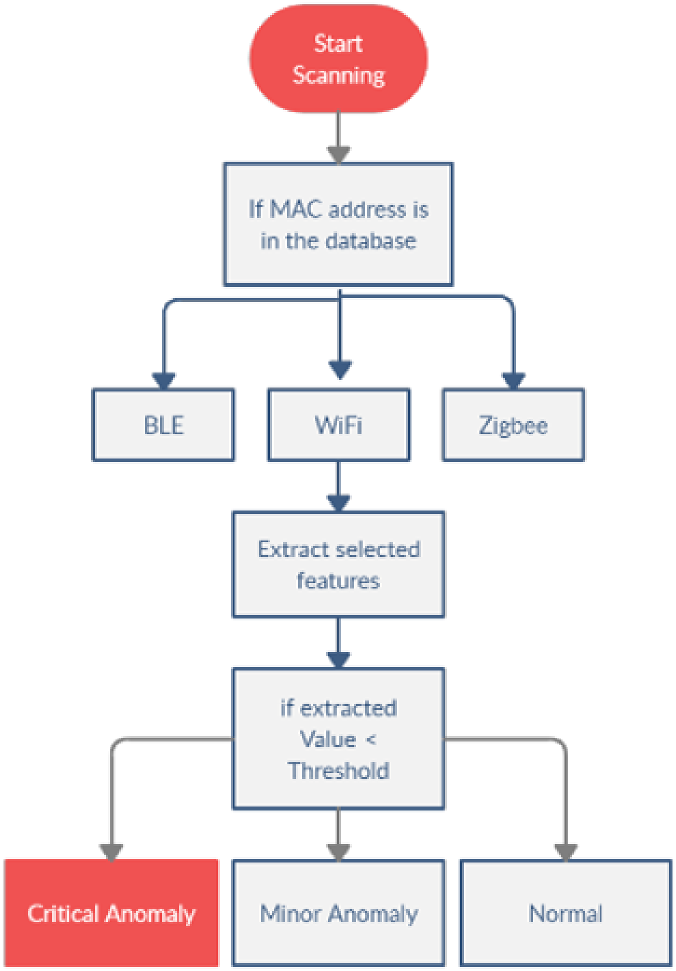

Start Scanning
If MAC address is in the database
BLE
WiFi
Zigbee
Extract selected features
if extracted Value < Threshold
Critical Anomaly
Minor Anomaly
Normal

Figure 3.2: Flow Chart of Method

3.2 Anomaly Detection

3.2.1 RSSI

The authentic value of the physical features (e.g., RF signal's strength) of the advertising packets would be different at different channel path, this mechanism ensures that even an attacker with the capability to mimic all the physical features, cannot trick correctly imitated values at the same time.

The detection system keeps track of RSS values of all network nodes and analyzes them for signs of spoofing attempts. Once an attack is detected, the detection module sends a message to the victim, informing it about the presence and identity of an attacker. If R doesn't lie within the range Rmin < R < Rmax, the algorithm stops at this step and raises the alarm declaring the presence of a spoofing attack.

This experiment will involve a static threshold as the dynamic threshold means more computations and introduces a short delay for queuing and clustering of frames.

3.2.2 Advertising Interval

If both the BLE device and attacker are broadcasting advertising packets, the real-time advertising interval can be lower than the expected INT value, and hence can be utilized to detect spoofing attacks. Some BLE devices may keep advertising even during the connection state, but most (intermittent) will not advertise after connection and those devices can be kept under state supervision to detect advertisement on their behalf by a malicious node even after the legitimate device is connected.

3.2.3 Traffic Features

Numerous researchers are trying to identify the characteristics of traffic generated as a product of IoT device communication. The traffic characteristics (such as number of packets, last active time duration) generated by individual IoT devices can be a key factor in researching the relationships of generated traffic to certain processes in the communications network. Such features are used to identify IoT devices in the network, detect unauthorized devices in the network, and detect network traffic anomalies.

3.2.4 Sequence Number

Both 802.11 management and data frames carry a sequence control field in their MAC headers. Each frame carries a sequence number one unit greater than the one in the immediately preceding frame. The sequence number can be used to detect impersonation attacks because when spoofing a MAC address as the attacker cannot keep the spoofed frames sequence number.

3.2.5 Techniques for Detection

- In general, any density estimation approach can be applied to model the normal classes. These rely either on the distance from the neighbouring point or relative data density. Density around an anomalous point should be significantly lower than the normal ones. Some well known approaches are kNN and LoF. Normal data are far larger in number than anomaly data hence the one-class classification algorithm can be used to identify one particular class by primarily learning from a training set.

- Statistical Methods: Statistical methods detect anomalies mainly dependent on the predefined threshold, mean and standard deviation, and probabilities. It follows the rule that normal data will fall under high probability regions and outliers will fall under the low probability area. Probability threshold will divide the mentioned two regions.

- Isolation Forest is designed to support anomaly detection algorithm which is based on decision trees approach. It works by randomly selecting features and isolates each point with the assumption that anomalous points will be isolated first. The contamination parameter in this represents the percentage of outliers in the dataset and can be found by trial and error.
- Clustering based techniques can also be used where the normal belong to clusters whereas anomalous ones do not

CHAPTER 4

Experiment Design

4.1 Experiment

4.2 Hardware Description

In this section, the software and measuring tools used in the experiments will be shown and explained. Furthermore, the commands used to control the software and the tools will also be shown and explained.

4.2.1 Raspberry Pi

Raspberry Pi is a line of small single-board computers that come in a variety of configurations, each having different central processing unit, memory capacity, networking capabilities, and peripheral device support. In June 2019, the Raspberry Pi 4 Model B (Fig. 4.1) was released, featuring a 1.5 GHz 64-bit quad-core ARM Cortex A72 processor, full gigabit Ethernet (throughput not limited), onboard 802.11ac Wi-Fi, Bluetooth 5, dual-monitor support via a pair of micro HDMI ports for up to 4K resolution, two USB 3.0 ports and two USB 2.0 ports. Raspberry Pi ran the operating system Kali Linux. Raspberry Pi was chosen due to its compactness and portability.

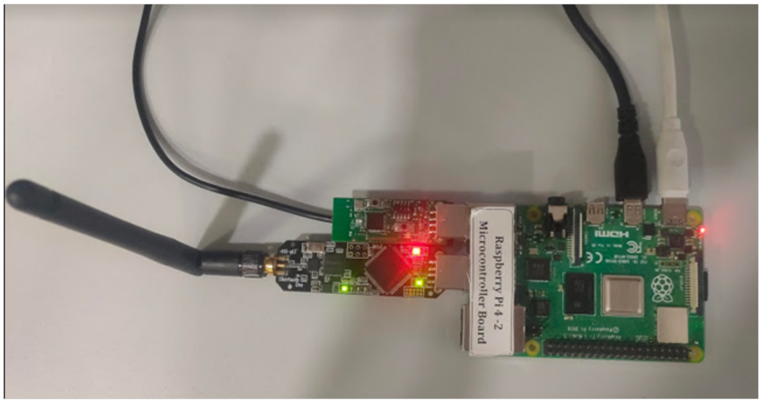


Figure 4.1: Raspberry Pi 4 Model B

4.2.2 Ubertooth One

Michael Ossmann's Ubertooth One is an open-source Bluetooth test tool. It's the world's first low-cost Bluetooth monitoring and development platform, and it's completely open-source (both hardware and software). The device's main job is to decode Bluetooth packets, but it can also determine the the master piconet MAC address, get the packet's Signal to Noise (SNR) ratio, figure out the device's hopping sequence, and use the spectrum analyzer to reveal any Wi-Fi or Bluetooth activity in the 2.4 GHz region. Ubertooth consists of an antenna, RF front end, and a wireless transceiver composed of two integrated circuits, which are responsible for conditioning the received signal and preparing it for processing by the microcontroller.

4.2.3 WiFi Internal Card

The system has an internal Wifi card of Broadcom BCM 4321 that supports wireless monitor mode. When set up in monitor mode (default is managed mode), it enables a device with a wireless network interface controller to monitor all traffic received from the wireless network. In short, it helps to “monitor” the packets that are received without any filtering and without linking any access point. The tool used to change the mode to monitor mode is Aircrack-ng and with the following commands:

```
sudo airmon-ng start wlan0
sudo airmon-ng check kill
sudo airodump-ng wlan0mon
```

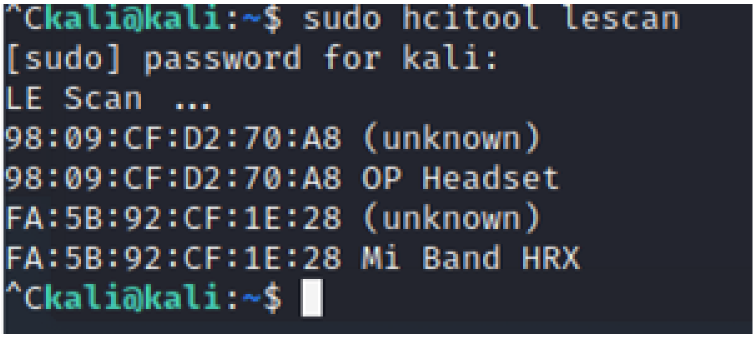
4.2.4 Zigbee Sniffer CC2531

The CC2531 USB dongle is a fully operational USB device that provides a PC interface to IEEE802.15.4 / ZigBee applications. The dongle can be plugged directly into your PC/Raspberry etc used as a Zigbee packet sniffer. Commands used are with the zigbee2mqtt tool [31]:

```
cd /opt/zigbee2mqtt
npm start
```

4.2.5 Software Tools

- Spooftooph: Spooftooph is a tool designed to automate spoofing or cloning Bluetooth information (Name, Class, and Address). We spoofed the BLE earphones and started advertising packets as the same, shown in Fig 4.2 in the name of OP Headset and 98:09:CF:D2:70:A8 MAC address discarding its original name and MAC address.



```
^Ckali@kali:~$ sudo hcitool lescan
[sudo] password for kali:
LE Scan ...
98:09:CF:D2:70:A8 (unknown)
98:09:CF:D2:70:A8 OP Headset
FA:5B:92:CF:1E:28 (unknown)
FA:5B:92:CF:1E:28 Mi Band HRX
^Ckali@kali:~$ 
```

Figure 4.2: Malicious Node

- Python: Scapy, Bluepy are python libraries used that can handle Wifi and Bluetooth scanning.
- Wireshark: Wireshark is a free and open-source packet analyzer. It helps in deep packet inspection of protocols, live inspection, and offline analysis.

4.3 Testbed

To measure the performance of the proposed spoofing detection mechanism, we experimented with a heterogeneous network testbed. The testbed was located in an office building at the IoT lab in CR Rao AIMSCS Institute, Hyderabad. The network contained seven nodes including one victim, an attacker, and five genuine nodes. We utilized six different IoT devices as shown in figure 4.4, these cover the mainstream IoT applications (e.g. Smart Plug, Watch, Camera, earphones, Voice Assistant) and popular manufacturers like Xiaomi, Qubo, OnePlus, Tuya. Labeled dataset for the internet of things (IoT) is very rare and difficult to find. So with this testbed, our database was created and stored. IoT devices were arranged as shown in Fig. 4.3 with a static environment.

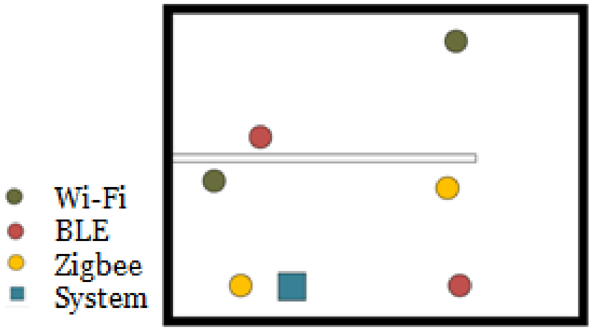


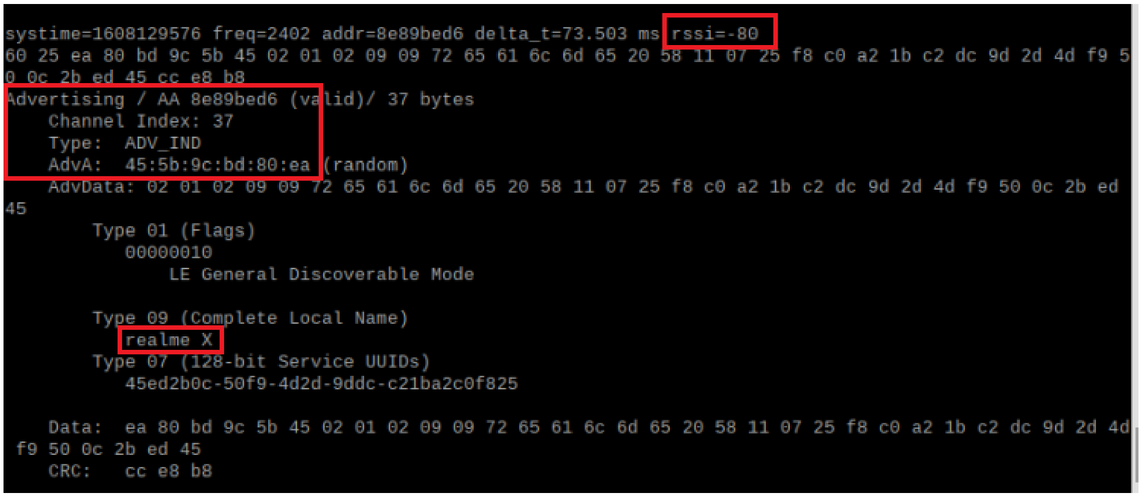
Figure 4.3: Overview of IoT Lab

Figure. 4.3 shows the block diagram of the connection between the nodes placed in different places and the RPi

In practice, before any connection takes place, we want to find our device first (Node Scanning). The system will scan the data packets in the nearby area with

Figure 4.4: IoT Devices

the help of given sniffers. Most of the BLE devices send broadcasts on three channels (37, 38 and 39) periodically, so that they can be detected. We have used Raspberry Pi 4 as our system, Ubertooth one as BLE sniffer and the packets sniffed are being analyzed in the Wireshark tool. Wireshark is a free and open-source packet analyzer. It can be used to capture, examine, analyze and visualize packets or frames.



```
systime=1608129576 freq=2402 addr=8e89bed6 delta_t=73.503 ms rssi=-80
60 25 ea 80 bd 9c 5b 45 02 01 02 09 09 72 65 61 6c 6d 65 20 58 11 07 25 f8 c0 a2 1b c2 dc 9d 2d 4d f9 5
0 0c 2b ed 45 cc e8 b8
Advertising / AA 8e89bed6 (valid)/ 37 bytes
    Channel Index: 37
    Type:  ADV_IND
    AdvA:  45:5b:9c:bd:80:ea (random)
    AdvData: 02 01 02 09 09 72 65 61 6c 6d 65 20 58 11 07 25 f8 c0 a2 1b c2 dc 9d 2d 4d f9 50 0c 2b ed
45
        Type 01 (Flags)
           00000010
               LE General Discoverable Mode

        Type 09 (Complete Local Name)
           realme X
        Type 07 (128-bit Service UUIDs)
           45ed2b0c-50f9-4d2d-9ddc-c21ba2c0f825

    Data:  ea 80 bd 9c 5b 45 02 01 02 09 09 72 65 61 6c 6d 65 20 58 11 07 25 f8 c0 a2 1b c2 dc 9d 2d 4d
 f9 50 0c 2b ed 45
    CRC:   cc e8 b8
```

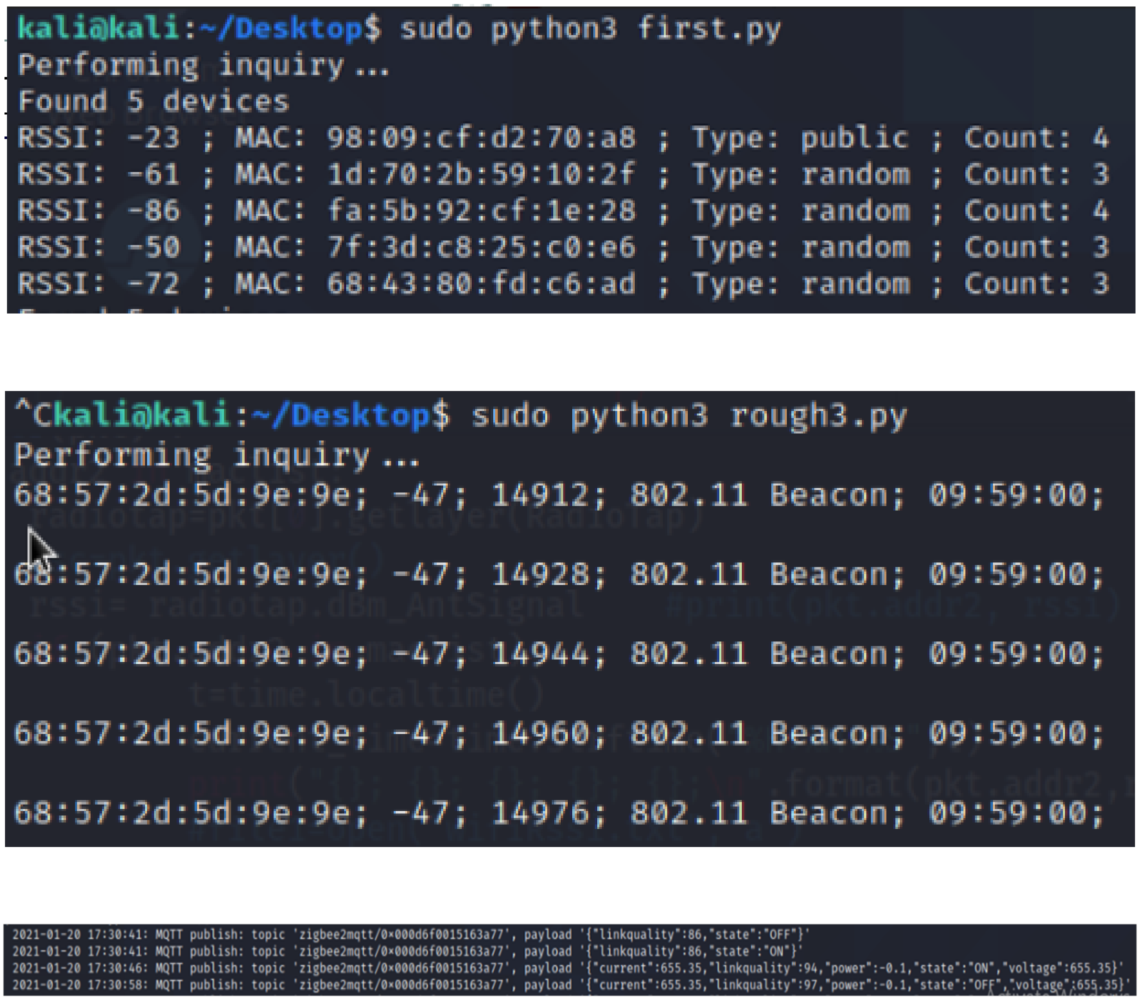
Figure 4.5: Packet Capture

Wireshark captures Bluetooth Low Energy Link Layer format (btle) as the outermost layer of the packet. It contains Access Address, PDU and CRC calculated based on the Payload content. Advertisement data is the most important part of the BLE packet. Its content usually contains flags, the short name of a device and the manufacturer-specific data. All these parameters are visible in Fig 4.5 where the packet dissection of a Realme mobile is shown. Following commands are needed, the first one turns Bluetooth on and the second one is to start capturing BLE packets.

sudo hciconfig hcio up

ubertooth-btle -f -c /tmp/pipe

Figure 4.6 shows how the packet captured can be seen and its MAC address needs to be extracted to identify whether the node belongs to our network. After collecting network traffic, the second step involves extracting the features to identify the devices. We further develop Python codes using the libraries bluepy and scapy to extract the details.



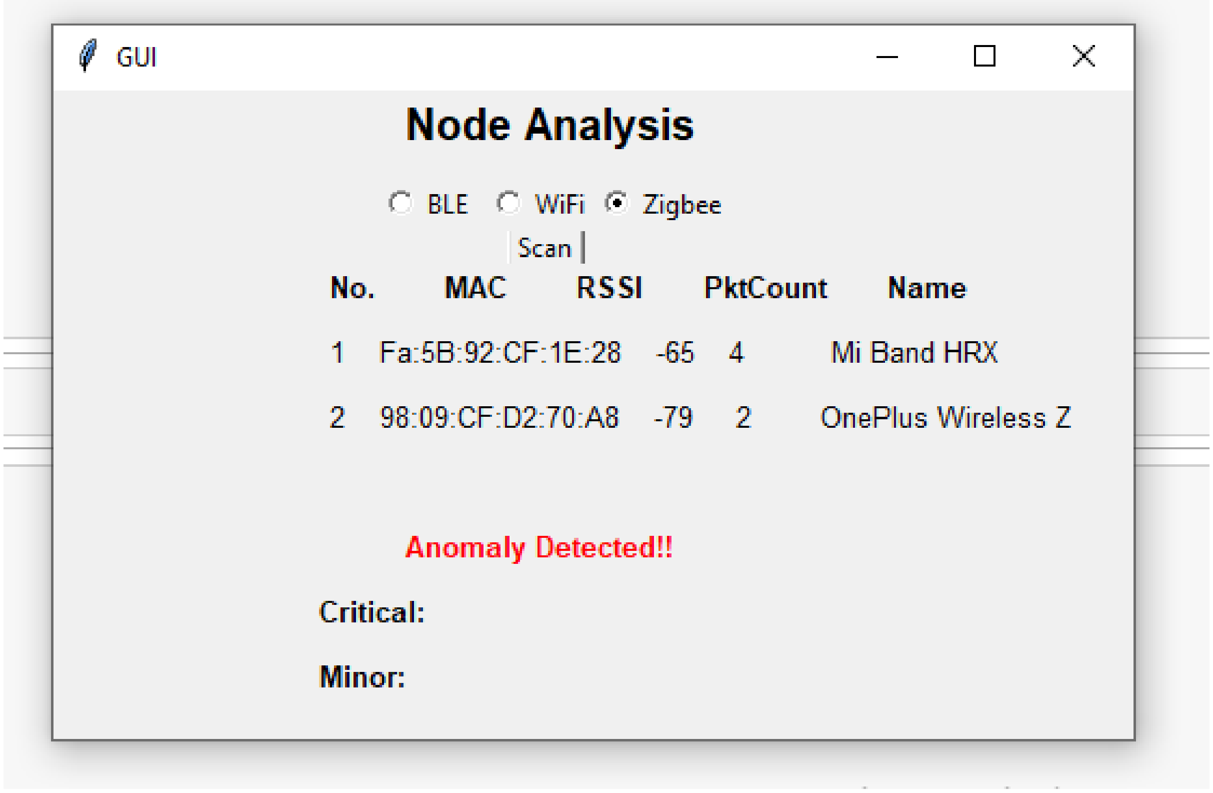
```
kali@kali:~/Desktop$ sudo python3 first.py
Performing inquiry ...
Found 5 devices
RSSI: -23 ; MAC: 98:09:cf:d2:70:a8 ; Type: public ; Count: 4
RSSI: -61 ; MAC: 1d:70:2b:59:10:2f ; Type: random ; Count: 3
RSSI: -86 ; MAC: fa:5b:92:cf:1e:28 ; Type: random ; Count: 4
RSSI: -50 ; MAC: 7f:3d:c8:25:c0:e6 ; Type: random ; Count: 3
RSSI: -72 ; MAC: 68:43:80:fd:c6:ad ; Type: random ; Count: 3
```

```
^Ckali@kali:~/Desktop$ sudo python3 rough3.py
Performing inquiry ...
68:57:2d:5d:9e:9e; -47; 14912; 802.11 Beacon; 09:59:00;

68:57:2d:5d:9e:9e; -47; 14928; 802.11 Beacon; 09:59:00;

68:57:2d:5d:9e:9e; -47; 14944; 802.11 Beacon; 09:59:00;

68:57:2d:5d:9e:9e; -47; 14960; 802.11 Beacon; 09:59:00;

68:57:2d:5d:9e:9e; -47; 14976; 802.11 Beacon; 09:59:00;
```

```
2021-01-20 17:30:41: MQTT publish: topic 'zigbee2mqtt/0×000d6f0015163a77', payload '{"linkquality":86,"state":"OFF"}'
2021-01-20 17:30:41: MQTT publish: topic 'zigbee2mqtt/0×000d6f0015163a77', payload '{"linkquality":86,"state":"ON"}'
2021-01-20 17:30:46: MQTT publish: topic 'zigbee2mqtt/0×000d6f0015163a77', payload '{"current":655.35,"linkquality":94,"power":-0.1,"state":"ON","voltage":655.35}'
2021-01-20 17:30:58: MQTT publish: topic 'zigbee2mqtt/0×000d6f0015163a77', payload '{"current":655.35,"linkquality":97,"power":-0.1,"state":"OFF","voltage":655.35}'
```

Figure 4.6: Scanning of BLE, Wifi and Zigbee

It extracts the device name, the MAC address, and the advertising data INT value = subtracting the time-of-arrival of the current advertising packet from that of the previous advertising packet. As defined by the BLE specification (p. 2750 in [32]), delay = 0 to 10 ms stores the determined characteristics of the BLE device along with an assigned device identifier. After retrieving the advertising packets and their features, inspect these features for each device. The interval between any two advertising packets must always be more than the lower bound of 'INT'. if not, the monitor considers it an anomaly and raises an alarm.



Node Analysis

○ BLE ○ WiFi ● Zigbee

No.	MAC	RSSI	PktCount	Name
1	Fa:5B:92:CF:1E:28	-65	4	Mi Band HRX
2	98:09:CF:D2:70:A8	-79	2	OnePlus Wireless Z

Anomaly Detected!!

Critical:

Minor:

Figure 4.7: Traffic Capture of BLE

Representation of the system was formed with the help of GUI library in Python (Fig 4.7) and the user interface is simplified for the user to check if any anomaly present.

Chapter 5

Performance Evaluation

The experiment was performed with data collected over the testbed for a period of few hours. When few RSSI values were plotted over different distances, we could observe there was a sudden shift in the RSSI values. Hence RSSI could prove the most decisive factor to be considered with respect to the distance factor and such abrupt change in RSSI values of advertising packets can detect an attack. Fig. 5.1 shows the variations and we can easily distinguish among three different distances.

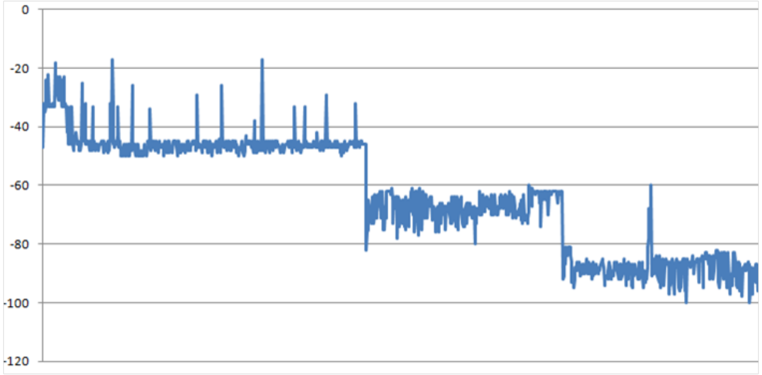


Figure 5.1: RSSI with varying distance

Collection of such values was observed to have close to Gaussian Distribution [35] having probability density function (Fig 5.2) $f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$. This statistical distribution method was used to predict the anomalies along with others. Other unsupervised algorithms were also used as the amount of abnormal data or the anomalies is few or rare, supervised algorithms could not be used. Unsupervised would analyze the data and form boundaries according to normal

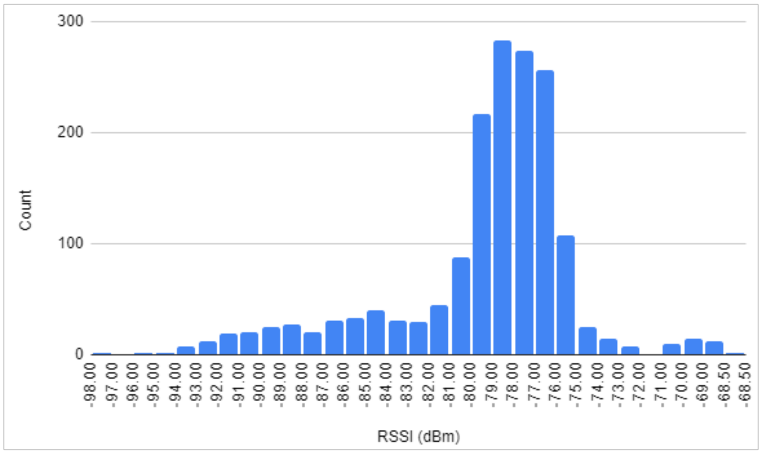


Figure 5.2: Distribution of RSSI

data only, anything that does not fits the data falls in the anomaly category.

Algorithm	TP	FN
Statistical	0.98	0
Isolation Forest	0.83	0.19
Local Outlier Factor	0.98	0.05

Metrics considered for evaluating the algorithms are TP (True Positive) and FN (False Negative). Abnormal RSS values can be due to malicious activity or environmental changes and hence can also cause false alarms. The statistical algorithm has proved to be more accurate and with the least false negatives but others seem to have large FP which is the major issue in anomaly detection problems.

CHAPTER 6

Conclusion and Future Work

Malicious node detection is an important issue in IoT. In the last couple of years, numerous outlier detection approaches have been proposed for IoT. The proposed anomaly detection system based on portable raspberry pi detects anomalies in the network (that points to an attack) irrespective of IoT Communication protocols. The statistical approach has an acceptable detection rate, and the lowest false positives and negatives. It does not add any overhead to the IoT devices and network. The application for this system is to increase network security in small companies or offices in order to protect their systems against increasingly complex threats in such environments. When there is no spoofing, for each MAC address, the sequence of RSS sample vectors are close to each other and will fluctuate around a mean vector. However, under a spoofing attack, there is more than one node at different physical locations claiming the network.

Future work includes developing a classification model of IoT devices in a smart home environment. And integrating our spoofing detector into a real-time localization system that can both detect the spoofing attacks, as well as localize the adversaries in our IoT network. The experiment can be further modified by considering dynamic systems and hence using dynamic thresholds that will constantly be trained during implementation and change itself according to prevailing scenarios. Such can be a case in the Smart City network. Further, more unsupervised algorithms can be considered and comparison can be done. As more and more IoT protocols are taking the lead like LoRa, cellular IoT, considering these systems can be truly heterogeneous.

References

1. IC Insights,accessed 24 December 2020 [Online]. Available: https://www.icinsights .com/news/bulletins/Internet-Of-Things-Market-To-Nearly-Double-By-2019-/

2. IoT Analytics, K. Lueth, accessed 18 December 2020 [Online]. Available: https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/

3. SonicWall, “Mid-Year Update 2020 SonicWall Cyber Threat Report”, accessed 18 December 2020,<https://www.sonicwall.com/2020-cyber-threat-report/>

4. Chaabouni, N.; Mosbah, M.; Zemmari, A.; Sauvignac, C.; Faruki, P. ,“Network Intrusion Detection for IoT Security Based on Learning Techniques”. IEEE Commun. Surv. Tutor. 2019, 21, 2671–2701.

5. J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, “A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions,” Electronics (Basel), vol. 9, no. 7, pp. 1177–, 2020.

6. S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, “Advanced lightweight encryption algorithms for IoT devices: Survey, challenges and solutions,” J. Ambient Intell. Humanized Comput., pp. 1–18, May 2017

7. P. Satam, S. Satam, and S. Hariri, “Bluetooth intrusion detection system (BIDS),” in Proc. IEEE/ACS 15th Int. Conf. Comput. Syst. Appl. (AICCSA), Oct. 2018, pp. 1–7

8. Y. An, F. R. Yu, J. Li, J. Chen, and V. C. M. Leung, “Edge intelligence (EI)-enabled HTTP anomaly detection framework for the Internet of Things (IoT),” IEEE Internet Things J., vol. 8, no. 5, pp. 3554–3566, Mar. 2021.

9. M. Hasan, M. M. Islam, M. I. I. Zarif, and M. M. A. Hashem, “Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches,” Internet Things, vol. 7, Sep. 2019, Art. no. 100059

10. U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, and M. Stanley,“A brief survey of machine learning methods and their sensor and IoT applications,” in Proc. IEEE Conf. Inf. Intell. Syst. Appl., Mar. 2018, pp. 1–8.

11. Q. Du , Y. Wei , Y. Mao ,“Distributed Deployment of Anomaly Detection Scheme in Resource-Limited IoT Devices”, 2019 IEEE 19th International Conference on Communication Technology (ICCT), 2019

12. I. Alrashdi, A. Alqazzaz, E. Aloufi, R. Alharthi, M. Zohdy, and H. Ming, “Ad-iot: Anomaly detection of iot cyberattacks in smart city using machine learning,” in 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, 2019, pp.0305–0310

13. B. B. Zarpelo et al., "A Survey of Intrusion Detection in Internet of Things", J. Network and Computer Applications, vol. 84, pp. 25-37, 2017

14. M. Roesch.,“Snort - lightweight intrusion detection for networks”, In Proceedings of USENIX LISA’99, 1999

15. A. Sforzin, F. G. Mármol, M. Conti and J.-M. Bohli, "Raspberry Pi IDS: A fruitful intrusion detection system for IoT", Proc. 13th IEEE Int. Conf. Adv. Trusted Comput. (ATC), pp. 440-448, 2016

16. Gajewski M, Mongay Batalla J, Mastorakis G, Mavromoustakis CX, “Anomaly traffic detection and correlation in smart home automation IoT systems”, Trans Emerg Telecommun Technol. 2020.

17. Alrashdi, I.; Alqazzaz, A.; Aloufi, E.; Alharthi, R.; Zohdy, M.; Ming, H.,“AD-IoT: Anomaly Detection of IoT Cyberattacks in Smart City Using Machine Learning”, In Proceedings of the 2019 IEEE 9th Annual CCWC, January 2019; pp. 305–310.

18. D. Midi et al., "Kalis-A System for Knowledge-Driven Adaptable Intrusion Detection for the Internet of Things", Proc. 2017 IEEE 37th Int’l Conf. Distributed Computing Systems (ICDCS 17), pp. 656-666, 2017.

19. Jianliang Wu, Yuhong Nan, Vireshwar Kumar, Mathias Payer, and Dongyan Xu., “BlueShield: Detecting Spoofing Attacks in Bluetooth Low Energy (BLE) Networks”. In 23nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020), 2020

20. D.H. Summerville, K.M. Zach, and Y. Chen, “Ultra-Lightweight Deep Packet Anomaly Detection for Internet of Things Devices,” Proc. 2015 IEEE 34th Int’l Performance Computing and Comm. Conf. (IPCCC 15), 2015.

21. J. S. Lee, Y. W. Su, and C. C. Shen, “A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi,” in Proc. IEEE 33rd Annu. Conf. IECON, Nov. 2007, pp. 46–51.

22. P. Jokar, N. Arianpoo, and V. C. M. Leung, “Spoofing detection in IEEE 802.15.4 networks based on received signal strength,” Ad Hoc Netw., vol. 11, no. 8, pp. 2648–2660, 2013.

23. B. Stelte and G. D. Rodosek, “Thwarting attacks on ZigBee—Removal of the KillerBee stinger,” in Proc. Netw. Service Manag. (CNSM), 2013, pp. 219–226.

24. Y. Chen, W. Trappe, and R. P. Martin, “Detecting and localizing wireless spoofing attacks,” in Proc. IEEE Conf. on Sensor, Mesh and Ad Hoc Commun. and Networks, 2007.

25. G. Cerar, H. Yetgin, and C. Fortuna, “Learning to Detect Anomalous Wireless Links in IoT Networks,” in IEEE Access, 2020, pp. 363–368.

26. Z. Tang, Y. Zhao, L. Yang, S. Qi, D. Fang, X. Chen, X. Gong, and Z. Wang, “Exploiting Wireless Received Signal Strength Indicators to Detect Evil-Twin Attacks in Smart Homes”, Mobile Information Systems, vol. 4, , pp. 1-14, 2017.

27. Microchip Developer, "BLE Link Layer Roles and States", accessed on Jan 2021, [Online]. Available: https://microchipdeveloper.com/wireless:ble-link-layer-roles-states

28. S. Siby, R. R. Maiti, and N. Tippenhauer, "IoTScanner: Detecting and classifying privacy threats in IoT neighborhoods.", Jan, 2017

29. S. Hussain, M. S. Rahman, Using received signal strength indicator to detect node replacement and replication attacks in wireless sensor networks, SPIE Proceedings on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, Orlando, USA, Vol. 7344, 2009

30. S. Sadowski and P. Spachos, “RSSI-based indoor localization with the Internet of Things,” IEEE Access, vol. 6, pp. 30149–30161, 2018.

31. G. pages, “Zigbee2mqtt documentation,” url: https://www.zigbee2mqtt.io/, 2019.

32. Bluetooth, “Bluetooth Smart or Version 4.0+ of the Bluetooth specification”, accessed 18 December 2020

33. X. Fan, F. Susan, W. Long and S. Li, "Security Analysis of Zigbee", Pdfs.semanticsc holar.org, 2017. [Online]. Available: https://pdfs.semanticscholar.org/3d1d/5a51d05 cde08b6e52afd5b d7bc325b487a10.pdf.

34. F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, “Machine learning in IoT security: Current solutions and future challenges,” ArXiv, vol. abs/1904.05735, 2019.

35. S. A. Aljawarneh and R. Vangipuram, “GARUDA: Gaussian dissimilarity measure for feature representation and anomaly detection in Internet of Things,” J. Supercomput., pp. 1–38, 2018.