Threats, Countermeasures, and Research Trends for BLE-based IoT Devices

by

Ashwath Anand Pammi

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2017 by the
Graduate Supervisory Committee:

Adam Doupé, Chair
Partha Dasgupta
Gail Joon-Ahn

ARIZONA STATE UNIVERSITY

December 2017

ABSTRACT

The Internet of Things has conjured up a storm in the technology world by providing novel methods to connect, exchange, aggregate, and monitor data across a system of inter-related devices and entities. Of the myriad technologies that aid in the functioning of these IoT devices, Bluetooth Low Energy also known as BLE plays a major role in establishing inter-connectivity amongst these devices. This thesis aims to provide a background on BLE, the type of attacks that could occur in an IoT setting, the possible defenses that are available to prevent the occurrence of such attacks, and a discussion on the research trends that hold great promise in presenting seamless solutions to integrate IoT devices across different industry verticals.

To my mom and dad, for their patience,

To my brother and his family, for being supportive,

To my bud for making this journey a memorable trip.

ACKNOWLEDGMENTS

I would like to take this opportunity to thank the people behind the success of this thesis.

Special thanks to Dr. Adam Doupé, my thesis chair and advisor for his unwavering support and his attention to detail. I sincerely appreciate your counsel and cherish the time that you spent with me. Thank you for being a constant motivator by making sure that I strive for excellence at all the things I do.

I would like to thank Dr. Partha Dasgupta for the meaningful discussions that we had and for helping me choose the right path during the formative stages of the thesis.

Finally, I would like to thank Dr. Gail Joon-Ahn, for being a part of the committee and for providing his input to make the thesis more impactful.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Internet Of Things (IoT) is an upcoming technology which has gained traction in the recent years due to rapid advancements in various areas such as embedded system design, sensors, hardware support, software protocols and communication infrastructures. IoT can be loosely referred to as a conglomerate of sensors, actuators and embedded chips spread out across various commodity appliances, locomotives, and devices such as computers, mobile phones and complex human-computer interfaces which communicate with each other to accomplish a specific task. A more formal definition from [35] defines IoT as follows:

"The Internet of things (IoT) is the network of physical devices, vehicles, and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to collect and exchange data."

A report by Gartner [30] reveals that there will be 8.4 billion IoT devices by the end of 2017. Further, Gartner predicts the number of IoT devices to breach the 20 billion mark by 2020. IoT plays a vital role in many fields such as healthcare, military, manufacturing, smart homes, and security etc. Of the many technologies that are available to assist inter-connectivity amongst the IoT devices, BLE is poised to be a key player in the IoT space due to its recent progress.

With the introduction of Bluetooth 5.0 specification, there is a sizable impact to be made by adopting the benefits provided by the new Bluetooth protocol stack for resource-constrained devices. More often than not, IoT devices are often held back by their limitation to computational power and energy efficiency. In order to alleviate this issue, BLE has become a major contestant for providing seamless solutions to integrate IoT devices across multiple industry verticals.

Careful measures must be taken when designing applications which partake in mission-critical operations. With rapid progression in the number of devices that partake in the IoT ecosystem, there is also a growing concern about the state of security for these IoT devices. A report by [10] disclosed that millions of IoT devices were left unprotected and exposed to unfettered malicious access by an adversary. Security is a serious liability when considering the current IoT ecosystem. In most cases, security takes a backseat for enabling smooth user experience. This, in turn, creates an incentive for a malicious adversary to target the IoT devices. In order to address the lack of security in current IoT ecosystem, this thesis talks about the looming threats that the IoT devices face and the possible countermeasures that could be deployed to mitigate the chance of a successful attack.

1.1 Scope of Thesis

The scope of the thesis is provided in the following section:

- A comprehensive background on the BLE technology that supports communication between the IoT devices.
- A systematic classification of IoT threats and possible countermeasures that could be adopted to thwart off the occurrence of a successful attack.
- Identify and provide a discussion on future research trends that hold great promise to contribute to the improvement of the current IoT ecosystem.

1.2 Document Structure

- Chapter 2 discusses the background on the BLE technology that facilitates inter-connectivity amongst the IoT devices.
- Chapter 3 describes the different kinds of threats and classification of those attacks in an IoT setting.
- Chapter 4 describes the different countermeasures for mitigating the occurrence of successful attacks.
- Chapter 5 provides a discussion on emerging research ideas and related work.
- Chapter 6 provides ending remarks and concludes this thesis.

Chapter 2

BACKGROUND

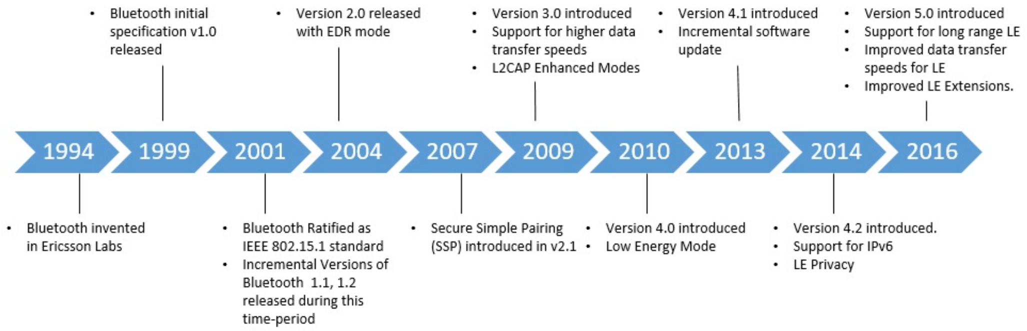
Bluetooth started as a research project set in motion by telecommunications giant Ericsson. It became a technology standard for enabling wireless devices to exchange data over short distances. It was used in the creation of Wireless Personal Area Networks (WPAN). A group of telecommunication vendors and manufacturers collaborated to form an organization called as the Bluetooth Special Interest Group(SIG). The SIG holds oversight on the Bluetooth protocol standards and also provides certification to device manufacturers. This chapter provides brief information on the progress of Bluetooth, the importance it holds in the current IoT ecosystem and finally the entire protocol stack to further understand its potential impact.

2.1 Bluetooth Specification History

The following section provides a brief history on the timeline for each bluetooth specification.

Timeline Snapshot Initial versions of Bluetooth specification v1.0, v1.1 and v1.2 had incremental improvements with respect to hardware and software changes. Bluetooth versions 1.1 and 1.2 were later ratified as the IEEE 802.15.1 standard in 2002 and 2005 respectively. Features such as Host Control Interface, Received Signal Strength Indicator (RSSI), flow control and retransmission modes in L2CAP layer were introduced in these initial versions.

Bluetooth v2.0 specification was released in 2004. The new version provided a faster and better data transfer rate by leveraging the power of Gaussian frequency-



1994
1999
2001
2004
2007
2009
2010
2013
2014
2016
Bluetooth initial specification v1.0 released
Version 2.0 released with EDR mode
Version 3.0 introduced
Support for higher data transfer speeds
L2CAP Enhanced Modes
Version 4.1 introduced
Incremental software update
Version 5.0 introduced
Support for long range LE
Improved data transfer speeds for LE
Improved LE Extensions.
Bluetooth invented in Ericsson Labs
Bluetooth Ratified as IEEE 802.15.1 standard
Incremental Versions of Bluetooth 1.1, 1.2 released during this time-period
Secure Simple Pairing (SSP) introduced in v2.1
Version 4.0 introduced
Low Energy Mode
Version 4.2 introduced.
Support for IPv6
LE Privacy

Figure 2.1: Bluetooth Specification Timeline.

shift keying (GFSK) and Phase Shift Keying (PSK). Bluetooth v2.1 specification came out with a pairing model called Secure Simple Pairing in 2007.

Specification of Bluetooth v3.0 was adopted as the IEEE 802.15.1 standard in 2009. The specification mainly consisted of an alternative MAC/PHY feature which would be used for larger data transfers by switching to an 802.11 device. Other features such as Enhanced Reliable Transmission Modes (ERTM), enhanced power control and unicast connectionless data were also included.

Bluetooth 4.0 specification got adopted as the new standard in 2010. This specification introduced the Bluetooth Low Energy Mode. It provided support for establishing short range communication in resource-constrained devices. With the introduction of BLE, bluetooth devices were classified into the following modes:

- BR/EDR Mode
- Single Mode
- Dual Mode

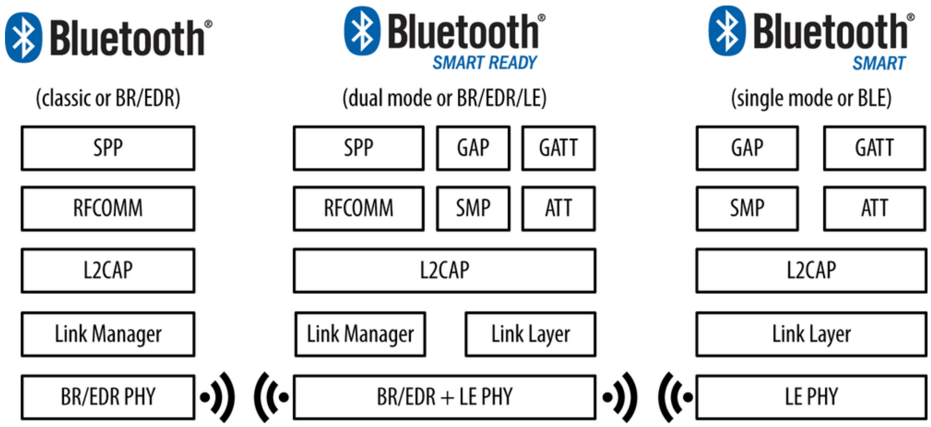


Figure 2.2: Bluetooth Device Modes [28].

BR/EDR Mode Devices that support BR/EDR Mode are usually called by the trademark name Bluetooth Classic. They support high data transfer rates and are usually supported by smartphones,laptops and computers.

Single Mode Devices that operate under single mode usually have the protocol stack of BLE. They are called by the trademark name Bluetooth Smart. They provide low latency, low power consumption at the cost of reduced data transfer rate. They are usually used for wearables and other sensor devices which have limitations on battery usage. Most of these devices can only communicate with devices which operate in the same protocol and cannot connect with Bluetooth Classic Devices.

Dual Mode Devices that support both BR/EDR and BLE protocol stacks operate in the dual mode. They are called by the trademark name Bluetooth Smart Ready. These devices have the ability to communicate with both Bluetooth Classic and Bluetooth Smart device.

In 2013, Bluetooth v4.1 was released. It was an incremental software update with no changes to hardware. It mainly provided co-existence support for LTE networks, improving the data exchange rates and providing flexibility to developers by enabling multiple roles for Bluetooth devices. Bluetooth v4.2 was released in 2014 with support for IPv6 and additional security mechanism for protecting privacy of the device in LE mode. Pairing mechanisms were updated to use Elliptic-Curve Diffie-Hellman (ECDH) to prevent MITM attacks during the pairing process of the devices.

Bluetooth 5.0 specification was announced in 2016. Features such as extended range, improved data transfer rates of upto 2 Mbps and LE channel advertising and selection filters were rolled out in the new specification. These features provided Bluetooth with the necessary arsenal to make it a serious contender for establishing communication between IoT devices.

2.2 Bluetooth Low Energy

BLE mode resonates with the energy requirements of IoT devices and is suitable for the limited computational power that is available in the IoT devices. It is of paramount importance to understand the components of the BLE protocol stack to understand the ramifications in the event of an attack. A detailed outlook on the bluetooth protocol stack is presented in the following section.

Figure 2.3 shows us the various components in the BLE protocol stack.

Controller

The controller consists of the components mentioned in the section below.

Physical Layer The physical layer mainly consists of the hardware components such as radio and communication circuits that assist in conversion of digital signals to RF waves. The BLE device has access to 37 data channels which it can hop for transmitting data packets. They have access to 3 different advertising channels indexed at 37, 38 and 39 which are used to transmit advertising packets. The hop

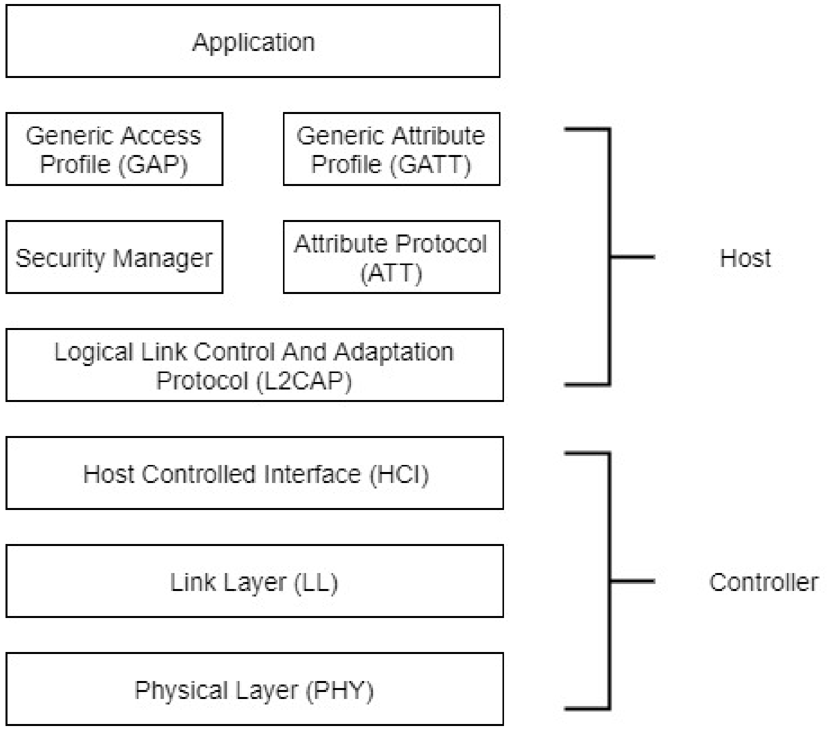

Application
Generic Access Profile (GAP)
Generic Attribute Profile (GATT)
Security Manager
Attribute Protocol (ATT)
Logical Link Control And Adaptation Protocol (L2CAP)
Host
Host Controlled Interface (HCI)
Link Layer (LL)
Physical Layer (PHY)
Controller

Figure 2.3: BLE Protocol Stack.

values are known to the device during connection establishment and varies for every other subsequent connection. The next channel to hop is calculated using the formula:

$$channel = (\ curr\_channel + hop\ )\ mod\ 37$$

Link Layer The Link Layer interfaces with the Physical layer and takes care of all the heavy lifting to be done in order to keep in compliance with the timing requirements. This is accomplished with the help of real-time hardware and software which comes in-built by the device manufacturer. The main responsibilities of this layer is to support encryption, maintain link state of the radio, perform data whitening and to assist in CRC generation and verification.

Link layer also defines roles that a device can have which is shown in Table 2.1. Another logical way of grouping these roles are as follows:

Role	**Description**
Advertiser	Device that advertises packets
Master	Device that initiates a connection
Scanner	Device that scans for advertisement packets
Slave	Device that accepts a connection

Table 2.1: Link Layer Roles.

- Active connection - Master and Slave
- Inactive connection - Advertiser and Scanner

Host Controller Interface “ The Host Controller Interface is a standard protocol that allows for the communication between a host and a controller to take place across a serial interface ” [28]. The HCI is used for interfacing the Host with the Controller using components such as UART, USB or SDIO.

Host

The Host is comprised of the components that are mentioned in the following section.

L2CAP The Logical Link Control and Adaptation Protocol plays an important role in the BLE stack. It is responsible for breaking down a large packet of information from upper layers into manageable BLE-sized chunks. The chunks are usually 27 bytes long. L2CAP also helps in combining the smaller BLE packets into a single large data packet that can be sent to the upper layers. This process is known as fragmentation and recombination. L2CAP services the ATT and SMP protocol thereby acting as a protocol multiplexer.

ATT The Attribute protocol acts a client/server protocol. It is stateless and generally uncompromising with respect to sequencing of requests. The attribute protocol is responsible for the following operations:

- Error Handling
- Server Configuration
- Finding Information
- Read/Write Operations
- Asynchronous Server Intitiated Operations

Security Manager Security Manager acts both as a security algorithm as well as a protocol. The responsibility of this layer is to assist the device in generating and exchanging security keys to establish an encrypted communication channel. Security Manager defines two roles namely: Initiator and Responder which corresponds to the Link Layer Master and Slave respectively. Security Manager partakes in the following security procedures:

- Pairing —A procedure where a temporary key is exchanged to switch to a secure encrypted link.
- Bonding —A procedure where a sequence of key exchanges occur to generate a long term key that is to be used for setting up a secure encrypted channel.
- Encryption Re-establishment —A procedure that is triggered when both participants of the connection have encryption keys stored. It is used to set up an encrypted connection without undergoing pairing or bonding procedure.

Security Manager contains a list of security keys that are used for establishing a secure connection. The keys and their usage is listed in Table 2.2

Keys	**Distributor Usage**	**Acceptor Usage**
LTK, EDIV, Rand (Encryption)	Used to encrypt the link when a slave	Used to encrypt the link when a master
IRK, BD_ADDR (Privacy)	Used to generate resolvable private addresses	Used to resolve resolvable privaSM Key Usagete addresses
CSRK (Signing)	Used to sign data	Used to verify signatures

Table 2.2: Security Manager Key Usage [28].

Pairing Algorithms The Security Manager has the following pairing algorithms at its disposal for establishing a connection.

Legacy Pairing Key exchange occurs in unencrypted plain text format. Devices in communication should enter the same PIN key to establish a connection. Usually the PIN is pre-determined by the manufacturer. This method of pairing is opted for achieving interoperability with Bluetooth 2.0 devices and its earlier versions.

Secure Simple Pairing Secure Simple Pairing came into existence from v2.1 Bluetooth Specification. Support for many pairing algorithms were added in this incremental update. The list below describes the various pairing mechanisms that are available.

- Just Works —No user interaction is enforced in this type of pairing. This pairing mechanism is used when one of the device has no input or output capability. This pairing mechanism is susceptible to MITM attacks.

- Passkey Entry —User interaction maybe required in this type of pairing. This pairing mechanism is used when only one of the device has input capability and

GAP Role	**Link Layer Role**	**Description**
Broadcaster	Advertiser	Device that sends advertising data packets
Central	Master	Device that listens for advertising packets and intitates connection
Observer	Scanner	Device that listens for advertising packets
Peripheral	Slave	Device that sends advertising data packets and waits for connection requests

Table 2.3: GAP Role Relation to Link Layer Role.

the other device only has output capability. This pairing mechanism is also susceptible to MITM attacks.

- Numeric Comparison —This type of pairing enforces user interaction. This pairing mechanism is used when both devices have the ability to display six digit number and the user has the ability to enter yes/no. This pairing mechanism provides protection against MITM attacks.

- Out Of Band —Uses another channel of communication such as Near Field Communication (NFC), RFID tags, biometrics etc. This pairing mechanism provides protection against MITM attacks.

GAP Generic Access Profile is the control layer that dictates how procedures such as security establishment, connection, device discovery and data exchange occur between different vendor devices.

Mode	**Role(s)**	**Applicable Peer Procedure(s)**
Broadcast	Broadcaster	Observation
Non-discoverable	Peripheral	N/A
Limited discoverable	Peripheral	Limited and General discovery
General discoverable	Peripheral	General discovery
Non-connectable	Peripheral, broadcaster, observer	N/A
Any connectable	Peripheral	Any connection establishment

Table 2.4: Gap modes and applicable procedures [28].

Procedure	**Role(s)**	**Applicable Peer Mode(s)**
Observation	Observer	Broadcast
Limited discovery	Central	Limited discoverable
General discovery	Central	Limited and General discoverable
Name discovery	Peripheral, central	N/A
Any connection establishment	Central	Any connectable
Connection parameter update	Peripheral, central	N/A
Terminate connection	Peripheral, central	N/A

Table 2.5: Gap procedures and their required modes [28].

Table 2.4 shows the various GAP Roles that are allowed to operate in a particular mode and their respective applicable peer procedures.

Table 2.5 shows the various GAP roles that are allowed to perform certain procedures and their respective applicable peer modes.

GATT The Generic Attribute Profile defines the ways through which the profile and user data can be exchanged over a BLE connection. GATT uses the ATT Protocol for transporting data between devices. There are two roles defined in the GATT protocol which is GATT client and GATT server. The roles are independent of the GAP roles. Data is organized into sections called services. Services group related pieces of user data called as characteristics.

Application Layer The Application Layer is the final layer which interfaces the GATT layer with the application data. This data is then passed down to the respective GATT services and characteristics. Proper protection measures have to be put in place to prevent data exposure. The application layer offers some sort of end-to-end encryption to prevent unexpected data exposure.

In this chapter, we have seen in detail about the Bluetooth protocol stack; it's corresponding components and the functionality of each component. Now that we have an understanding of how the BLE stack operates, we can proceed to identify the various attack surface vectors and the corresponding threats in the next chapter.

Chapter 3

THREATS

3.1 Overview

BLE based IoT devices are more often than not susceptible to threats because of insufficient end-to-end security. A targeted attack from an adversary could cripple the functioning of a system dependent on these devices. In this chapter, an overview about the different types of threats that exist in the IoT space is visited. A classification of those threats meaningful to the subject IoT devices is described.

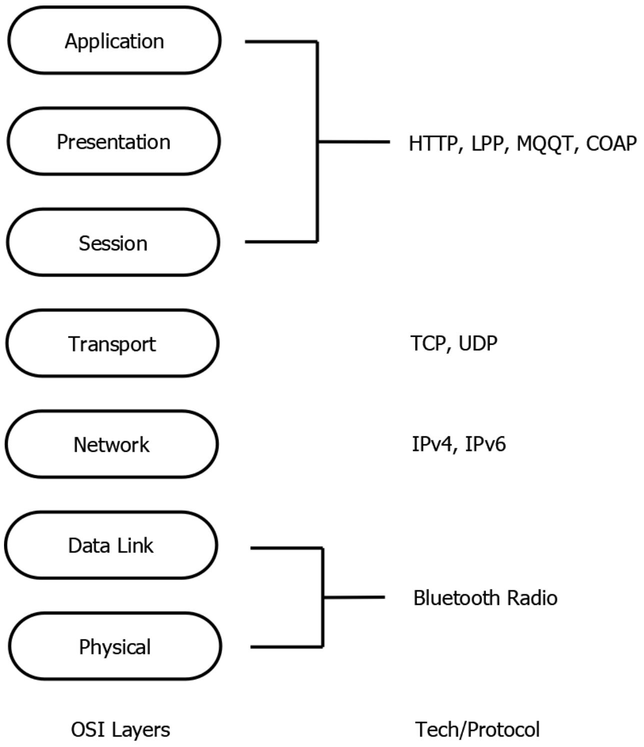


Figure 3.1: An Illustration of the OSI Layers Relevant to IoT Devices

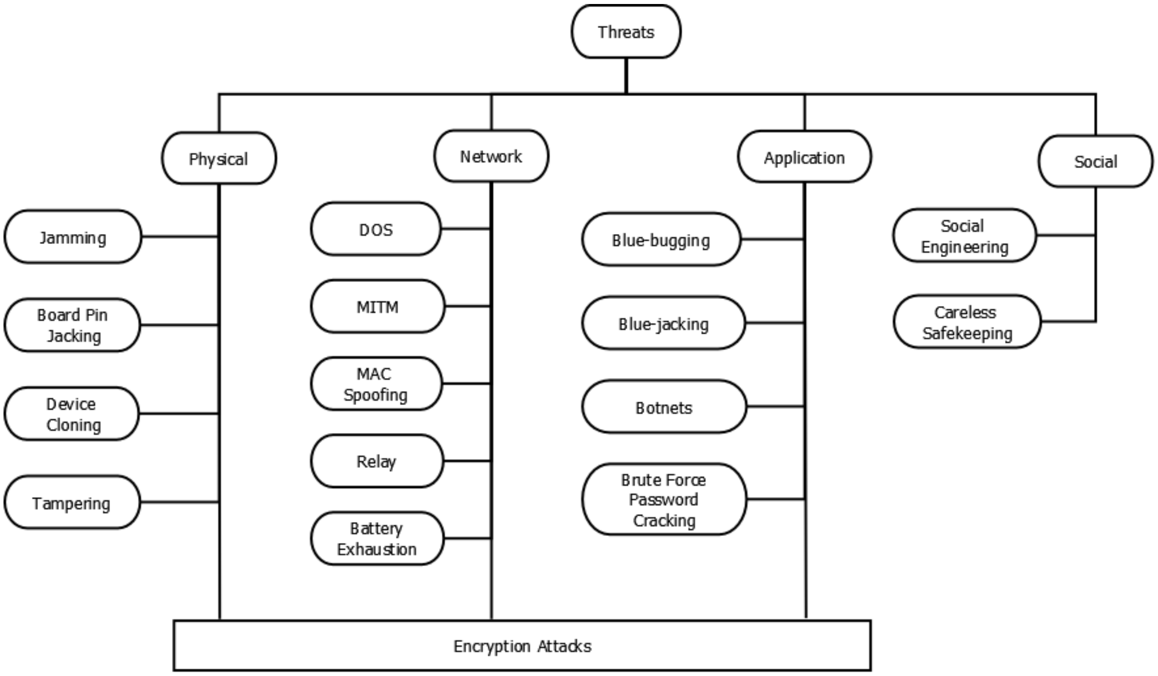


Figure 3.2: Classification of BLE Threats

3.2 Classification

This section covers the classification of threats found in BLE based IoT devices. The classification is based on the BLE layers upon which a potential adversary could target to perform an attack. The threats are classified as follows:

- Physical Attacks
- Network Attacks
- Application Attacks
- Social Attacks
- Encryption Attacks

3.2.1 Physical Attacks

This subsection of threats describe the attacks that occur on the physical layer of the IoT device.

Jamming

Jamming attacks pose a serious threat to IoT devices by disrupting their ability to communicate with other devices which utilizes the ISM 2.4 GHz spectrum. A targeted disruption in this channel will render systems dependent upon the communication and coordination of these IoT devices pointless.

Jamming attacks achieve their intended purpose by intentionally emitting radio signals on communication channels used by the IoT devices such that the Signal-to-Noise Ratio is skewed to disrupt the reception of messages between the devices. Broadly speaking , jamming attacks could either be Wide-band or Pulse-band. The former targets the entire RF spectrum that the IoT devices use, while the latter targets specific channels which are used by the IoT devices. Variants of the Pulse-band jamming attacks are discussed in the section below.

Constant Jamming Constant jamming attack continuously emits a jamming signal to disrupt the communication channels between the IoT devices. Although effective in accomplishing it's desired purpose, it is highly energy inefficient because of the continuous dispersion of the jamming signal. This jamming attack is easily detected due to its constant disruption over IoT communication channels.

Deceptive Jamming Deceptive jamming attack emits signals at periodic intervals to disrupt the communication channels. This type of attack is energy efficient because the jamming frame is sent at periodic instances. Therefore disruption experienced is intermittent and not continuous. Since the attack is periodic it is harder to detect when compared to a constant jamming attack.

Random Jamming Random jamming involves a combination of both Constant and Deceptive jamming. At any given instance of time the jamming method used could be random. Since no particular pattern is involved in this attack it makes it harder to detect. Energy efficiency averages between the former two jamming attacks.

Reactive Jamming Reactive jamming kicks off only when the jammer senses ongoing network activity over the target channel. Channel interruption by the adversary happens when any activity is sniffed by the jammer. The activity could be as simple as an interrupt such as a frame delimiter or looking out for change in the RSSI threshold values. If any of these conditions occur, it triggers the jammer to perform channel disruption. Since the attack is very specific it is much harder to detect and generally energy efficient than the other variants of jamming attacks.

Device Cloning

This type of attack involves in creating an exact replica with capabilities and functionality similar to the target device. Although not widespread, this type of attack can occur provided the adversary has physical access to the subject device of interest. Cloning could help the adversary into tricking the system to divulge sensitive information.

Board PIN jacking

This type of attack makes the assumption that the adversary has access to the physical hardware of the device. Certain devices hold critical information in their non-volatile memory which could be accessed through PINs on the Printed Circuit Board (PCB) which are left open. Using the standard input/output PIN on the PCB of the target device as the access point, an adversary could extract sensitive information. The gathered information can then be used to perform various nefarious actions which could aid the adversary in accomplishing its end goals.

Tampering

This attack involves modification of the target device in order to assist the adversary with critical information which will lead to an exploit. Examples of this type of attack is the modification of the point of sale terminal to capture the RFID tags which are then relayed to the adversary through covert communication channels. Tampering attacks requires the adversary to perform physical modifications to the target device to assist in subterfuge of the network of IoT devices.

3.2.2 Network Attacks

This subsection of threats describe the attacks that occur on the network layer of the IoT device.

Denial Of Service

DoS attacks are a subset of network attacks where the adversary tries to disrupt service to the target network. This attack achieves its purpose by flooding the connections with malformed connection or service requests. This flooding of requests hogs the entire bandwidth of a particular servicing node in the target network. This will cause the disruption of service to genuine incoming requests from various clients.

Battery Exhaustion

Generally IoT devices are energy efficient and are made to run on the energy derived from a single battery operated Li-ion coin cell. Battery exhaustion is another form of exploit, which the adversary has in it's arsenal to wreck havoc on the services rendered by a system which is predominantly dependent on the coordination and functioning of many devices which are located at inaccessible spots. IoT devices which operate using BLE are specifically designed to be awake for a short period of time for conservation of energy. When the target BLE device is bombarded with requests which actively create a situation in which the device is prevented from reaching a sleep state, the life expectancy of the device significantly reduces due to persistent awake state of the device. Uher *et al.* [29] show that a single software defined radio attack platform under the control of an adversary can drastically bring down the life of a BLE device through coordinated connection attempts from an army of adversarial devices.

Man-in-the-Middle

MITM attacks usually occur in the form of eavesdropping wherein the adversary would be able to sniff on the traffic between the IoT devices. Secure Simple Pairing protocol between the IoT devices can be forced to adopt the Just Works mode by manipulating the input/output capabilities of the devices. Since Just Works SSP does not provide protection from eavesdropping and MITM attacks, the adversary can then proceed to use the information collected from the devices to orchestrate a targeted attack.

MAC Spoofing

Mostly recognized as the evil twin attack, this type of attack deliberately tries to spoof the MAC address of a target device to misdirect a system dependent on the information from the target device. An example of such attack is the MAC spoofing of BLE sensor beacons which communicate with a central server to provide information about its location. Spoofing attacks on such BLE beacons tend to corrupt the data sent to the server thereby causing the results to be inaccurate as shown in the study of William *et al* [19].

Relay

Relay attacks usually depend on the adversary's capability to re-transmit messages from the originator to the target device. An example of this attack in the context of IoT devices is the relay of e-key from key fobs/mobile phones to smart locks.

3.2.3 Application Attacks

This subsection of threats describe the attacks that occur on the application layer of the IoT device.

Password Cracking

Password Cracking can be done through brute force attacks which can be performed online or offline. An offline brute-force password cracking attack is much more dangerous due to unrestricted attempts by the adversary to decipher the password to obtain sensitive information. Many variants of this attack such as chosen plaintext, dictionary, known ciphertext help to achieve the desired results considerably faster than brute force attacks.

Bluejacking

This type of attack usually occurs in Bluetooth enabled devices such as mobile phones and laptops. Adversary initiates the attack by sending unsolicited information to the target user. This attack helps the adversary in performing actions such as adding new contacts to the address book which is akin to mobile phishing and spam attacks. Some of these attacks could elicit a vile response when the target user responds to the specific messages sent by the adversary.

Bluebugging

This type of attack relies on a security flaw that is found on older Bluetooth devices which enable the execution of commands unbeknownst to the user of the target device. The execution of such covert commands will enable an adversary to start a phone call, eavesdrop on messages, track the location of the user and other services which are available on the compromised device.

Botnets

A botnet is a collection of compromised computers, devices and other internet connected devices often referred to as zombies. These devices are infected with malware which enables an adversary, who owns the botnet, to control the devices by means of a covert channel such as Internet Relay Chat (IRC). They can issue commands to perform malicious activities such as distributed denial-of-service (DDoS) attacks, the sending of spam mail, and information theft.

On October 2016, Mirai botnets were used to bring down the service of popular websites such as Netflix, Reddit and GitHub for several hours. Mirai botnet leveraged the weak security in DVRs, webcams and many other IoT devices to create a formidable army of zombies ready to disrupt targets. Variants of the Mirai botnet, namely Hajime emerged in late 2016. Hajime botnet is an improved version wherin it has a decentralized architecture and utilizes the torrent protocols for discovery and infection. The botnets leverage weaknesses in application security such as default device credentials, port exposure and spread the malware across easy targets.

3.2.4 Social Attacks

This subsection of threats describe the attacks that occur by means of social constructs.

Social Engineering

Humans, as stated by security literature, are the weakest link in preserving the security of a system. An adversary could employ subtle psychological ploys to gauge and extract information which might aid in performing a successful exploit. Social Engineering is akin to phishing or spam e-mail when taken in the context of a conversation with human beings. An adversary equipped with a certain understanding of human behavior could plot complex questionnaires which will assist in the disclosure of sensitive information which could later be used for nefarious purposes.

Careless Safekeeping

This type of attack encompasses situations wherein the passwords or sensitive information are not properly stored. Improper safekeeping of sensitive information and exposure of such information will often lead to situations wherein the adversary could learn more about the target. Though this type of attack seems trivial, many security breaches have occurred due to ignorance of best practices within an organization.

3.2.5 Encryption Attacks

Encryption attacks are unique in the fact that they are not tied to one particular layer. This type of attack could either target a single layer or a combination of layers.

The layers which are subject to these kinds of attacks are the Physical, Network and Application layers. Encryption attacks mainly try to decipher the encrypted data by sophisticated means or through plain brute force methods. When no proper end-to-end encryption is provided the data exchanged between devices is disclosed to potential adversaries located around those devices. Common encryption attacks are known plaintext, chosen ciphertext or cryptanalysis attacks.

Chapter 4

COUNTERMEASURES

This chapter describes the various countermeasures that are currently available to help prevent the occurrence of a successful attack by an adversary.

4.1 IDS

An intrusion detection system refers to a standalone device or a software residing on a network of devices that actively monitors traffic information for any malicious activity or violations of security policy [36].

Detection Methods IDS aid in the detection of attacks by the following detection methods:

Signature based detection The IDS systems compares pre-determined and known attack patterns with suspicious network packets to detect the presence of an attack.

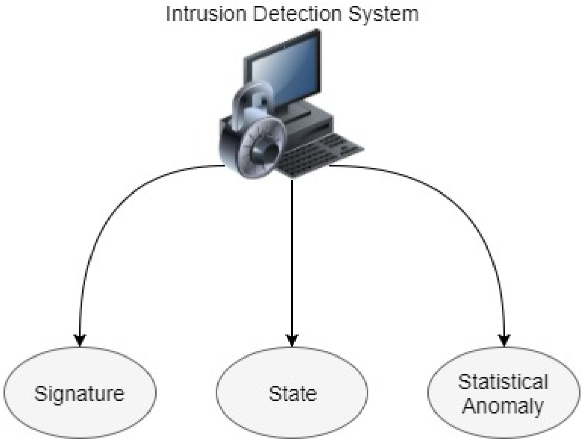


Figure 4.1: IDS Detection Methods

State based detection The IDS system has a pre-compiled list or a specification of profiles which are deemed to be benign for a set of actions or an event. A mismatch between a current behavior and the specification will be reported as an attack.

Statistical anomaly based detection This method of detection relies on the presence of a baseline against which the IDS can compare and monitor the network traffic. Whenever a part of the network crosses this threshold for observed behavior the IDS jumps in to detect and perform appropriate actions to mitigate an attack.

4.1.1 Classification of IDS

This subsection talks about the different classification of IDS based on their architecture.

Standalone IDS architecture This IDS architecture comprises of an IDS agent which runs independently on each node in the network. These standalone agents help to detect intrusions independently without coordination or communication with other IDS agents. This IDS architecture would work best for flat network infrastructure than for multi-layered network infrastructure because of the said absence of data exchanges between the IDS agents.

Collaborative IDS architecture This IDS architecture comprises of IDS agents running on each node which collectively participate in intrusion detection and response. Each IDS agent in the network must participate in the coordination and communication of events which are local as well as global to the network it resides in. This helps in orchestrating mitigation of any intrusion detected in the network. Collaborative IDS agents fit well in almost any network structure because of their clout for coordination and communication amongst a network of nodes.

Hierarchial IDS architecture This IDS architecture comprises of IDS agents which span across clusters in multi-layered network infrastructures. Each cluster in the network will have a cluster head which will act similar to that of common control points such as routers, switches and gateways. The cluster head will have the IDS agent which helps in performing the aforementioned collaborative tasks to detect presence of an intrusion. Hierarcial IDS agents find merit in multi-layered network infrastructures which are separated by geographical boundaries. These IDS agents collaborate globally and mitigate intrusion locally.

Mobile IDS architecture This IDS architecture comprises of IDS agents which are mobile and not constrained to a particular location. The IDS agents running on these mobile devices help in aggregating data and reporting the information to cluster heads which will then perform collaborative tasks to detect and mitigate intrusion.

4.1.2 Existing IDS systems

The following subsection talks about the existing state of the art IDS systems that are already available for intrusion detection and prevention [3], [12], [23].

Bayesian IDS This type of IDS system uses a game theoretic approach which depends on a predetermined set of beliefs that the IDS agent will have in order to make decisions on intrusion detection and prevention. The game formulation is between a pair of attacking and defending nodes in the network. The aim of the game is to maximize the payoff on the network of nodes by having predetermined beliefs on whether a particular node in the network is benign or malicious. The beliefs can be static wherein they represent a constant value for a particular set of actions or event upon which the IDS agent is entrusted to decide and react. The beliefs can be dynamic wherein the values are updated to reflect the environment. The dynamic Bayesian IDS agents fare much better than the static Bayesian IDS agents because of their flexibility and its capability to reflect and represent real-world situations.

Classification-Based IDS This type of IDS system uses a set of supervised classification algorithms to assist in intrusion detection and prevention. Common examples of classification algorithms used in these IDS systems are the Multi-Layer Perceptron, Linear Classifier, Gaussian Mixture Model, Naive Bayes and Support Vector Machines. A number of studies were made by comparing all these algorithms against a set of known attacks. Of these algorithms, the Support Vector Machine model seems to be the most accurate classification algorithm for all the attacks that were chosen.[3]

Zone-Based IDS This type of IDS system relies on the technique of partitioning the network into inter and intra nodes. The network is divided into zones in order to save the bandwidth for specific sub-networks. Any node found within a particular network is considered to be an intra-node. A local IDS agent will be aggregating and collecting reports on that local zone to help the IDS system in taking appropriate decisions with respect to intrusion detection and prevention. A node which acts as a gateway to nodes in other zones is considered to be an inter-node. These nodes use some sort of aggregation algorithms to present information of their particular zone to other zonal IDS agents. It was found out that the local anomaly detection model worked well for low mobility environment than high mobility environment.

Specification-Based IDS This type of IDS system uses a state-based approach to detect the occurrence of malicious behavior within a network. The IDS system relies on the fact that each node in the network is being tracked by a network monitor. The

IDS system then analyzes the network traffic of the nodes to determine if there are any discrepancies in the routing behavior by comparing the analyzed traffic with a finite state machine model as an operative specification. This method is not pragmatic because of mobile nodes which may or may not be covered by the network monitor.

Fuzzy Logic IDS This type of IDS system uses a computational paradigm that helps in making reasonable decisions about uncertainty. Parameters such as the number of packets dropped, DestThreshold and SourceThreshold are fed into a mathematical model to help local IDS agents predict the appropriate routing behavior. This IDS system mostly finds its application in medical diagnosis because of its capability to handle imprecise information and uncertainty.

Cross-Layered IDS This type of IDS system relies not only on the network layer of a particular system but also relies on parameters that are obtained from the MAC and the physical layer. Since the information available to the IDS agent is quite comprehensive, this approach has better intrusion detection accuracy when compared to other intrusion detection methods. A trade-off must be made for the overhead it has on the network for the perceived accuracy.

4.2 Security Policy

The following section discusses the various components of security policy that could be adopted to prevent the occurrence of a successful attack.

Blacklist A blacklist is a table or a list of nodes which are identified to be malicious within a network. This list could be updated dynamically to reflect upon the threats which the network might be exposed to. Broadcasting and updating this list across many routers and gateways could help prevent the intrusion of threats across the network.

Whitelist A whitelist is complementary to a blacklist. It is a list of verified devices in a network which is deemed to be safe to interact. Whitelists help the routers and gateways in a network to make decisions on allowing data or request packets to flow through the network.

Resolvable Private Addresses RPAs are a solution to the most prevalent threat of sniffing and spoofing static public device addresses in mobile ad-hoc networks. BLE 4.2 has a mechanism to create an RPA by means of an Identity Resolving Key (IRK) or a Peer Identity Resolving Key (IRP). Using either of these two keys RPAs can be generated to establish a connection with master or slave devices. When a connection request is made to a particular device, the connection will succeed only if the initiating device finds out the RPA for the target device by using the correct IRK. Using link layer device filtering, the RPA can be resolved and authenticated at the link layer. If not, the connection request will fail and the device wouldn't be susceptible to battery exhaustion or spoofing attacks.

Link Layer Device Filtering Proper link layer device filtering has to be set up to prevent unwanted malicious nodes from penetrating the network. Measures should be taken to make sure that none of the devices in the network use static public addresses unless otherwise is absolutely required and essentially harmless to the network. A connection request will be allowed only if the RPA can be resolved and found in the whitelist of the target device. This rule should be followed across many roles which a BLE based IoT device might partake such as Advertiser, Scanner, and Initiator. Only devices which have their RPAs on the whitelist should be processed [29].

4.3 Multi-factor Authentication

Multi-factor Authentication refers to a method of access control by which an user requesting access to a system presents multiple pieces of information or evidence to prove the user's authenticity. Factors that could be considered as evidence for proving one's authenticity are usually of the following three types:

- Type I Something that the user has.
- Type II Something the user knows.
- Type III Something the user is.

Examples of Type I could be possession of physical objects such as key fobs, physical access cards, USB stick or a QR code [18]. Examples of Type II could be knowledge of secrets such as PIN, time based one-time token or a simple alphanumeric 8-16 digit secret password. Examples of Type III could be a characteristic of the user which usually is related to bio-metric factors such as fingerprint, voice or facial recognition.

Care must be taken to ensure that the BLE based IoT devices enforce reliance upon multiple factors of authentication rather than depending on a single factor of authentication. This will help mitigate the successful penetration of attacks such as brute force or password cracking because of the multiple factors involved in authentication. It is important to note that thought the attacks are mitigated by multi-factor authentication, breaches could occur when the attack involves social engineering.

4.4 Security Tools

This section describes about the different suite of pen-testing tools that are readily available to help secure the attack surface vectors that may possibly be left exposed to an adversary.

Ubertooth Ubertooth is an opensource Bluetooth test tool for experimentation developed by Michael Ossmann and Dominic Spill. Ubertooth provides the hardware platform necessary to perform passive monitoring of communication between bluetooth devices. The first version Ubertooth Zero was released in October of 2010 which was then superseded by Ubertooth One which was released in January 2011. Ubertooth provides basic sniffing capabilities of the communication that happens between BLE devices.

Wireshark Wireshark [38] is a network protocol analyzer which is widely used across commercial, governmental, educational and non-profit enterprises to understand both at a macroscopic and microscopic level the proceedings of their network. Wireshark provides a wide variety of tools to help analyze communication that happens over a wireless or wired network. Wireshark provides rich features such as capturing data from various mediums such as Ethernet, IEEE 802.11, Bluetooth, USB and many other supported communication channels. Wireshark has the capability to support decryption for many protocols such as SSL/TLS, IPsec, SNMP, WEP and WPA. Wireshark has the ability to output the captured communication packets into formats such as CSV, PCAP, XML or simple plain text. The features provided by wireshark makes it an indispensable tool for performing all kinds of security testing.

crackle crackle [24] is a decryption tool that cracks BLE encryption. crackle capitalizes on a flaw that occurs during the pairing process between two BLE devices and will be able to obtain the Temporary Key (TK) from the packets captured during the initial pairing process. Once the TK is obtained, crackle can help decrypt the Short Term Key (STK) and Long Term Key (LTK) thereby decrypting the entire communication that happens over BLE channel. It has two modes of operation namely:

- Crack TK
- Decrypt with LTK

Crack TK This mode requires a PCAP file which contains initial pairing conversations between BLE devices. This PCAP file can be obtained by using Ubertooth as the hardware sniffer and Wireshark as the software platform to capture the sniffed data into PCAP format which is understandable by the crackle program. crackle exploits the fact that the initial pairing process mostly uses Just Works mode in most of the BLE devices which usually contains a 6-digit PIN ranging from 0 to 999999 padded to 128 bits. To analyze each connection in the input PCAP file and output the results to stdout.

```
crackle -i <file.pcap>
```

To decrypt the contents of the PCAP file use the following command:

```
crackle -i <file.pcap> -o <out.pcap>
```

The output file will contain all the decrypted packets from the original PCAP file.

Decrypt with LTK This mode requires the PCAP file to contain LL_ENC_REQ and LL_ENC_RSP and the LTK used to encrypt the communications between the two BLE devices. To check if the PCAP file contains necessary information it requires to perform decryption provide the following command:

```
crackle -i <file.pcap> -l<ltk>
```

To output the contents of the entire PCAP file into target output file using the following command:

```
crackle −i <file.pcap> −o <target.pcap> −l <ltk>
```

The output file will contain all the information of the encrypted packets in a human readable format.

4.5 Best Practices

This section talks about the best practices that a user should follow when using a BLE-based IOT device.

Proper Bluetooth Usage Steps must be taken to ensure that BLE remains at appropriate sleep states without actively engaging in the continuous broadcast of an advertisement. BLE device should resume communication with another device only if it knows for certain that the authenticity of the device is verifiable by multi-factor authentication. In addition to that, communication should not happen with another device unless it is part of a whitelist. Bluetooth devices operate in any of the following security modes:

- Security Mode 1 — Promiscuous mode which accepts any connection with no encryption or authentication. Provides no protection.
- Security Mode 2 — Authentication and encryption supported at service level which means that security will be added after pairing occurs.
- Security Mode 3 — Similar to Security Mode2 except that the security functions are supported at the link level. This mode provides security before a physical link is created.
- Security Mode 4 — This mode has Secure Simple Pairing protocol which provides resistance against MITM attacks by utilizing Elliptic Curve Diffie-Hellman based key exchange.

Any pairing process must enforce Security Mode 4 as it provides highest level of security for link layer even before link establishment. Because ECDH is used, MITM attacks are thwarted off.

Dissociate Lost or stolen Bluetooth devices Any device that is suspected to have been lost or stolen must be dissociated from the BLE device's whitelist. Any further pairing with that device requires careful attention to it's claims of authenticity. Users should never accept transmissions from unknown or unverifiable devices.

Secure Pairing Pairing process is the weakest link in most of the BLE based IoT devices. If possible, the pairing process should occur in a secure environment which the user has considerable amount of control. For cases where the nature of the environment is uncertain, user must make sure to enforce appropriate application and link layer safety measures mentioned above.

Disabling Unused SDIO pins BLE devices that operate in mission-critical systems should make sure that their PCB shouldn't have any SDIO pins left exposed for an adversary to exploit. Any unused board multiplexer or RFCOMM channel should be covered up.

Intrusion Detection System A proper IDS can help thwart off potential threats and rectify malicious nodes in the network by analyzing the traffic flow in the network. Updating firewall rules, frequent update and broadcast of whitelists and blacklists throughout the entire network will help mitigate the occurrence of a breach by an unauthorized device.

OOB channels Pairing process can be forced to take place in OOB communication channels such as NFC to facilitate secure transfer of session keys. Mandating the initial pairing process over OOB channel will most likely thwart off potential threats.

A summary of the different threats and their countermeasures is described in Table 4.1.

Layer	Threats	Confidentiality	Integrity	Availability	Impact	Countermeasures
Physical	Jamming [7], [37]			x	●	FHSS/DHSS, Directional Transmission
	Board PIN Jacking [5]	x	x		◑	Securing PCB SDIO PIN [5]
	Device Cloning [14]		x		◑	Enforcing Physical Security
	Tampering [14]		x		◑	Surveillance
Network	DoS [4]			x	●	Firewall, IDS, Honeypots [3], [12]
	MITM [11], [17]	x	x		◑	End-to-end encryption [1], [2], [6]
	MAC Spoofing [19]	x	x		◑	User authentication checks [4]
	Relay [4]	x	x		◑	Integrity Checks [14]
	Battery Exhaustion [4]			x	●	Whitelist authorized devices, use RPA [13]
Application	Brute Force Password cracking [14]	x	x		◑	Multi-factor authentication [1]
	Blue bugging [14]	x	x		○	Use Bluetooth 4.0+
	Blue jacking [14]	x	x		○	Use Bluetooth 4.0+
	Botnets [16]			x	●	Perform Malware Analysis [26]
Social	Social Engineering [14]	x	x		◑	Follow Best Practices [14]
	Careless safekeeping [14]	x	x		◑	Follow Best Practices [1]

○ Low ◑ Moderate ● High

Table 4.1: Snapshot —Threats and Countermeasures for BLE-Based IoT Devices

Chapter 5

DISCUSSION

This chapter provides an analysis of the issues that hinder the progress of BLE-based IoT devices. A discussion on emerging research trends that address these issues is also provided in this chapter.

5.1 Analysis

With the proliferation of IoT devices in many fields, there is a real need to address the issues that impede their progress. Usually, IoT devices are built with functionality and user experience in mind, thereby ensuring that the users have quick turnaround times for each interaction. In order to improve usability and user experience, security trade-offs have to be made because the BLE-based IoT devices are not designed to support heavyweight and computationally intensive security protocols. Attack surfaces emerge when an imbalance occurs in the trade-off between security and user experience. A retrospective of issues plaguing the current IoT ecosystem is analyzed. In addition to that, research ideas that could help resolve these grievances are also addressed in this section.

Metadata exposure Metadata is information that is used to provide additional information about other data. Especially of importance in the context of BLE-based IoT devices are metadata information of BLE packets such as source address, destination address and MAC address. As discussed in section 3.2.2, the root cause for a subset of network attacks such as MAC spoofing and relay attacks is metadata exposure. These attacks are perpetrated by the adversary by using metadata information to perform reconnaissance, modification, and misdirection of systems dependent on these BLE-based IoT devices. A classic example of this problem is the spoofing of BLE-beacons which provide location information. Misdirection or misinformation will cause a decrease in the accuracy of a system dependent on these beacons. Research work that targets ways to tackle the problem of metadata exposure will help improve the network security of BLE-based IoT devices.

Metadata exposure can be prevented by leveraging the power of Black Software Defined Networks. Black SDN operates on the premise that both metadata and payload are encrypted throughout the network. Chakrabarty *et al.* [8], [9] proposed a new format of BLE packets for use in Black SDN networks. Since the metadata has to be encrypted, proper routing mechanisms have to be implemented by a Trusted Third Party (TTP) device. In most cases, the TTP would be entrusted to be the responsibility of the Black SDN controller. It is important to note that network security is achieved at the cost of symmetric key management, decreased routing and payload efficiency. Research work that targets the improvement of the performance metrics of a Black SDN network such as better routing mechanisms, improved payload efficiency, and reduced overhead on network traffic may present promising solutions that could be integrated into BLE-based IoT systems.

Device Authenticity IoT networks may comprise of both static and dynamic nodes in the network. These nodes have to be monitored to prevent a malicious node from executing actions that cause service disruption. Attacks such as MAC spoofing from section 3.2.2 could be mitigated by verifying the authenticity claims of a device that is requesting access to the IoT network. Complex security protocols and tracking mechanisms are enforced in an IoT networks to verify the authenticity of a device. It is cumbersome for a single supervisory node in the network to keep track of the dynamic changes that happen over the entire IoT network. Research work that prioritizes novel methods to verify authenticity claims of a device find a crucial role in improving the overall security of the IoT ecosystem.

Blockchain [34] is an emerging technology that could verify the authenticity claims of a device entering the IoT network. Blockchain is an open decentralized ledger that holds public records of financial transactions which can be verified throughout the network. The advantage of the blockchain is the ability to provide proof of a legal transaction without depending on a verification response from a central authority. The transaction is deemed legal only when a majority of the participants in the blockchain agree upon a proof of computational work. Any new device that is trying to gain access to the IoT network must have reached a consensus by a majority of the participating nodes in the network which could vouch for the incoming device's authenticity. Research that targets the adaptation of blockchain to resource constrained BLE-based IoT networks could possibly improve the security of the IoT network.

Updating security test-beds The security test-beds have to be redesigned to account for the features that are relevant to IoT networks. Sachidananda *et al.* [26] proposed a IoT security test bed highlighting the features necessary to create a holistic approach towards IoT risk analysis and assessment. It is of paramount importance that security analysis be performed on both the hardware and the software of IoT devices. Attacks from section 3.2.1 and 3.2.5 can be addressed by strengthening the capabilities of IoT security test-beds. Research that targets penetration testing methodologies for IoT networks, context-based attack detection models and modular test-bed architectures could possibly help improve the overall security of an IoT network.

Security test-beds could also leverage the power of learning models available in artificial intelligence. An Intrusion Detection System could employ these learning models to identify patterns of network traffic to distinguish between benign and malign data packets. Network attacks from section 3.2.2 can be mitigated by conducting automated tests that are assisted by AI learning models. The learning models help the IDS to identify traffic patterns that occur in the network. With the assistance of such learning models, the IDS could play a major role in early detection and reduced turnaround time for mitigation of perceived threats.

Secure Pairing Pairing process is the weakest link in most of the BLE-based IoT devices. The current state-of-the-art devices do not enforce appropriate protection mechanisms which are available in the latest Bluetooth specification. Even with appropriate protection mechanisms available, the BLE device can be forced to choose a weak pairing mechanism by sending different control messages during the pairing process. In order to address this issue, innovative ways of pairing have to be explored. Attacks from section 3.2.3 can be mitigated by exploring new venues to establish secure pairing.

Delgado *et al.* [20] proposed a novel approach for the re-construction of secret key from stable Static Random Access Memory cells available in the BLE chip. The advantage of this novel approach is that a BLE device can leverage the intrinsically available Physically Unclonable Function to assist in the process of secret key generation and reconstruction without any modification to its physical hardware.

Body Sensor Networks find a unique position in helping these BLE-based IoT devices to establish pairing. Body Sensor Networks rely on the features present in most of the BLE-based IoT devices such as fingerprint sensors and touchpads. Using capacitive coupling, Hessar *et al.* [15] achieved data transfer rates of 50 bits per second through the human body by utilizing the capability of a body sensor network. An example use case is the transfer of digital keys from mobile phone to smart locks present on doors through the human body. Research that improves the data transfer rates achieved through body sensor networks could be a game changer in providing a secure way of transferring critical information amongst BLE-based IoT devices.

5.2 Related Work

Yasmin M. Amin and Amr T. Abdel-Hamid [4] provided a comprehensive and taxonomic analysis of PHY and MAC layer attacks that occur on 802.15.4 devices. The authors provided a classification relevant to Low Rate - Wireless Personal Area Networks (LR-WPANs) which is quite different from 802.15.1 devices (Bluetooth). A detailed explanation on the evaluation criteria and the reasoning for the classification of the attacks were provided. It is important to note that there are various sub categories in the 802.15.4 standard. Each subcategory has very specific application scenarios such as RFID, smart utility networks or industrial applications. The 802.15.4 devices differ from Bluetooth devices in that the former is constrained to the PHY and MAC layer of the OSI model while the latter utilizes additional layers present in the OSI model.

Albahar *et al.* [1] provided a survey on the bluetooth based MITM vulnerabilities. The authors mainly talked about improving the process of SSP with their idea - the Enhanced SSP (ESSP). The authors also presented general security guidelines when dealing with MITM vulnerabilities. Papers [12], [3] provide detailed information on the various Intrusion Detection Systems.

Hassan *et al.* [14] provided a comprehensive classification on Bluetooth attacks and the diffferent types of malware that are lurking out in the Internet. Their paper provided an illustration of Bluetooth attacks. Malwares that were discussed occurred mostly occur on Symbian OS. Andrea *et al.* [5] provided a threat classification for IoT devices based on four classes of attacks. Further based on those classification, the appropriate countermeasures were presented along with general framework on how the future IoT devices can be secured. Eyal Ronen and Adi Shamir [22] provided a new taxonomy on the IoT attacks based on how the attacker deviates a feature from its intended functionality. They also provided a case study of extended functionality attacks on smart lights.

In comparison, our work provides a classification of threats specific to BLE-based IoT devices. Countermeasures subject to threats of BLE-based IoT device were also discussed. Finally we provide the latest research findings, which are more specific to the target BLE-based IoT devices.

Chapter 6

CONCLUSION

We discussed in detail the background of BLE technology, explored the BLE protocol stack, and the functionality of its associated components. Further, we analyzed the common threats that BLE-based IoT devices face along with a meaningful classification of those threats. An in-depth analysis of the various countermeasures that are available to these IoT devices for the looming threats was also discussed. Finally, we discussed the research trends that hold great potential for improving the functionality and security of the BLE-based IoT devices.

REFERENCES

[1] Albahar, M. A., K. Haataja and P. Toivanen, "Bluetooth MITM Vulnerabilities: A Literature Review, Novel Attack Scenarios, Novel Countermeasures, and Lessons Learned.", International Journal on Information Technologies & Security **8**, 4 (2016).

[2] Albahar, M. A., K. Haataja and P. Toivanen, "Towards Enhancing Just Works Model in Bluetooth Pairing", International Journal on Information Technologies & Security **8**, 4 (2016).

[3] Alnaghes, M. and F. Gebali, "A Survey on Some Currently Existing Intrusion Detection Systems for Mobile Ad Hoc Networks", in "2nd International Conference on Electrical and Electronics Engineering, Clean Energy and Green Computing (EEECEGC)", (2015).

[4] Amin, Y. M. and A. T. Abdel-Hamid, "A Comprehensive Taxonomy and Analysis of IEEE 802.15.4 Attacks", Journal of Electrical and Computer Engineering **2016**, 4 (2016).

[5] Andrea, I., C. Chrysostomou and G. Hadjichristofi, "Internet of Things: Security Vulnerabilities and Challenges", in "Computers and Communication (ISCC), 2015 IEEE Symposium on", pp. 180–187 (IEEE, 2015).

[6] Benin, A., S. Toledo and E. Tromer, "Secure Association for the Internet of Things", in "Secure Internet of Things (SIoT), 2015 International Workshop on", pp. 25–34 (IEEE, 2015).

[7] Brauer, S., A. Zubow, S. Zehl, M. Roshandel and S. Mashhadi-Sohi, "On Practical Selective Jamming of Bluetooth Low Energy Advertising", in "Standards for Communications and Networking (CSCN), 2016 IEEE Conference on", pp. 1–6 (IEEE, 2016).

[8] Chakrabarty, S. and D. W. Engels, "Black Networks for Bluetooth Low Energy", in "Consumer Electronics (ICCE), 2016 IEEE International Conference on", pp. 11–14 (IEEE, 2016).

[9] Chakrabarty, S., D. W. Engels and S. Thathapudi, "Black SDN for the Internet of Things", in "Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on", pp. 190–198 (IEEE, 2015).

[10] Charlie Osborne, "Researchers discover over 170 million exposed IoT devices in major US cities", URL `http://www.zdnet.com/article/researchers-expose-vulnerable-iot-devices-in-major-us-cities/`, [online; accessed on June-2017] (2017).

[11] Das, A. K., P. H. Pathak, C.-N. Chuah and P. Mohapatra, "Uncovering Privacy Leakage in BLE Network Traffic of Wearable Fitness Trackers", in "Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications", pp. 99–104 (ACM, 2016).

[12] Elboukhari, M., M. Azizi and A. Azizi, "Intrusion Detection Systems in Mobile Ad Hoc Networks: A Survey", International Journal on Computational Sciences & Applications (IJCSA) **5**, 2, 27–36 (2015).

[13] Guo, Z., I. G. Harris, Y. Jiang and L.-f. Tsaur, "An Efficient Approach to Prevent Battery Exhaustion Attack on BLE-based Mesh Networks", in "Computing, Networking and Communications (ICNC), 2017 International Conference on", pp. 1–5 (IEEE, 2017).

[14] Hassan, S. S., S. D. Bibon, M. S. Hossain and M. Atiquzzaman, "Security threats in Bluetooth technology", Computers & Security (2017).

[15] Hessar, M., V. Iyer and S. Gollakota, "Enabling On-Body Transmissions with Commodity Devices", in "Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing", pp. 1100–1111 (ACM, 2016).

[16] Kolias, C., G. Kambourakis, A. Stavrou and J. Voas, "DDoS in the IoT: Mirai and Other Botnets", Computer **50**, 7, 80–84 (2017).

[17] Lotfy, K. and M. L. Hale, "Assessing Pairing and Data Exchange Mechanism Security in the Wearable Internet of Things", in "Mobile Services (MS), 2016 IEEE International Conference on", pp. 25–32 (IEEE, 2016).

[18] Marktscheffel, T., W. Gottschlich, W. Popp, P. Werli, S. D. Fink, A. Bilzhause and H. de Meer, "QR Code Based Mutual Authentication Protocol for Internet of Things", in "World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016 IEEE 17th International Symposium on A", pp. 1–6 (IEEE, 2016).

[19] Oliff, W., A. Filippoupolitis, G. Loukas *et al.*, "Evaluating the Impact of Malicious Spoofing Attacks on Bluetooth Low Energy Based Occupancy Detection Systems", (2017).

[20] Prada-Delgado, M., A. Vázquez-Reyes and I. Baturone, "Physical Unclonable Keys for Smart Lock Systems using Bluetooth Low Energy", in "Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE", pp. 4808–4813 (IEEE, 2016).

[21] Qu, Y. and P. Chan, "Assessing Vulnerabilities in Bluetooth Low Energy (BLE) Wireless Network Based IoT Systems", in "Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), 2016 IEEE 2nd International Conference on", pp. 42–48 (IEEE, 2016).

[22] Ronen, E. and A. Shamir, "Extended Functionality Attacks on IoT Devices: The Case of Smart Lights", in "Security and Privacy (EuroS&P), 2016 IEEE European Symposium on", pp. 3–12 (IEEE, 2016).

[23] Roux, J., E. Alata, G. Auriol, V. Nicomette and M. Kaâniche, "Toward an Intrusion Detection Approach for IoT Based on Radio Communications Profiling", in "13th European Dependable Computing Conference", p. 4p (2017).

[24] Ryan, M., "crackle", URL `https://github.com/mikeryan/crackle`, [online; accessed on April-2017] (2017).

[25] Ryan, M. *et al.*, "Bluetooth: With Low Energy comes Low Security.", WOOT **13**, 4–4 (2013).

[26] Sachidananda, V., S. Siboni, A. Shabtai, J. Toh, S. Bhairav and Y. Elovici, "Let the Cat Out of the Bag: A Holistic Approach Towards Security Analysis of the Internet of Things", in "Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security", pp. 3–10 (ACM, 2017).

[27] SIG, B., "Bluetooth core specification version 4.2", URL `https://www.bluetooth.com/specifications/bluetooth-core-specification` (2014).

[28] Townsend, K., C. Cufí, R. Davidson *et al.*, *Getting started with Bluetooth low energy: tools and techniques for low-power networking* (" O'Reilly Media, Inc.", 2014).

[29] Uher, J., R. G. Mennecke and B. S. Farroha, "Denial of Sleep Attacks in Bluetooth Low Energy Wireless Sensor Networks", in "Military Communications Conference, MILCOM 2016-2016 IEEE", pp. 1231–1236 (IEEE, 2016).

[30] van der Meulen, R., "Gartner Report", URL `http://www.gartner.com/newsroom/id/3598917`, [online; accessed on March-2017] (2017).

[31] Wikipedia, "Bluetooth security, wikipedia - the free encyclopedia", URL `https://en.wikipedia.org/wiki/Bluetooth#Security`, [online; accessed on 15-October-2016] (2016).

[32] Wikipedia, "Bluetooth, wikipedia - the free encyclopedia", URL `https://en.wikipedia.org/wiki/Bluetooth`, [online; accessed on 15-October-2016] (2016).

[33] Wikipedia, "Mobile security, wikipedia - the free encyclopedia", URL `https://en.wikipedia.org/wiki/Mobile_security#Attacks_based_on_communication_networks`, [online; accessed on 15-October-2016] (2016).

[34] Wikipedia, "Blockchain", URL `https://en.wikipedia.org/wiki/Blockchain`, [online; accessed on June-2017] (2017).

[35] Wikipedia, "Internet of things", URL `https://en.wikipedia.org/wiki/Internet_of_things`, [online; accessed on April-2017] (2017).

[36] Wikipedia, "Intrusion detection system", URL `https://en.wikipedia.org/wiki/Intrusion_detection_system`, [online; accessed on March-2017] (2017).

[37] Wilhelm, M., I. Martinovic, J. B. Schmitt and V. Lenders, "Short Paper: Reactive Jamming in Wireless Networks - How Realistic is the Threat?", in "Proceedings of the fourth ACM conference on Wireless network security", pp. 47–52 (ACM, 2011).

[38] Wireshark, "Wireshark", URL `https://www.wireshark.org`, [online; accessed on April-2017] (2017).

APPENDIX A

LIST OF ACRONYMS

ATT	Attribute Protocol
BD_ADDR	Bluetooth Address
BLE	Bluetooth Low Energy
BR/EDR	Basic Rate/ Enhanced Data Rate
CSRK	Connection Signature Resolving Key
CSV	Comma Separated Value
DoS	Denial of Service
DDoS	Distributed Denial of Service
ECDH	Elliptic-Curve Diffie-Hellman
GAP	Generic Access Profile
GATT	Generic Attribute Profile
HCI	Host Controller Interface
HDA	Helper Data Algorithm
IDS	Intrusion Detection System
IoT	Internet of Things
IRC	Internet Relay Chat
IRK	Identity Resolving Key
IRP	Peer Identitiy Resolving Key
ISM	Industrial, Scientific and Medical Standard
LL	Link Layer
L2CAP	Logical Link Control and Adaptation Protocol
MAC	Media Access Control
MITM	Man in the Middle
NFC	Near Field Communication
OOB	Out Of Band
OSI	Open Systems Interconnection
PCB	Printed Circuit Board
PIN	Personal Identification Number
PUF	Physically Unclonable Function
RFID	Radio Frequency Identification
RPA	Resolvable Private Address
RSSI	Received Signal Strength Indicator
SDIO	Standard Input Output
SDN	Software Defined Network
SNMP	Simple Network Management Protocol
SRAM	Static Random Access Memory
SSL	Secure Socket Layer
SSP	Secure Simple Pairing
TLS	Transport Layer Security
TTP	Trusted Third Party
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
WEP	Wireless Encryption Protocol
WPA	Wireless Protected Access
WPAN	Wireless Personal Area Networks