5-2022

A COMPARATIVE ANALYSIS OF DEVICES VIA THE BLUETOOTH PROTOCOL IN A TIME SERIES ANALYSIS

Joseph Vazquez

A COMPARATIVE ANALYSIS OF DEVICES VIA THE BLUETOOTH PROTOCOL IN A TIME SERIES ANALYSIS

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Information Systems and Technology

by

Joseph Corona Vazquez

May 2022

A COMPARATIVE ANALYSIS OF DEVICES VIA THE BLUETOOTH PROTOCOL IN A TIME SERIES ANALYSIS

A Project

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Joseph Corona Vazquez

May 2022

Approved by:

Dr. Harold Dyck, Committee Chair

Dr. Conrad Shayo, Committee Member

Dr. Jay Varzandeh, Department Chair, Information and Decision Sciences

ABSTRACT

The utilization of the Bluetooth protocol has provided many with the seamless transmission of data to multiple devices. Given its versatility and being an efficient process of connectivity, it has become one of the preferred methods of wireless connections. Despite this, an aspect of the Bluetooth function is still vulnerable to being exploited by having the data transmission stolen. This project answered the following questions: “How does one reduce the vulnerability by comparing normal and abnormal Bluetooth data?”, “How does one identify outlying variables within the data?” and “How can we improve the Bluetooth function?”. This project relied on previous research based on establishing patterns of life in Bluetooth devices in order to categorize such devices using their data. By applying a similar approach, this research is focused on creating a methodology of capturing, detecting, and analyzing normal and abnormal Bluetooth data. By creating two scenarios involving Bluetooth devices, one where a normal transmission happens and another where a Bluetooth Hijacking occurs, comparable scans were made and then compared. The findings were as follows: The analysis shows it is possible to categorize the Bluetooth devices and attribute their data to create a pattern of life. By comparing normal and abnormal Bluetooth data, vulnerability can be reduced by detecting abnormal data much sooner and thus alerting the user of any attacks. To identify the outlying variables, certain characteristics within the Bluetooth packet in Wireshark can be selected and shown in the RStudio graph. Having these variables displayed

creates a better visual to further analyze the data captured and identify any outlying variables. This project also introduced methods that the Bluetooth function can be improved on by including the introduction of more pin inputs when entering Bluetooth networks, as well as the idea to introduce a feature that authenticates the termination of a Bluetooth connection. The conclusion of this project revealed that these captures and analysis allow for establishing a pattern of life of what would be considered normal and abnormal data within the Bluetooth IoT and can be expanded into other Bluetooth devices.

ACKNOWLEDGEMENTS

The author of this project would like to thank Arturo Rubio, Jordan Hazelip, Joseph Mercado, Nabil Chandoo, and Oscar Romero from the INSuRE class of Fall 2020 for working on the previous iteration of this project. In addition, the author would like to give thanks to Eric Brasile and Stephanie Polczynski, Technical Directors from the NSA who provided the initial project idea. The author would also like to thank Faculty Advisor, Dr Tony Coulson for leading the entire INSuRE program for numerous universities, federal agencies, and national labs. Final acknowledgements to the primary chair and professor at the time of writing this culminating project, Professor Harold Dyck, Professor Conrad Shayo, and Department Chair Professor Jay Varzandeh.

DEDICATION

To my family, friends, and professors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE:

INTRODUCTION

Functions and Utilization

When it comes to devices having the capability of connecting to networks or to one another, they belong to a category known as the Internet of Things (IoT). These devices allow for a more convenient method of performing tasks and are becoming more present as technology continues to advance. With IoT devices, the majority have the capacity of communicating through Bluetooth and Bluetooth Low Energy frequencies as a means of wireless transmission. In the average home, there are multiple devices that are capable of transmitting and receiving information through the BT (Bluetooth) protocol and are consistently broadcasting the name of the device which is their Service Set Identifier (SSID). In a public setting, there are going to be numerous devices broadcasting their information making them become vulnerable to be scanned by any other IoT device utilizing the Network Interface Card (NIC). On a Bluetooth capture, the data displayed would depict a variety of information about the transmission including the name, channel frequency used, length of a data packet and much more. While the information captured in a scan can be used for malicious reasons, it can also be used in creating a methodology to establish behavioral patterns in devices (Aparicio-Navarro et al., 2017). This would involve the use of a capture device, a sniffing application to help capture data packet captures

(PCAPs), and a statistical program to help display the capture and analyze the data and create data patterns. Creating the data patterns allows for establishing normal and abnormal data and can be used to detect and stop attacks from occurring.

Problem Statement and Plan

Bluetooth and Bluetooth Low Energy (BLE) devices are actively broadcasting when pairing for a connection. Even while connected, it is possible to monitor the connection to display information about the transmission. Using the information, anyone can connect to your current connection and steal it. The Bluetooth connection can be singular or multiple, in other words, while connected to one device, the same device can be connected to another simultaneously. This multi-connection can happen to any other device requesting to pair. While newer Bluetooth versions allow for a password input to authenticate a connection, most devices do not have the prompt to input a password. Most users are then not aware of how to activate this prompt or have a device that allows them to do so (Chu, 2022). When the Bluetooth feature is on, its name is broadcasted and open to any new connection. This leaves it open to be exposed to any kind of attack including active/passive eavesdropping, man in the middle attacks, and data modifications (Gupta & Kumar, 2015, pg.155).

Categorization of BT devices have been done before utilizing their profiles and Radio Frequencies (Ali et al. 2018). But no experiment has been conducted to compare and analyze normal and abnormal Bluetooth data. A possible

methodology to identify threat actors is by establishing the behavioral signatures of Bluetooth devices prior to scanning and creating a comparison on similar device behaviors (Baxley et al., 2017). Creating and performing this methodology would involve testing on normal devices that contain the Bluetooth function and performing an analysis while in different states of activity. The data itself is recorded and can be converted and shown in a time series analysis that can display the data at any point in time of the capture based on chosen variables. The data given from IoT devices have patterns based on their behavior, known as pattern-of-life. An analysis between the pattern-of-life data and abnormal data captured in the PCAPs should yield results that display the data deviating from the normalcy and can be used to detect attacks utilizing the Bluetooth protocol.

From the discrepancies within the Bluetooth protocol, the following problem statements are proposed:

- How does one reduce the vulnerability by comparing normal and abnormal Bluetooth data?
- How does one identify outlying variables within the data?
- How can we improve the Bluetooth function?

CHAPTER TWO:

LITERATURE REVIEW

Prior Research

As devices are becoming wireless to be more appealing to the consumer, manufacturers opt for Bluetooth pairing as a means of seamless connectivity. Bluetooth itself as a means of communication has benefits including being low cost and can be integrated into virtually anything. Popular examples include the introduction of wireless earbuds, keyboards and mouse, and televisions. But as this trend continues, the attention paid to the security aspect to the Bluetooth protocol is lacking in comparison to other security implementations. This is a considerable issue as cyber-attacks are always prevalent and have occurred on more than one occasion involving protocols with a lot more security features.

In regard to the security features of Bluetooth and Bluetooth Low Energy devices, it centers around five basic security functions: Authentication, Confidentiality, Authorization, Message Integrity, and Paring (Padgette et al., 2022). To adhere to these 5 principles, the Bluetooth protocol utilizes various seeding algorithms and asymmetric key encryptions to protect the data being transferred. Emphasizing on these principles has made communication between devices secure with other security mechanisms in place. The first is when it can be turned on or off. This is its discoverable mode feature and by turning the Bluetooth function off, you're disabling the device from being detectable and thus, denying all attempting connections. The next feature is the Pairing Mechanism,

this is how the device authenticates a connection with another device. There are 3 Pairing Mechanisms, the first being Legacy Pairing which involves the use of a password or a pin. Upon connection, there will need to be an input of the passcode to establish a secure connection. The second is Secure Simple Pairing (SSP), which utilizes the Elliptic Curve Diffie Hellman protocol to establish a connection using public and private key pairs. Lastly is the Secure Connection (SC) mechanism that builds on the SSP mechanism by using longer key sizes and stronger algorithms. The SSP mechanism uses SAFER+ for encryption and authentication and P-192-ECDH with HMAC-SHA256 for key generation when connecting while the SC connection uses HMAC-SHA256 for authentication, P-256-ECDH with HMAC-SHA256 for key generation and AES-CTR for encryption (Lounis & Zulkernine, 2020).

The level of security of Bluetooth and Bluetooth Low energy devices have been tested before through scenarios involving common IT attacks. These attacks include Denial of Service Attacks, Man-in-the-Middle, message modifications, and resource misappropriation (Gupta & Kumar, 2015, pg.155). These are just a few examples of what can be exploited when it comes to Bluetooth devices. One of the Bluetooth attacks include Bluesniffing which involves eavesdropping on an already established Bluetooth connection. Here the attacker can capture the data packets in the transmission which can include files, passwords, and even audio data. Another is Bluespoofing where an attacker spoofs the name of a known device causing someone to pair it on

purpose. As depicted in **Table 1,** other types of possible exploited vulnerabilities are classified by the types of attack (Mukhtar Ahmad Sofi, 2016).

Attack Type	Threats	Purpose
Surveillance	Blueprinting, bt_audit, redfang, War-nibbling, Bluefish, sdptool, Bluescanner, BTScanner	observe and gather information about the device and its location
Range Extension	BlueSniping, bluetooone, Vera-NG	extend the device range so that attacks could be conducted from far way distance
Man In The Middle	BT-SSP-Printer-MITM, BlueSpooof, bthidproxy	place a device between two connected devices. All the information sent through the channel are available to the device in between.
Denial Of Service	Battery exhaustion, signal jamming, BlueSYN, Blueper, BlueJacking, vCardBlaster	Deny resources to a target by saturating the communication channel.
Obfuscation	Bdaddr, hciconfig, Spooftooph	hide the attacker's identity.
Fuzzer	BluePass, Bluetooth Stack Smasher, BlueSmack, Tanya, BlueStab	submit non standard input to get different results.
Malware	BlueBag, Caribe, CommWarrior	carry out attacks typically using self replicating form of software.
Unauthorised Direct data access	Helomoto, Bloover, BlueBug, BlueSnarf, Unauthorized BlueSnarf++, BTCrack, Car Direct Data Whisperer, HeloMoto, btpincrack	gather private information in an unauthorized manner.
Sniffing	FTS4BT, Merlin, BlueSniff, HCIDump, Wireshark, kismet	capture the Bluetooth traffic in transit.

Table 1: BT Threat Classification

When a BT device connects to another, this ad-hoc area of connectivity is known as piconet. In this network, the devices are communicating on the same frequency and channel and additional devices can be added. Once inside a piconet, they have the capability to communicate with other piconets if they get near one resulting in a scatternet as shown in **FIGURE 1** (Asaf et al., 2017). The units represented as dark circles are called "Masters" and these are the main devices that are supplying communication to the white circles called "Slaves". This depiction shows how the 3rd device (Master) is acting as a bridge to the 2nd

piconet and supplying data to the 5th and 6th devices. From here, any new device would be able to connect into any circle and begin a new transmission with the main Bluetooth device (Master).

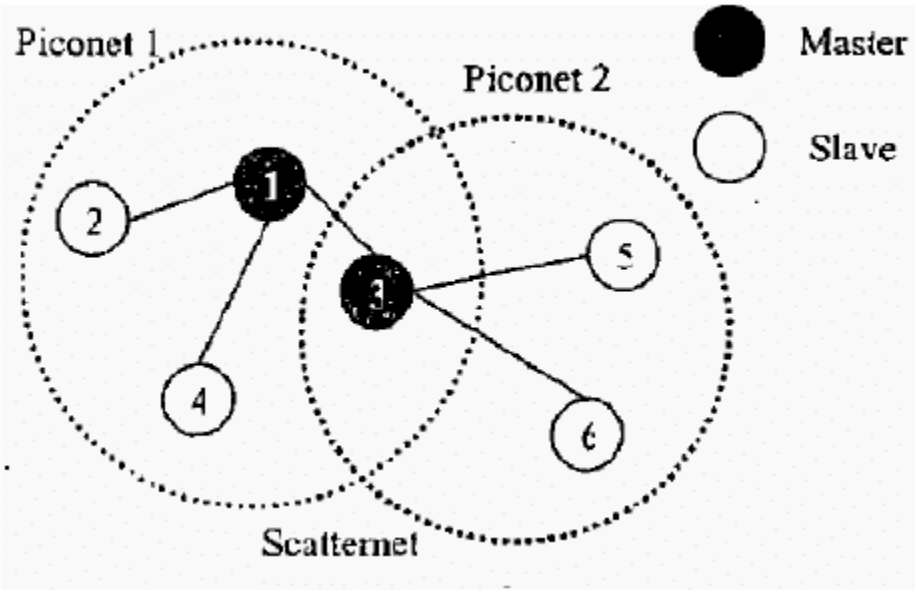


Figure 1 Scatternet

In order to prevent threats from happening on these devices, a method of fingerprinting has been researched and conducted on wireless communication devices. Specifically, Radio Frequency (RF) fingerprinting as it's seen as a promising solution for security problems by enhancing security within the physical layer. Fingerprinting is a process in which devices can be identified and classified within a given network by its frequencies, including Bluetooth. In a research study conducted by Ali et al. (2018), they were able to categorize 20 phones utilizing their frequencies and using the Empirical Mode Decomposition and the Hilbert Spectrum to display the phones' information. This method of analysis allowed for

the categorization of the devices using this technique, but this research did not focus if the capturing data could be analyzed to detect malicious behavior. This opens the opportunity for this project to document the potential findings when it comes to analyzing the Bluetooth captures in a time series format in order to establish a pattern-of-life on its data.

CHAPTER THREE:

PROJECT RESEARCH

Project Research Methodology

The origin of this methodology is derived from the INSuRE project in collaboration with the NSA. The initial project involved creating a methodology to actively scan and analyze Bluetooth data, and the ultimate goal of the project was to create method that would scan incoming BT devices in a certain vicinity and establish a pattern of life. This project is mainly focusing on the comparative analysis of normal and abnormal Bluetooth data. Unlike the INSuRE project that focused on the active scanning of Bluetooth devices inside a given area, the purpose of this project is to create a comparison of normal and abnormal data captured from a BT transmission utilizing RStudio to aid in the analysis.

In the INSuRE project, before performing the data analysis, data captures were first collected from Bluetooth emitting devices. To capture BT packets, a special device was used to only capture Bluetooth LE (Low Energy) frequencies. The Bluefruit LE Sniffer from Adafruit was used in combination with Wireshark to capture the Bluetooth data. Wireshark is capable of scanning for Bluetooth signals but using the attached Bluefruit Sniffer, it can scan only for Bluetooth LE signals. The output of this resulted in scans displaying packet sizes, signal strength, and channel frequency. As for the devices, a variety of devices were used in the initial captures including a Samsung Galaxy S10, Fitbits, Apple Airpods, and Sony headphones. The purpose of using multiple devices was to

establish a pattern of life and create a profile on devices to determine if different devices have a different data pattern.

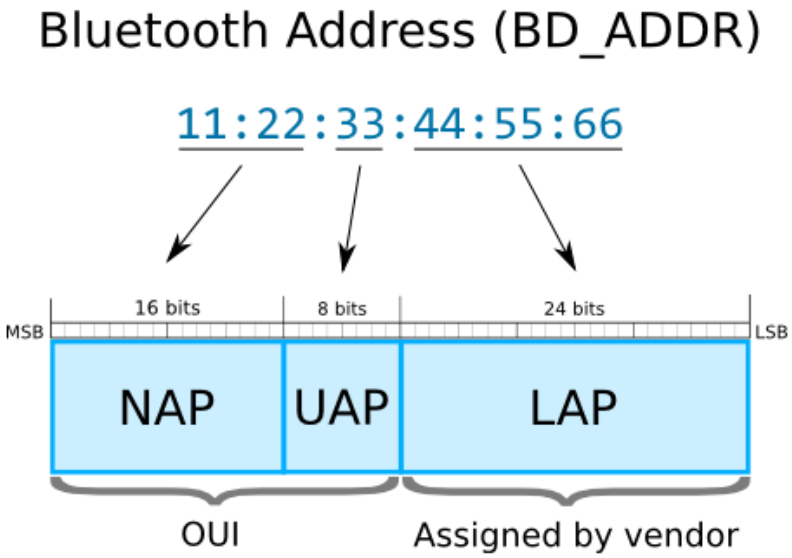
For this project, only a handful of devices will be used to determine if the methodology will be successful. The collection of normal data will be produced from scanning a regular BT connection using the NIC from a Raspberry Pi. Utilizing the CLI (command Line Interface), the hope is to mimic an attacker attempting to hijack a BT connection. The experiment involving the abnormal data will consist of a Sony speaker, a laptop, and the Raspberry Pi. The Sony speaker will be connected to the laptop and play music while the Raspberry Pi will be performing a scan for BT signals. The scan will display the MAC and the advertising BT address for the Sony Speaker which the Raspberry Pi can use to establish its own connection to the speaker. This type of attack is known as Bluejacking, and the entirety of this attack will be captured in a Wireshark scan.

Capturing Data

To begin capturing the data, devices would be placed in proximity to the Raspberry Pi and have their Bluetooth feature turned on and connected to a device. On the Wireshark application, the data packets will be captured and exported as PCAPs or CSV files. Within each of these entries, the data packets themselves can be opened and viewed to see the contents of a Bluetooth packet. **Figure 2** depicts how the MAC address is composed of 3 segments divided into two halves (Deconstruct bluetooth address and Bluetooth MAC address query - programmer sought 2018). The first half, known as the top half, is called the

Organizationally Unique Identifier (OUI) and is determined by the device's manufacturer while the second half is a value determined by the vendor. As for the segments, the first segment is called the Non-Significant address (NAP) and is a value used in Frequency Hopping Synchronization frames. Segment 2 is the Upper Address Part (UAP), and it is used for seeding various Bluetooth algorithms. The third segment is the Lower Address Part (LAP) and is a value unique to that specific device.

Figure 2: Bluetooth Address Structure



The illustration in **Figure 3** details the contents within a BLE data packet (RF Wireless World, 2012). As depicted, the data packet is composed of 4

segments with the Protocol Data Unit (PDU) being the main segment. The PDU segment is what will be advertised when the device is broadcasting its address.

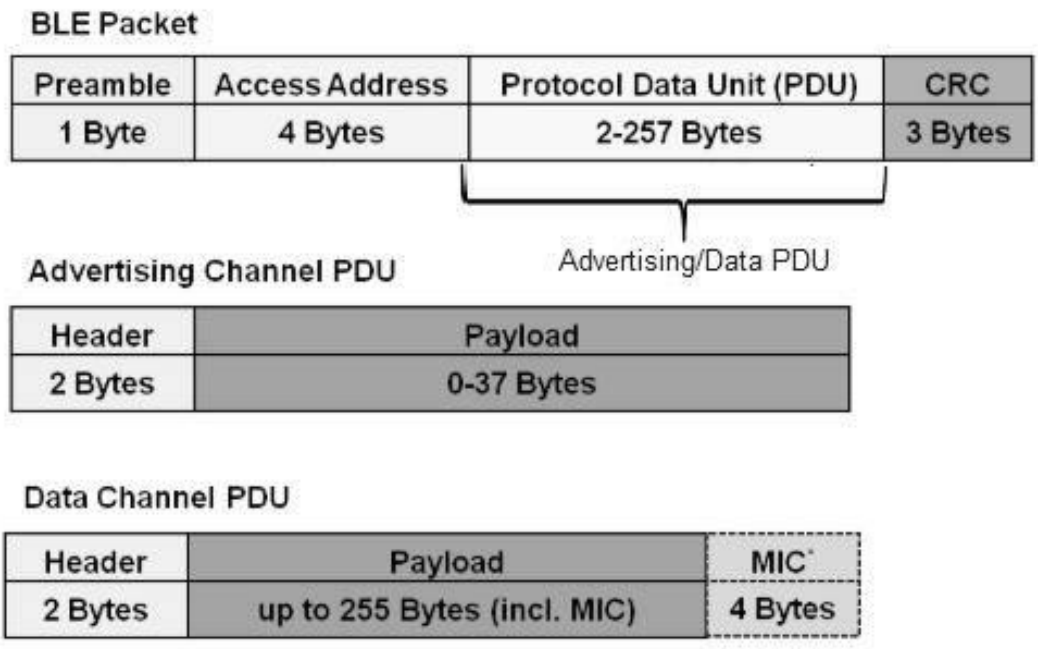


Figure 3: BLE Packet

As for the other segments, the following is an overview for what they're for:

- Preamble: used for the synchronization and timing estimation in the receiver.
- Access Address: this is derived from the LAP
- CRC: Cyclic Redundancy Check is a checksum for the data packet to ensure that the packet is not corrupted when sent.

RStudio

The data collected will be converted from PCAP files to CSV in order to input the dataset into RStudio. A code for the program is made to display certain variables within the CSV file. Opening the CSV in RStudio allows data to be selected and then displayed in a time series format. The final output is then opened in the RStudio application or in a browser. This output allows for a further analysis of the data using the mouse to determine data in a specific point of time. An example of the code and the output can be seen in **Figure 4** and **Figure 5**. The two datasets that will be analyzed are the FinalNormal2.csv and the FinalHijack.csv. Each of the datasets will be loaded into the “MyData” variable and then the data variables to be analyzed are then selected. Some of the packages used to help create the time series graphs are dplyr and dygraphs. The dplyr package is used to enable data manipulation and the dygraphs package is used to plot the data in a time series graph.

```
title: "BluetoothProj"
author: "Joseph Corona Vazquez"
output: html_document
---

# Use dygraph
# Install the package dygraphs only once.
#install.packages('dygraphs')
library(dplyr)
library(dygraphs)

# Read the dataset
MyData = read.csv('FinalNormal2.csv')
MyData = read.csv('FinalHijack.csv')

#MyData = read.csv('FinalVictim.csv')


# Then you need to specify which columns you will use:
My_Data_2 = MyData %>% select(matches('Time'),matches('Frame'),
                              matches('Destination'), matches('Protocol'),
                              matches('Source'))


#Graph Output
#Time_Series_Graph <- dygraph(My_Data_2)
#Time_Series_Graph

#Range Input
dygraph(My_Data_2) %>% dyRangeSelector()
```

Figure 4: RStudio Code

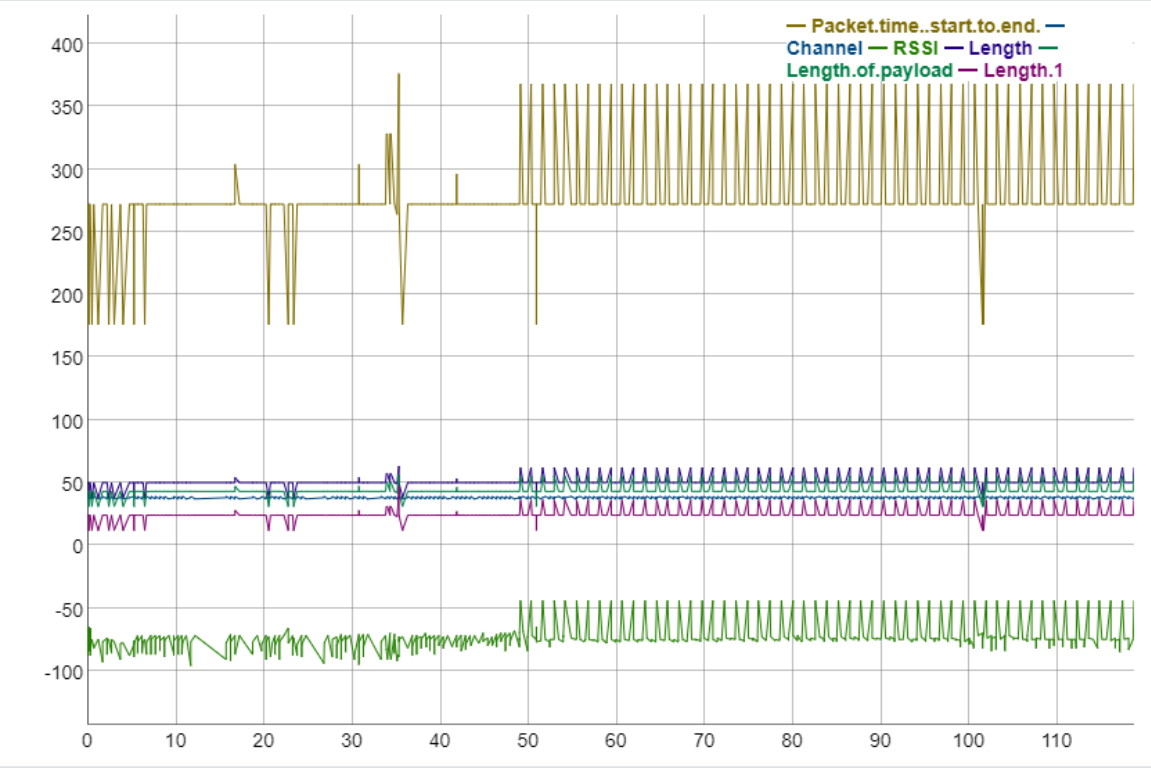


Figure 5: Time Series Graph Example

CHAPTER FOUR:

ANALYSIS AND RESULTS

Bluetooth Pattern of Life Analysis

Identifying any outlying data variables to differentiate normal and abnormal activity involved the examination of the Bluetooth captures. Captures from the Sony speaker were used to analyze and infer any behavioral findings within the data. To create a capture of abnormal data, an attack was performed on the speaker while it was being used. In this attack, a laptop was connected to the speaker and playing music. A Raspberry Pi was used to scan for Bluetooth signals using the CLI and common commands found in a Kali Linux environment. Once the scan found the Sony Speaker, the Raspberry Pi hijacked the connection from the speaker in place of the laptop. After establishing the new connection, the Raspberry Pi then played audio files to confirm a successful Bluejacking attack. This scenario will be similar to entering a Piconet and stealing the Bluetooth connection from an advertising device. A diagram of the attack can be viewed in **Figure 6**.

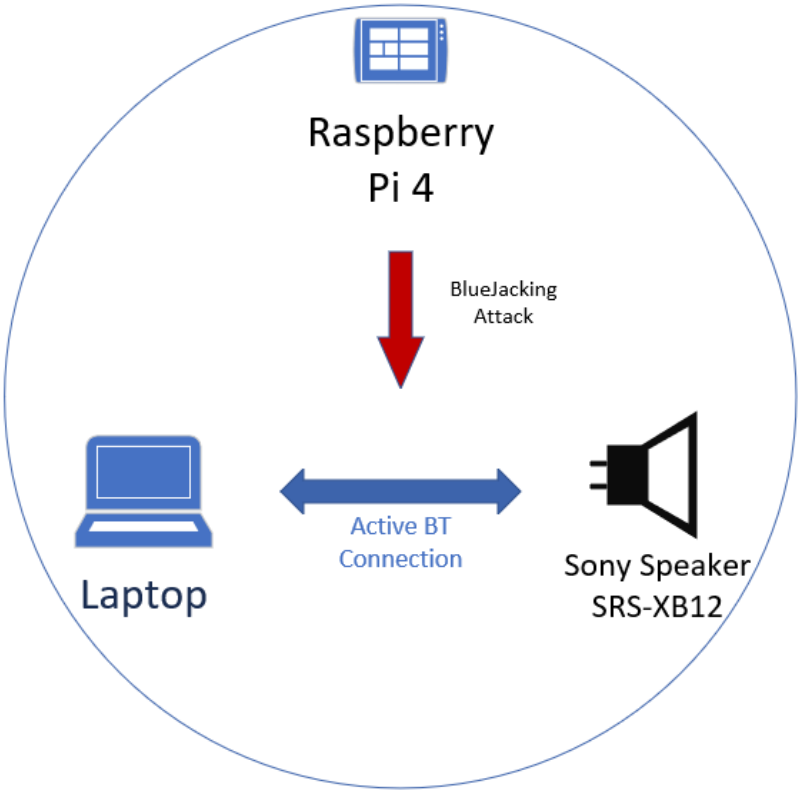


Figure 6: Diagram of Experiment

Comparisons

The total length of the captures are around sixteen thousand packets displaying multiple protocols, packet data lengths, and the source and destination MAC address. The protocols and MAC addresses were each assigned numerical values in order to be graphed. **Table 2** helps define what the various protocols are and their corresponding numerical values (for full definitions see Appendix A). **Table 3** displays the MAC addresses captured in the scan and their corresponding numerical value.

Protocol	Definition	Numeric Map
CMD	Command Mode	5
EVT	Engineering Validation Testing	10
L2CAP	Logical Link Control and Adaptation Protocol	15
SDP	Service Discovery Protocol	20
RFCOMM	Radio Frequency Communication	25
HFP	Hands Free Profile	30
AVDTP	Audio/Video Data Transport Protocol	35
AVRCP	Audio/Video Remote Control Profile	40
SBC	Low Complexity Subband Coding	45
HCI_5	Host Controller Interface 5	50
HCI_10	Host Controller Interface 10	60

Table 2: Bluetooth Protocols

MAC Address	Device	Numeric Map
98:52:3d:46:d5:5f	Sony Speaker SRS-XB12	57
00:00:00:00:00:00	Local Host (Raspberry PI)	48

Table 3: MAC Addresses

For the first capture, the Raspberry Pi monitored a normal Bluetooth connection between the Sony Speaker and the Laptop to create a baseline for normal traffic. Results from the scan can be seen in **Figure 7.** The scan itself contained over five thousand packets and displayed multiple Protocols and various frame length sizes captured at each interval. The preselected variables are displayed at the top right of the graph in **Figure 7** and are displaying the Time, Frame Length, Destination, Protocol, and the Source of each individual packet. For each of the datasets, the data within the CSV file were parsed to get a better representation of what was present in the captures. See Appendix B to view the full parse of the datasets.

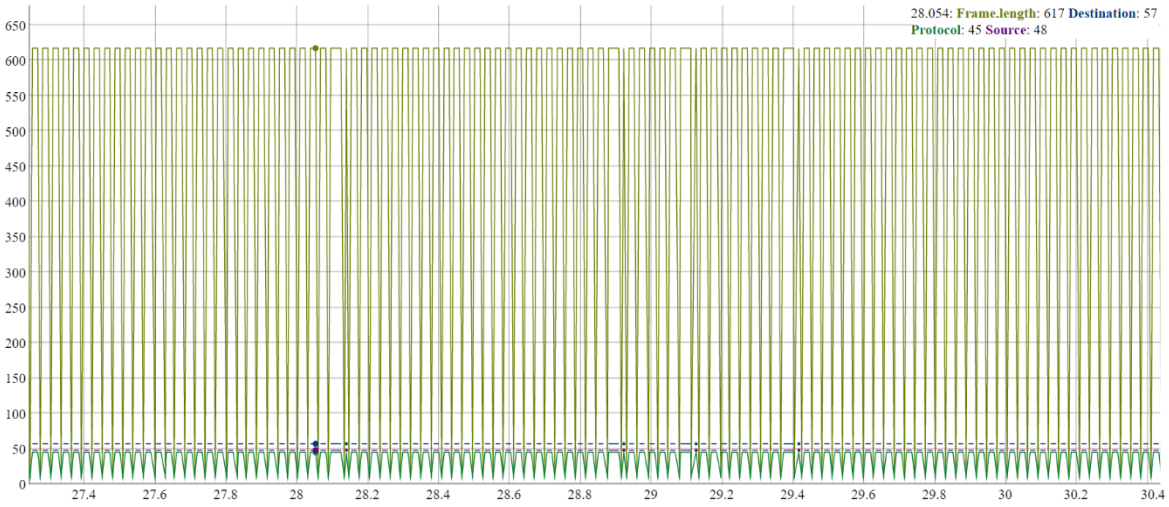


Figure 7: Normal Traffic Graph

In this capture, the mouse cursor is hovering over a specific instance of the data. In this instance, the current value for Frame length is equal to 617 bits, with the Protocol in that instance being SBC and the time of the capture being 28.054 seconds. The Source (Laptop) and the Destination (Sony Speaker) remain static as the normal capture only monitored the Bluetooth connection. The rest of graph not displayed is consistent with what is shown in **Figure 7** and there are no signs of any abnormal variables within the parsed information. The most frequent protocol recorded was the SBC protocol as this is the main protocol for transferring audio data. The most consistent packet size was 617 bits with over three thousand occurrences followed by a packet size of 8 bits with over nineteen hundred occurrences.

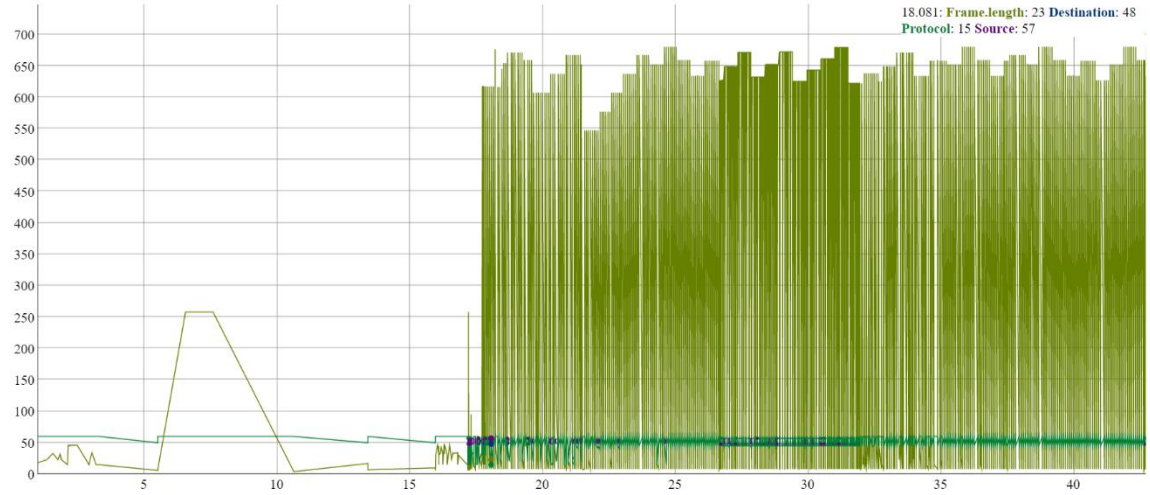


Figure 8: Hijack Graph

In **Figure 8,** it shows the Bluejacking attack attempted on the Sony Speaker. This was recorded from the Raspberry Pi depicting an initial scan for Bluetooth Addresses then depicting a forced connection to the Sony Speaker. The beginning of the graph shows a lack of data being displayed; this is where the scan for the Sony Speaker occurs with the average packet size being less than 50. Then once the Raspberry Pi is forcing a connection with the Sony Speaker, the average packet size begins to fluctuate with packet sizes around 600 bits. After a few seconds, the graph then begins to resemble the graph of the normal Bluetooth Scan (see **Figure 9**).

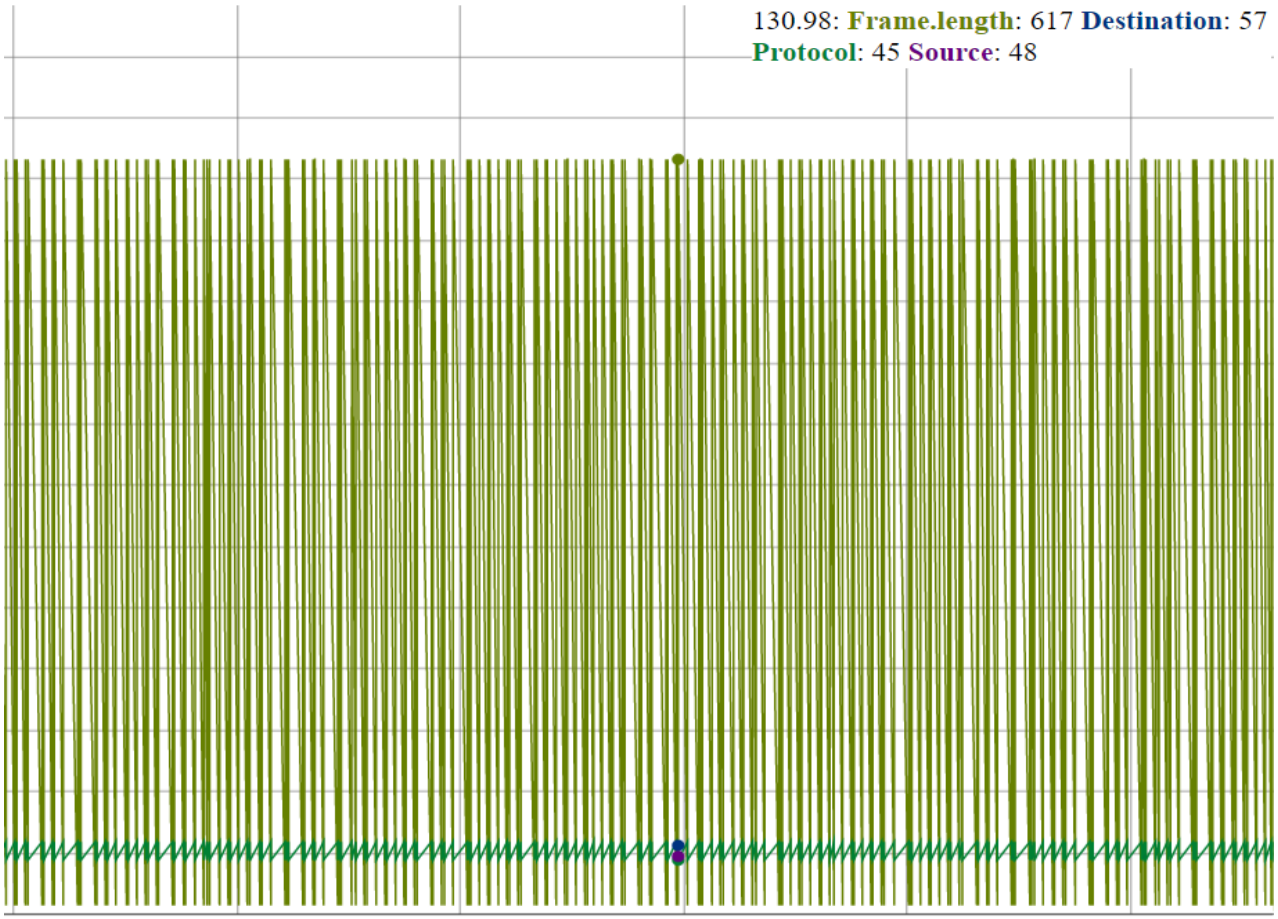


Figure 9: Hijack Graph Displaying Normal Values

In the initial scan of the Hijack capture, the average size of the data packets were less than 50 bits and the most consistent protocols used were HCI_10 and L2CAP. Since the scan was actively scanning, the two protocols were being sent out to identify a connection to then establish a Bluetooth connection. After the connection was successfully hijacked, the scan then displayed similar data to that shown in the Normal capture showing an average packet size of 617 bits and the most used protocol being the SBC to transfer audio data. To fully view the parsed data of the capture, see Appendix C.

CHAPTER FIVE:

DISCUSSION, CONCLUSIONS, AND FUTURE RESEARCH

Conclusion

The Bluetooth protocol is used by a variety of devices allowing for a seamless way of connecting and playing media. Already in its 5.2 version, it shows no sign of becoming obsolete while other forms of technology become outdated. As newer technology comes to fruition, Bluetooth always finds its way to be integrated one way or another. Despite the already-in-place security methods, Bluetooth and Bluetooth LE technology are still susceptible to attacks. Through the conducted research, the following problem statements were proposed:

How does one reduce the vulnerability by comparing normal and abnormal Bluetooth data?

The methodology researched and performed displayed the possibility of capturing Bluetooth data and displaying it in a time series graph. Through the graphs, it is possible to analyze the data and allowed the opportunity to infer how the data is being utilized and if it can be used to detect any mal-intent. Further research into this area can introduce multiple scans on different devices to create a library of normal and abnormal data behavior and signatures.

How does one identify outlying variables within the data?

Through the created methodology, it is possible to capture, scan and output the data in a way to display certain variables and create a visual to better detect any

abnormalities. By also creating a normal scan to compare it to, it helps create a better distinction between the two scans to establish a pattern of life. Establishing a pattern of life on the devise allows to better make distinctions.

How can we improve the Bluetooth function?

The introduction of entering passcodes and pins ensures some level of security to the device and user. But once a device is a part of a piconet that process of authenticating connections is completely bypassed, and the connection can be stolen by another user.

A possible technique to mitigate this from occurring is to include a second passcode and pin feature when a device enters a piconet. Similar to needing a pin code when connecting to a new device, a user would be required to enter a new pin to enter the piconet and be a part of the BT network. Alternatively, instead of requesting for a pin to authenticate, the host or master device can include a feature to verify if the user wants to disconnect. Parallel to entering a pin to establish a connection, the device can also request a pin to disconnect completely. This gives the user full control for both creating and closing connections.

Future Works

This research allows for others to apply the same or similar methodologies in order to create a larger dataset for already present BT emitting devices. Establishing a pattern-of-life on popular devices creates and establishes precautionary measures to mitigate cyber-attacks. By having preidentified data

signatures on devices, it allows spotting for any abnormal activity to be a lot simpler. As presented, a common IT attack on a Sony Speaker displayed data very abnormal when compared to that of normal data. By having the normal data already present and graphed, an analysis showed the disparity of the two data sets creating an evident display of the IT attack. This research is also open to the possible introduction of machine learning. The scans and analysis are all capable of being accomplished through a command line or through a graphic user interface. With the introduction of machine learning, this methodology can all be integrated into a single application. This application can perform the scans for nearby Bluetooth devices and then maps the data into a code similar to RStudio. This live scan can then be outputted into a live time series graph occurring in real time. Similar to an intrusion detection system, it can monitor for Bluetooth traffic and create alerts when it detects any abnormal data signatures. Future research into Bluetooth and Bluetooth LE should incorporate the capture and analysis of Bluetooth data to establish a pattern of life. This would not only allow for a more secure protocol but also introduce another security method for the IoT world.

APPENDIX A

DEFINITION OF BLUETOOTH PROTOCOLS

A TABLE CONTAINING THE FULL DEFINTIONS OF THE BLUETOOTH PROTOCOLS ANALYZED IN THE CAPTURES.

Protocol	Definition	Numeric Map
CMD	Command Mode	Verifies that the connection and settings are correct
EVT	Engineering Validation Testing	Protocol that verifies if the hardware meets specifications
L2CAP	Logical Link Control and Adaptation Protocol	Part of the Bluetooth Stack, it passes packets to the HCI and segments packets.
SDP	Service Discovery Protocol	Allows devices to discover services to help support one another and determine what parameters to use.
RFCOMM	Radio Frequency Communication	A set of transport protocols that provides reliable data stream.

HFP	Hands Free Profile	Protocol that allows for hands free communication with mobile devices.
AVDTP	Audio/Video Data Transport Protocol	Used to stream music over the L2CAP channel
AVRCP	Audio/Video Remote Control Profile	Provides a standard interface allowing for a single device to have user access
SBC	Low Complexity Subband Coding	Used for transferring audio data over BT
HCI_5 **HCI_10**	Host Controller Interface	A layer in the Bluetooth stack that transports commands between the host and controller

APPENDIX B

OCCURRENCES OF PROTOCOLS IN NORMAL SCAN

A TABLE CONTAINING THE AMOUNT OF PROTOCOL OCCURENCES IN THE NORMAL SCAN OF A BT TRANSMISSION

Protocol	Occurrences
AVDTP	2
SBC	3914
AVRCP	1957
EVT	3

APPENDIX C

OCCURRENCES OF PROTOCOLS IN HIJACK SCAN

A TABLE CONTAINING THE AMOUNT OF PROTOCOL OCCURENCES IN THE HIJACK SCAN OF A BT TRANSMISSION

Protocols	Occurrences
HCI_5	23
HCI_60	3416
L2CAP	42
SDP	18
RFCOMM	11
HFP	15
AVDTP	8
SBC	6447
AVRCP	20

REFERENCES

A. M. Ali, E. Uzundurukan and A. Kara, "Assessment of Features and Classifiers for Bluetooth RF Fingerprinting," in IEEE Access, vol. 7, pp. 50524-50535, 2019, doi: 10.1109/ACCESS.2019.2911452.

Aparicio-Navarro, Francisco & Kyriakopoulos, Konstantinos & Gong, Yu & Parish, David & Chambers, Jonathon. (2017). Using Pattern-of-Life as Contextual Information for Anomaly-Based Intrusion Detection Systems. IEEE Access. 5. 1-1. 10.1109/ACCESS.2017.2762162.

Asaf, Khizra & Sarwar, Muhammad & Kashif, Muhammad & Talib, Ramzan & Khan, Irfan. (2017). A Review of Bluetooth based Scatternet for Mobile Ad hoc Networks. International Journal of Advanced Computer Science and Applications. 8. 10.14569/IJACSA.2017.080653.

Baxley, R. J., Roland, C. J., & Engle, M. T. (2017, August 15). Anomalous behavior detection based on behavioral signatures.

“Bluetooth Mac Address Changer for Windows.” What Is Bluetooth Address (BD_ADDR) | Bluetooth MAC Address Changer for Windows, https://macaddresschanger.com/what-is-bluetooth-address-BD_ADDR.

Bluetooth command reference ... - SparkFun Electronics. (n.d.). Retrieved March 24, 2022, from https://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/bluetooth_cr_UG-v1.0r.pdf

Chu, M. (2022, March 19). Can you connect multiple bluetooth devices at once? see 7 concurrently. Data Overhaulers. Retrieved April 13, 2022, from https://dataoverhaulers.com/connect-multiple-bluetooth-devices/#:~:text=How%20To%20Connect%20Multiple%20Bluetooth%20Devices%201%20Power,paired%20or%20available%20devices.%20...%20More%20items...%20

Daniele Antonioli, Nils Ole Tippenhauer, and Kasper Rasmussen. 2020. Key Negotiation Downgrade Attacks on Bluetooth and Bluetooth Low Energy. ACM Trans. Priv. Secur. 23, 3, Article 14 (August 2020), 28 pages. DOI:https://doi.org/10.1145/3394497

F. J. Aparicio-Navarro, K. G. Kyriakopoulos, Y. Gong, D. J. Parish and J. A. Chambers, "Using Pattern-of-Life as Contextual Information for Anomaly-Based Intrusion Detection Systems," in IEEE Access, vol. 5, pp. 22177-22193, 2017, doi: 10.1109/ACCESS.2017.2762162.

Grant Collins. (2020, December 18). *I Tried Hacking a Bluetooth Speaker... (and failed...)* [Video]. YouTube. https://www.youtube.com/watch?v=9XURbq9jjQs&list=LL&index=3&t=360s&ab_channel=GrantCollins

Host controller interface (HCI). Host Controller Interface (HCI) - BLE-Stack User's Guide for Bluetooth 4.2 3.01.01.00 documentation. (n.d.). Retrieved April 3, 2022, from https://software-

dl.ti.com/lprf/simplelink_cc2640r2_latest/docs/blestack/ble_user_guide/html/ble-stack-3.x/hci.html

Junjie Yin, Zheng Yang, Hao Cao, Tongtong Liu, Zimu Zhou, and Chenshu Wu. 2019. A Survey on Bluetooth 5.0 and Mesh: New Milestones of IoT. ACM Trans. Sen. Netw. 15, 3, Article 28 (August 2019), 29 pages. DOI:https://doi.org/10.1145/3317687

K. Lounis and M. Zulkernine, "Attacks and Defenses in Short-Range Wireless Technologies for IoT," in IEEE Access, vol. 8, pp. 88892-88932, 2020, doi: 10.1109/ACCESS.2020.2993553.

Kumar, M., & Gupta, B. K. (2015, March). Security for Bluetooth enabled devices using BlipTrack Bluetooth detector. In *2015 International Conference on Advances in Computer Engineering and Applications* (pp. 155-158). IEEE.

M. Chernyshev, C. Valli and M. Johnstone, "Revisiting Urban War Nibbling: Mobile Passive Discovery of Classic Bluetooth Devices Using Ubertooth One," in IEEE Transactions on Information Forensics and Security, vol. 12, no. 7, pp. 1625-1636, July 2017, doi: 10.1109/TIFS.2017.2678463.

M. La Polla, F. Martinelli and D. Sgandurra, "A Survey on Security for Mobile Devices," in IEEE Communications Surveys & Tutorials, vol. 15, no. 1, pp. 446-471, First Quarter 2013, doi: 10.1109/SURV.2012.013012.00028.

Mukhtar Ahmad Sofi. (2016). Bluetooth protocol in internet of things (IOT), security challenges and a comparison with wi-fi protocol: A Review.

International Journal of Engineering Research And, V5(11). https://doi.org/10.17577/ijertv5is110266

Panigrahy, Saroj Kumar & Jena, Sanjay & Turuk, Ashok. (2011). Security in Bluetooth, RFID and wireless sensor networks. ACM International Conference Proceeding Series. 628-633. 10.1145/1947940.1948071.

Padgette, J. , Bahr, J. , Batra, M. , Holtmann, M. , Smithbey, R. , Chen, L. and Scarfone, K. (2017), Guide to Bluetooth Security, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.SP.800-121r2 (Accessed March 22, 2022)

Padgette, J. , Chen, L. and Scarfone, K. (2012), Guide to Bluetooth Security, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.SP.800-121r1 (Accessed March 22, 2022)

“RF Wireless World.” BLE Advertising Packet Format (2012) | BLE Data Packet Format, https://www.rfwireless-world.com/Terminology/BLE-Advertising-and-Data-Packet-Format.html.

Rohit Kapoor, Manthos Kazantzidis, Mario Gerla, and Per Johansson. 2001. Multimedia support over bluetooth Piconets. In Proceedings of the first workshop on Wireless mobile internet (WMI '01). Association for Computing Machinery, New York, NY, USA, 50–55. DOI:https://doi.org/10.1145/381472.381569

SBC codec in the telephony environment: NFON KNOWLEDGEBASE. SBC codec in the telephony environment | NFON Knowledgebase. (n.d.). Retrieved March 3, 2022, from https://www.nfon.com/en/get-started/cloud-telephony/lexicon/knowledge-base-detail/sbc-codec

Scarfone, K. and Padgette, J. (2008), Guide to Bluetooth Security, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.SP.800-121 (Accessed March 22, 2022)

Senin, P., Lin, J., Wang, X., Oates, T., Gandhi, S., Boedihardjo, A.P., Chen, C., & Frankenstein, S. (2015). Time series anomaly discovery with grammar-based compression. EDBT.

SL_BT_EVT_CONNECTION_PARAMETERSCONNECTION. sl_bt_evt_connection_parametersConnection - v3.1 - Bluetooth API Documentation Silicon Labs. (n.d.). Retrieved March 1, 2022, from https://docs.silabs.com/bluetooth/3.1/group-sl-bt-evt-connection-parameters

Sauter, M. (2021). From Gsm to Lte-advanced pro and 5G an introduction to mobile networks and Mobile Broadband. Wiley.

Townsend, K. (n.d.). Introduction to bluetooth low energy. Adafruit Learning System. Retrieved March 24, 2022, from https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt

Troy A. Johnson and Patrick Seeling. 2012. Localization using bluetooth device names. In Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '12). Association for Computing Machinery, New York, NY, USA, 247–248. DOI:https://doi.org/10.1145/2248371.2248408

Vigo. (2016, February 18). Update #8: From EVT to DVT. Medium. Retrieved March 8, 2022, from https://medium.com/@wearvigo/update-8-from-evt-to-dvt-5b493b42d56