System Requirements Specification

for

Bluetooth Protocol Analytical Research

Version 3.0

Gianna Scarangelli
Matthew Irvin
Carolina Terre
Jonah Rowell
Connal Grace

Embry Riddle Aeronautical University Daytona Beach

11/21/2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
	9/29/2023		v1.0
	10/31/2023	updates	v2.0
	11/21/23	Final Version	v3.0

1. Introduction

1.1 Purpose

This document should define all the requirements and conditions revolving around the protocols that Bluetooth and Zigbee follow. It will additionally follow the physical connections between the server/client as well as the user interface in which to view the findings.

1.2 Document Conventions

Currently no standard conventions are being used, which is scheduled to change as further research about the topic is uncovered. Each requirement will have a specific priority in the long term of this project, as sprints continue this is also set to change based on our progress or a change in priority.

1.3 Intended Audience and Reading Suggestions

The intended users for this finding would be cybersecurity researchers, developers using these wireless configurations, as well as the everyday person who has an IoT device and utilizes them. This document will contain different sections: Section 2 will discuss the functions and prospected results of the research; Section 3 focuses on how one would be able to use and read information from the product; Section 4 is the primary priorities attempted to achieve by the end of this semester; and Section 5 will be the standards that are to be followed as this system is developed.

1.4 Product Scope

With wireless IoT devices becoming ever more prevalent throughout the world, we wish to search for vulnerabilities between clients and servers as they connect using Bluetooth or Zigbee and create a system able to aptly display these findings. Our software will focus on the breach of the vulnerabilities, attempting to extract or disrupt information, and prioritizing integrity, availability, then confidentiality. Additionally, the information would be displayed in an orderly fashion to the user. We will use hardware to simulate our server/clients to further analyze possible physical constraints that may impact the security.

1.5 References

1.5.1 Common Weakness Enumeration (CWE), from https://cwe.mitre.org/

1.5.2 National Vulnerability Database, from https://nvd.nist.gov/

2. Overall Description

2.1 Product Perspective

Our product is self-contained to analyze the safety of Bluetooth and the signals between a server and a client. See Figure 1 and 2 below.

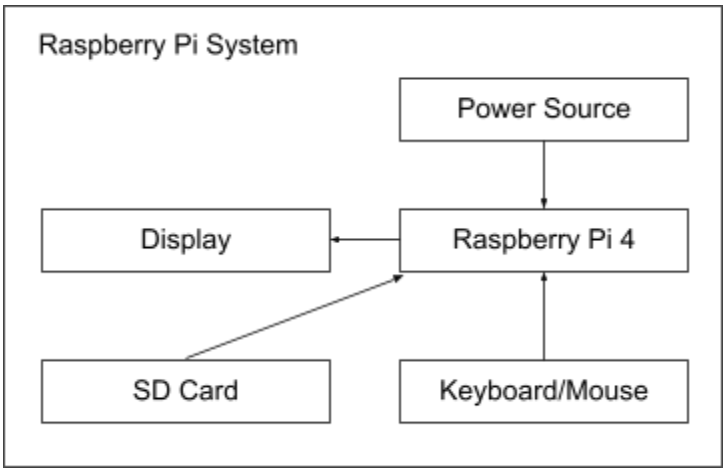

Figure 1: Diagram of the Raspberry Pi System

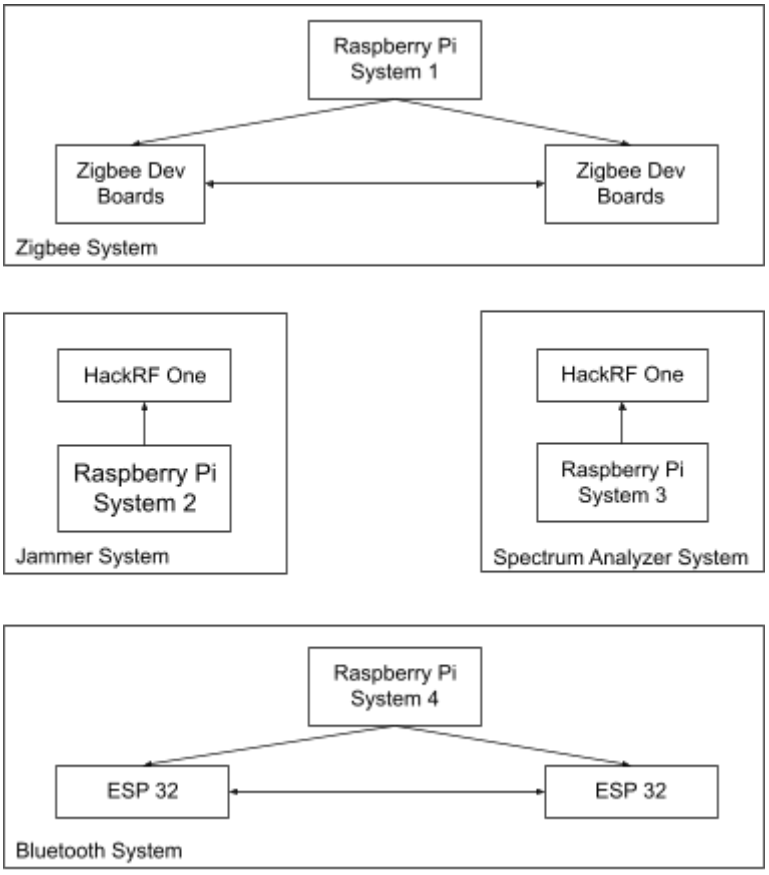

Figure 2: Diagrams of the Bluetooth, Zigbee, and Software Designed Radio Systems

2.2 Product Functions

2.2.1 The product should receive Bluetooth signals.

2.2.2 The signals should be identified and collected.

2.2.3 The HackRF One should intercept Bluetooth signals not intended for the device.

2.2.4 Using penetration methods to extract information.

2.2.5 The product should store and process information.

2.2.6 GNU Radio should display a graphical view of the transmitted frequency.

2.2.7 HackRF Spectrum Analyzer will monitor the frequency between 2.4 GHz and 2.5 GHz.

2.3 User Classes and Characteristics

Cybersecurity researchers who have previous knowledge of wireless communications would be responsible for the review of the results. With the information, they would be able to perform an in-depth analysis of the findings to further improve the security of the signals.Presentation of the information in an easy-to-understand form to the general person is also a necessecity.

2.4 Operating Environment

The software will operate running Raspberry Pis that display the information on a mini monitor through an HDMI connection. Using Python, a user will interact with the Raspberry Pis which must be able to connect to Kali Linux 2023.3 to collect data as well as to interact with it. Bluetooth 5.0 will create and send signals between the multiple Pis.

2.5 Design and Implementation Constraints

- Systems must be able to interact with Bluetooth devices within 5 feet
- System must be able to operate at the start of a Bluetooth connection
- System must be able to interact with Zigbee devices within 5 feet
- System must be able to operate at the start of a Zigbee connection
- HackRF must be able to operate on same frequency as Bluetooth and Zigbee

2.6 User Documentation

2.6.1 Zigbee Development Board User Manual: https://www.ti.com/lit/ug/swru209b/swru209b.pdf

2.6.2 ESP32 User Manual:

https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

2.6.3 HackRF One documentation:
https://hackrf.readthedocs.io/en/latest/

2.6.4 Raspberry Pi Documentation:
https://www.raspberrypi.com/documentation/

2.6.5 Kali Linux Documentation:
https://www.kali.org/docs/

2.6.6 HackRF Spectrum Analyzer GitHub:
https://github.com/pavsa/hackrf-spectrum-analyzer

2.7 Assumptions and Dependencies

2.7.1 There are no external RF signals on the same or similar frequencies.

2.7.2 Transmitter and receiver are within 5 feet of each other.

2.7.3 There are no objects or walls between transmitter and receiver.

2.7.4 The user has downloaded and installed all required software.

3. External Interface Requirements

3.1 User Interfaces

3.1.1 Arduino IDE: Arduino IDE will be used to push code to the ESP32. See Figure 3 below.

Figure 3: Arduino IDE

3.1.2 Zigbee Software: Zigbee Development software will be used to control the Zigbee Boards to send a signal.

3.1.3 HackRF Tools: HackRF Tools will be used to control the HackRF One.

3.1.4 GNU Radio: GNU Radio will be used to send RF signals via the HackRF One. Users can configure what frequency and range the signals are being sent as well as view Bluetooth packets and signals. See Figure 4 below.

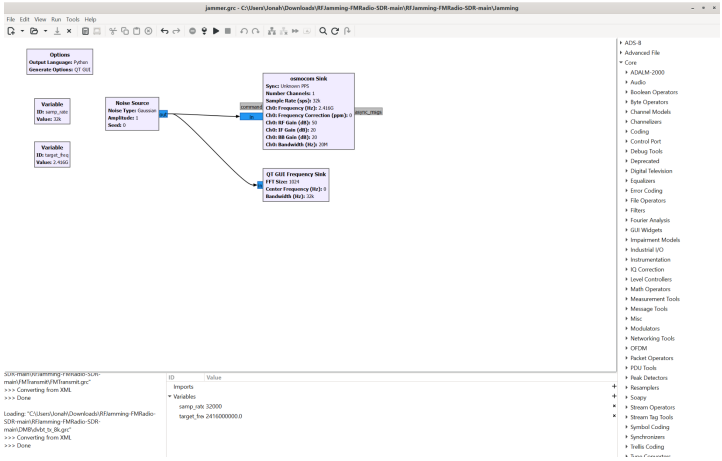

Figure 4: GNU Radio Jammer File

3.1.5 HackRF Spectrum Analyzer: The spectrum analyzer will be used to determine if the HackRF One is correctly transmitting on the right frequency as well as view the data transmitted by Bluetooth and Zigbee devices. See Figure 5 below.

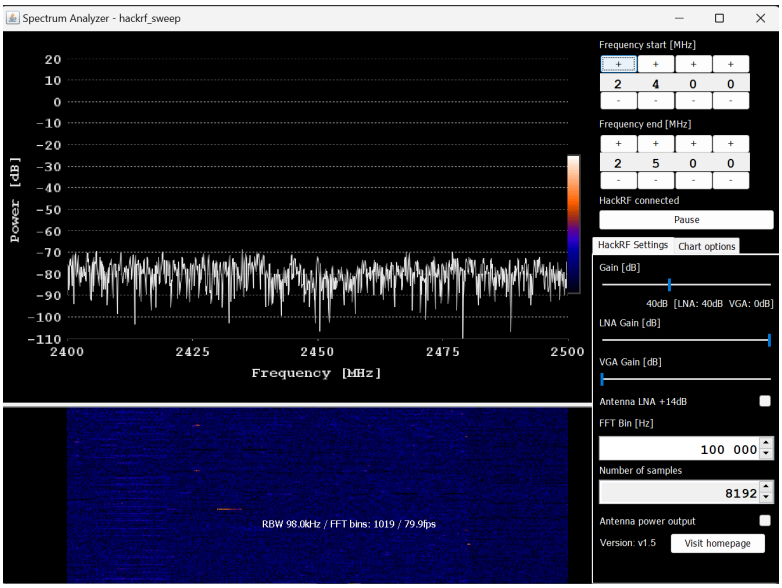

Figure 5: GNU HackRF Spectrum Analyzer

3.2 Hardware Interfaces

The Raspberry Pi system will contain all of the components needed to operate the Raspberry Pi as a standalone computer - consisting of the power source, display, keyboard, and mouse. An SD card will be used to run the applications and operating system of the Raspberry Pi. Four Raspberry Pis will be used to run the ESP32 and Zigbee modules, as well as one more Raspberry Pi for the HackRF Module.

3.2.1 Required Hardware:

3.2.1.1 Two Software Defined Radio (HackRF One)

3.2.1.2 Two ESP32 Bluetooth Modules

3.2.1.3 Two Zigbee Development Modules

3.2.1.4 Four Raspberry Pis

3.2.1.5 Four SD Cards

3.2.2 Preloaded operating system and software is to be loaded on the SD cards for use on Raspberry Pis.

3.3 Software Interfaces

Each Raspberry Pi will be preloaded with Kali Linux on the SD card, along with the software to run the required module. Two Raspberry Pis will have Arduino IDE to run the ESP32s, two will have the Zigbee Development Software, and one will have the HackRF Tools and GNU Radio installed.

3.3.1 Four Raspberry Pis will be loaded with Kali Linux (or other software) to run their respective hardware.

3.3.2 HackRF Tools will be used to establish a connection with the HackRF One

3.3.3 GNU Radio will be used to generate and send RF signals using the HackRF One.

3.3.4 Data for the ESP32 and Zigbee Boards will be sent via serial from the Raspberry Pis.

3.3.5 HackRF Spectrum analyzer will be used to measure the data transmitted by the jammer using a HackRf One.

3.4 Communications Interfaces

Each Raspberry Pi will function using an operating system and software through an SD card. The Raspberry Pis will connect to either the ESP32 or Zigbee Development boards through USB type A to USB micro B. The HackRF will connect to the Raspberry Pi via USB A to USB type C. The two ESP32 modules will communicate via Bluetooth, and the Zigbee Development boards will communicate via Zigbee. The HackRF will send signals through a set frequency to intercept or

block Bluetooth and Zigbee signals.

3.4.1 The HackRF, Zigbee, and ESP32 Bluetooth modules will be connected to the Raspberry Pis via USB type A to USB micro B.

3.4.2 The ESP32 modules will communicate via Bluetooth.

3.4.3 The Zigbee modules will communicate via Zigbee.

3.4.4 Peripherals such as the keyboard and mouse will connect via USB type A

3.4.5 The displays will be connected via micro HDMI to HDMI

3.4.6 The Raspberry Pis will be powered via a USB type C

4. System Features

4.1 Test Plan

4.1.1 Description and Priority:

The test plan provides the team with a plan of action to find and experiment with vulnerabilities and security flaws in Bluetooth and Zigbee.

4.1.2 Stimulus/Response Sequences:

There are no user actions or system responses for this feature.

4.1.3 Functional Requirements:

4.1.3.1: The test plan shall provide team members with a plan to test vulnerabilities in the Bluetooth protocol.
4.1.3.2: The test plan shall provide team members with a plan to test vulnerabilities in the Zigbee protocol.

4.2 Bluetooth Connectivity

4.2.1 Description and Priority:

Raspberry Pi computers connected to ESP32 development boards will be connected using Bluetooth. Implementing this feature is a high priority because it is integral to the identity of the project.

4.2.2 Stimulus/Response Sequences:

When a Raspberry Pi and ESP32 development board combination that is already connected to the network, powered on, and is in range of the network, it will connect to the network. When a Raspberry Pi and ESP32 development board combination that is not already connected to the network, and is in range of the network, is powered on, it will be able to be added to the network using Bluetooth pairing.

4.2.3 Functional Requirements:

A Raspberry Pi will be able to transmit and receive data with another Raspberry Pi using Bluetooth

4.3 Zigbee Connectivity

4.3.1 Description and Priority:

Raspberry Pi computers connected to Zigbee development boards will be connected using Zigbee. Implementing this feature is a medium priority because it is considered a lower priority than Bluetooth connectivity.

4.3.2 Stimulus/Response Sequences

When a Raspberry Pi and Zigbee development board that is already connected to the network and in the range is powered on, it will connect to the Zigbee network. When a Raspberry Pi and Zigbee development board that is not already connected to the network and in range is powered on, it will be able to connect to the existing Zigbee network.

4.3.3 Functional Requirements

A Raspberry Pi can transmit and receive data with another Raspberry Pi using Zigbee

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 Bluetooth and Zigbee connection will attempt to re-establish after the connection is lost.

5.1.2 Bluetooth transmission will be either partially or fully stopped.

5.1.2 Bluetooth service will be interrupted for a minimum of 1 second.

5.1.4 Zigbee service will be interrupted for a minimum of 1 second.

5.1.5 Bluetooth data rate will be a minimum of 1 MBps

5.1.6 Bluetooth data rate will be a maximum of 2 MBps.

5.1.8 Zigbee data rate will be a minimum of 20 kBps

5.1.9 Zigbee data rate will be a maximum of 250 kBps.

5.1.10 RF signals will be transmitted at the same frequency as Bluetooth.

5.1.11 RF signals will be transmitted at the same frequency as Zigbee.

5.1.12 Raspberry Pis will contain 128 GB of storage.

5.1.13 Service interrupt will occur within 1 second of execution.

5.2 Safety Requirements

5.2.1 All Raspberry Pis will remain in cases with fans running while powered.

5.2.2 All devices will be powered down before handling.

5.3 Security Requirements

5.3.1 The system will not be used within the range of extraneous electronic devices.

5.3.2 The system will only be used in a controlled environment.

5.3.3 All testing will be done on a closed network.

5.3.4 All transmitted RF signals must remain within 2.4 GHz and 2.5 GHz.

5.3.5 Signal data will be deleted after use.

5.3.6 Devices unrelated to the testing will be removed from the room.

5.3.7 Personal devices will not be connected to any of the systems.

5.4 Software Quality Attributes

5.4.1 RF signals will be transmitted between 2.4 GHz and 2.5 GHz.

5.4.2 Bluetooth signals will be transmitted between 2.4 GHz and 2.5 GHz.

5.4.3 Zigbee signals will be transmitted between 2.4 GHz and 2.5 GHz.

5.4.4 The Spectrum Analyzer will monitor a minimum frequency of 2.4 GHz.

5.4.5 The Spectrum Analyzer will monitor a maximum frequency of 2.5 GHz.

5.5 Business Rules

5.5.1 All testing must be preapproved in a controlled environment.

5.5.2 Signals will transmit a maximum of 20 feet.

6. Other Requirements

No further requirements not previously stated.

Appendix A: Glossary

Bluetooth - Short-Range Wireless Data Transmission Standard
Zigbee - Wireless Personal Area Network Standard
RF - Radio Frequency
IOT - Internet of Things
SDR - Software Defined Radio
USB - Universal Serial Bus
SDR - Software Defined Radio