

## Replay attacks and sniffing in Bluetooth low energy communications with mobile phone

Juan Sebastian Orozco Duran, Edith Paola Estupiñan Cuesta, Juan Carlos Martínez Quintero  
Telecommunications Engineering Program, Faculty of Engineering, Universidad Militar Nueva Granada, Bogotá, Colombia

### Article Info

#### Article history:

Received Feb 8, 2025  
Revised Aug 15, 2025  
Accepted Sep 11, 2025

#### Keywords:

Bluetooth  
Bluetooth low energy  
Replay attack  
Security  
Sniffing  
Software defined radio

### ABSTRACT

This article analyzes vulnerabilities in Bluetooth low energy (BLE) connections in smartphones against replay and tracking attacks using software defined radio (SDR), evaluating four scenarios with BLE headsets and smartphones from different manufacturers through HackRF one, GNU radio, and Wireshark. In scenario 1, the advertising message ADV\_NONCONN\_IND was captured and retransmitted, generating persistent and deceptive pairing pop ups on smartphones. In scenario 2, fake pairing request signals were replicated to simulate a connection attempt, causing interface errors and deceptive notifications for the user. In scenario 3, complete pairing sequences were captured and replayed, producing false connection alerts and fabricated information such as battery level indicators from non existent devices. In scenario 4, passive tracking enabled the extraction of sensitive data during the pairing process, including ADV\_IND packets, media access control (MAC) addresses, frequencies, manufacturer identifiers, and transmission power levels. A total of 93 successful and 123 failed attacks were recorded, with abnormal behaviors observed such as false pairing requests and manipulated device data, exposing users to risks of identity spoofing, denial of service (DoS) attacks, or targeted interference. The results highlight BLE protocol weaknesses against radio frequency (RF) based attacks and demonstrate the potential of SDR tools as powerful instruments for wireless protocol validation and cybersecurity research.

*This is an open access article under the [CC BY-SA](#) license.*



### Corresponding Author:

Juan Carlos Martínez Quintero  
Telecommunications Engineering Program, Faculty of Engineering  
Universidad Militar Nueva Granada  
Bogotá, Colombia  
Email: [juan.martinezq@unimilitar.edu.co](mailto:juan.martinezq@unimilitar.edu.co)

## 1. INTRODUCTION

Bluetooth, developed by the Bluetooth Special Interest Group, is a short range wireless personal area network (WPAN) technology that enables data sharing or transmission over distances of up to 200 m, with a maximum transmission speed of 50 Mbps in versions 5.0 to 5.3 [1]. Its development areas and challenges include improving data transmission rates, signal range, security mechanisms, and energy efficiency [2]. In the case of Bluetooth low energy (BLE), first implemented in Bluetooth version 4.0, typical transmission power levels range between -20 dBm and +10 dBm, with some variations depending on energy requirements, and an average connection establishment time of 3 to 7 ms. BLE remains inactive until data transmission is required [3].

The Bluetooth connection process begins with pairing, during which request messages, acceptance signals, security keys, and pairing confirmation data are exchanged. From a security perspective, Bluetooth implements mechanisms such as PIN code encryption, elliptic curve Diffie-Hellman (ECDH), and low

energy (LE) secure connections to enhance key generation and improve the pairing process [4]. The use of software defined radio (SDR) has significant potential for uncovering vulnerabilities by interacting with the radio signals involved in these processes, potentially leading to service unavailability, identity spoofing, or even the extraction of critical or confidential information [5].

The physical layer of Bluetooth technology is vulnerable to various types of attacks because wireless transmission allows interaction not only between the intended devices but also with unauthorized passive or active devices. SDR being reconfigurable facilitates interaction with radio signals at different frequencies, enabling demodulation, storage, information extraction, and, in some cases, Bluetooth signal spoofing. Devices such as the HackRF one have the necessary capabilities to generate and capture Bluetooth signals with bandwidths covering multiple channels [6]-[10]. This equipment enables sniffing across different Bluetooth channels to obtain critical information during device discovery and pairing. Additionally, it is possible to store and later retransmit signals to trick a Bluetooth device into believing that a nearby node is present, a technique known as a replay attack. This research analyzed the security of Bluetooth connections by capturing and retransmitting radio frequency (RF) signals using SDR. Unlike previous studies focused on data interception or man-in-the-middle (MITM) attacks, such as [11]-[15] which did not explicitly address RF signal replay, this study demonstrates how signal replay can exploit vulnerabilities in the mobile device pairing process.

The results obtained indicate that the analyzed smartphones are vulnerable to user experience (UX) disruptions such as repeated connection request pop ups, the appearance of fake connections showing battery levels of disconnected headsets or other internet of thing (IoT) devices, failed pairing attempts, and the detection of non-existent devices. A key deficiency identified in the reviewed literature is the limited attention paid to BLE replay attacks specifically targeting mobile UX. This study highlights this gap by providing empirical evidence that visual and functional elements of mobile systems can be manipulated through falsified signals, such as fake pairing notifications or falsified battery indicators, without the need for cryptographic keys or direct intervention in the authentication process. Furthermore, most prior research has focused on theoretical attack models or non mobile BLE enabled devices (e.g., IoT devices), rather than on commercial smartphones and headsets, which are the primary targets addressed in this work. This study provides empirical evidence of weaknesses in the implementation of the BLE protocol and emphasizes the need to strengthen its authentication mechanisms. As potential solutions, it is recommended to integrate packet filters to detect suspicious retransmissions, employ dynamic keys during the pairing process, and implement firmware updates that improve the management of incoming connections. However, the limited processing power and energy constraints of many IoT devices restrict the adoption of more advanced security measures. Nevertheless, manufacturers can leverage these findings to design future versions of Bluetooth that are more secure and resilient to RF based attacks. The objectives of this research focus on conducting a documentary exploration, then defining the scenario, and testing and analyzing the results of executing a replay attack using a HackRF one, with the aim of capturing and retransmitting signals from one or more headsets or IoT devices.

The results obtained (scenario 1) show that the massive retransmission of pairing requests affects the availability of the mobile phone by overloading the interface with recurring notifications, saturating the user's access to the mobile phone. The scenario 2 demonstrates that the attack not only replicates the headset's visibility signal, but also intercepts and retransmits the manually activated pairing process signal, the phone attempts to complete the connection process, but fails due to the lack of a valid response from the spoofed headset, but with the possibility of capturing and retransmitting the headset's response signal. In the scenario 3, the attack successfully retransmitted the message displaying the headset's battery levels, presenting misleading information in the user interface (UX) even when the original devices were not paired. Finally, sniffing techniques captured ADV\_NONCONN\_IND packets, allowing the extraction of key connection information from the headsets, thus exposing security vulnerabilities in the BLE protocol. This research was conducted under limited bandwidth conditions, which could have affected the packet capture success rate. The focus was limited to replay and sniffing attacks, without addressing other BLE vulnerabilities. Future scenarios open the possibility of further exploring vulnerabilities in BLE and other wireless technologies at the physical layer.

Finally, the article is organized as follows: section 1 presents the introduction. Section 2 describes the materials and methods used, including a documentary review of related research and an outline of the different experimental scenarios for the proposed attacks. Section 3 provides an analysis of the results and discussion. Finally, section 4 presents the conclusions.

## 2. METHOD

Figure 1 illustrates the method used in the research. Initially, in Phase A, a literature review is conducted on Bluetooth technology attacks, including MITM, denial of service (DoS), jamming, and replay, among others. In Phase B, the test scenario is presented, detailing the execution of the attack. Finally, in Phase C, an analysis of the obtained results is performed, followed by the conclusions.

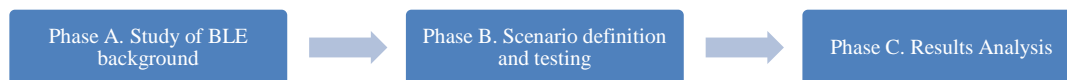


Figure 1. Research method

### 2.1. Phase A. study of Bluetooth low energy background

The BLE technology implements a model consisting of a physical layer, a link layer, and an application layer. At the physical layer, it operates in the 2.4 GHz band and uses GFSK modulation, similar to classic Bluetooth, although the frequency deviation modulation index is different. For BLE, it is 0.28, while for classic Bluetooth, it is 0.35. Classic Bluetooth has 79 channels, while BLE has 40 channels, as shown in Figure 2. The channel spacing is 2 MHz, and it is noteworthy that channels 37, 38, and 39 are advertising channels, used for request and data exchange during the connection process [16].

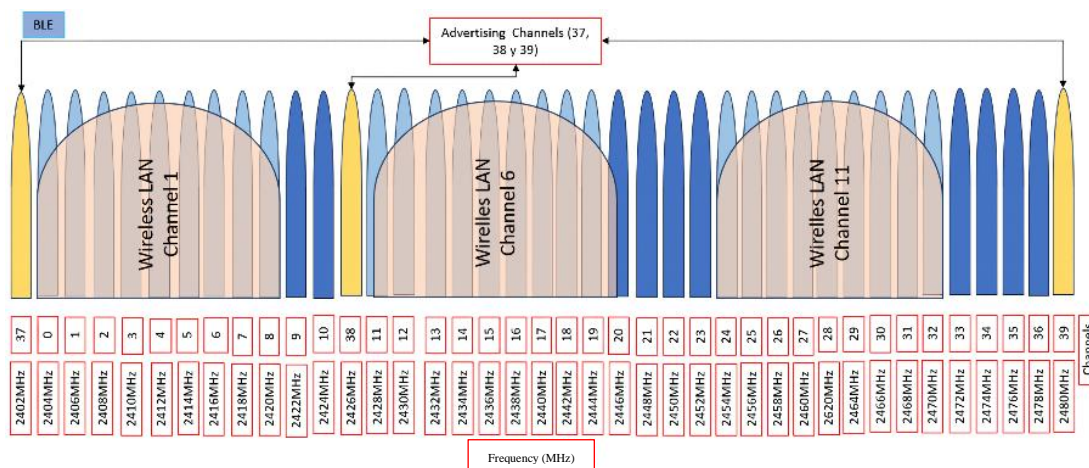


Figure 2. BLE channels

The link layer is responsible for pairing one or more devices, managing different broadcast or advertising states. For connection requests and establishment, there are four types of broadcast messages: connectable undirected advertising, connectable directed advertising, nonconnectable advertising, and discoverable advertising. These messages are used to announce the availability of a device or transmit data [17]. Figure 3 shows the broadcast message type ADV\_NONCONN\_IND, corresponding to nonconnectable advertising, generated by devices such as headsets on the advertising channels (37, 38, and 39). Figure 4 presents the structure of the BLE advertising channel packet, which consists of four fields. A preamble (1 byte) starts the packet, followed by the access address (4 bytes), which identifies the destination device. The PDU (2-39 bytes), responsible for data exchange, is divided into two parts: a header (2 bytes) and the payload (0-37 bytes). Finally, the CRC field (3 bytes) ensures the integrity of the transmitted data [18].

#### 2.1.1. Replay attack and sniffing

A replay attack is a technique in which a malicious actor captures legitimate signals transmitted between two devices and retransmits them without modification, aiming to deceive the receiving device into believing that the signal originated from the legitimate sender. In the context of BLE, this type of attack can be used to simulate pairing attempts, trigger false notifications, or cause UX errors without requiring access to encryption keys or compromising authentication protocols. These attacks exploit the lack of temporal verification or authenticity checks in certain BLE protocol messages [19]. A sniffing attack involves the

passive interception of data packets transmitted over the air, without modifying the communication flow. In the context of BLE, SDR tools were employed to capture packets transmitted during the advertising, connection, or pairing processes. Although the attacker does not actively participate or modify the data, the analysis of the captured traffic may expose sensitive information from the devices involved in the pairing process and connection establishment [20].

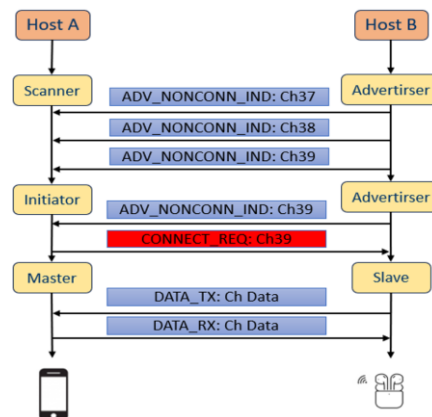


Figure 3. Addressing of advertising packets in the communication BLE

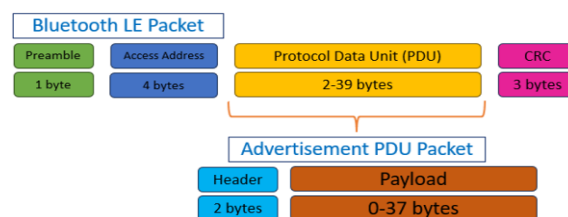


Figure 4. Packet type BLE for advertising

### 2.1.2. Bluetooth low energy documentary analysis

After completing the literature review, significant background information on attacks targeting the BLE architecture at the physical, link, and application layers was identified. In the case of link layer attacks, the objective is to exploit vulnerabilities in the connection processes, capture specific information packets, and perform spoofing. In [11], a MITM attack was executed on an IoT smartwatch and a smartphone, capturing encryption keys and exploiting vulnerabilities in the host authentication phase. In [12], a MITM attack was carried out on a smartphone and an IoT BLE wristband, exploiting the "just works" pairing vulnerability and gaining control of the wristband. In [13], a MITM attack captured fingerprint information packets from a mobile device connected to an IoT application from the Google Play Store. In [14], a MITM attack targeted a scientific robotic device called Qoopers, where a low power (LE) framework and tracker were installed, successfully capturing packets during data transfer to the robot. In [15], a MITM attack was carried out against data transmitted between a smartwatch (Mi Band 2) and an Android device. The attacker managed to pair the smartband and access user data, including sensor information. In [21], a blueprinting attack was carried out against the BLE protocol on an IoT shopping cart with Bluetooth connectivity, capturing the transmitted data to gather information about the shopping route and purchases. In [22], an attack vector that exposes mobile operating systems was revealed, obtaining the device's media access control (MAC) address to create an exploit that causes a Bluetooth network encapsulation protocol (BNEP) connection failure. In [23], a BlueBorne attack was executed against the logical link control and adaptation protocol (L2CAP), session description protocol (SDP), security manager protocol (SMP), and BNEP protocols, generating malicious packet parameters that allowed access to the victims' mobile devices. In [24], a Blacktooth attack targeted mobile devices and headsets, capturing information packets and performing brute force attacks to decrypt the pairing key. In [25], an attack was performed on the "just works" pairing mode of the BLE protocol on IoT devices in a home, allowing the attacker to send commands to the devices. Finally, [26] presents a lock system that uses security filters in a facial recognition application to unlock a

house door. These studies highlight the diverse range of attacks and vulnerabilities of BLE technology, from pairing issues to security risks in home IoT devices and mobile operating systems.

For attacks affecting the physical data and link layers, [27] shows the execution of MITM and DoS attacks on an IoT device unlocking a garage, capturing the packets sent at the physical layer to replicate and execute unauthorized commands. In [28], two attacks: MITM and DoS were performed on two Bluetooth enabled locks, successfully capturing commands sent from an IoT device. In [29], sniffing and DoS attacks were executed, exploiting a vulnerability in the SDP connection to obtain the MAC address of a device and launch a DoS attack. In [30], two attacks: sniffing and DoS, were performed targeting a mobile device. The data capture was successful, followed by the launch of a DoS attack, affecting the link management protocol (LMP) connection table. In [31], two attacks, Bluesmack and sniffing, were carried out on a pulse oximeter, a blood pressure monitor, and an electrocardiogram that send patient data over Bluetooth to a tablet using IoT. Successful monitoring of packet traffic during the connection caused outages by breaking the connection. In [32], an injection attack was executed, capturing pairing data using sniffing and then performing a DoS attack to deny service connecting to the original IoT device. In [33], sniffing and brute force attacks were executed on two paired mobile devices exchanging information. Packets were successfully captured and the IoT device information was decrypted. In [34], a brute force and DoS attack was carried out on a smartwatch, where user information was decrypted, followed by DoS on the device. At the application layer, the goal is to exploit vulnerabilities between the user application and the network. In [35], a Bluejacking attack was performed on a user's mobile device with the aim of entering the network by impersonating commands and orders from the original user. In [36], a Bluesnarfing attack was executed on a mobile device, where the vulnerability of the SDP protocol was exploited to obtain the MAC address of the target device and execute commands to modify personal data.

These studies highlight several conclusions:

- References [11]-[15], [21] present attacks targeting vulnerabilities in BLE, while [22]-[24], [26] demonstrate attacks on protocols such as L2CAP, SDP, SMP, OBEX, BR/EDR, LMP, BD/ADDR, and BNEP.
- From the theoretical references in [33], [34], vulnerabilities directed at the BLE protocol are evident. In [22]-[27] vulnerabilities targeting protocols such as L2CAP, SDP, LMP, BD/ADDR, and buffer issues are exploited.
- References [35], [36] focus on vulnerabilities in L2CAP, SDP, and RFCOMM protocols. Additionally, interesting mitigation proposals and mechanisms have been addressed in this scientific research.

Table 1 presents references of attacks on the BLE physical layer. In [6], [7], the objective is to identify which Bluetooth device is transmitting at a specific time using fingerprinting techniques and it is worth noting that they use GNU radio and HackRF one to perform MITM attacks and penetration tests with capture or replay. In [8]-[10], the use of retransmission attacks and successful capture of BLE packets to identify specific mobile devices and their MAC address are described.

Table 1. Investigations related to attacks using SDR

Research	Type of attack	Type of hardware and software	Description
Penetration tests for BLE and Zigbee using the SDR [6]	Penetration tests	GNU radio, R820T RTL2832U, Aispy R2, HackRF one, and Lima DEG y HojaRF	SDR was employed for penetration testing, capturing frequency hopping to extract MAC addresses and data connection information
MITM attack simulation on low energy wireless devices using SDR [7]	MITM	GNU radio, iBEACON, HackRF one, and RFX2400 y FPGA USRP	A MITM attack and sniffing were conducted on an iPad and a medical heart monitor, capturing information packets on channel 37 for subsequent retransmission
Bluetooth devices fingerprinting using low cost SDR [8]	Fingerprinting	LimeSDR Mini, Ubertooth, GNU radio, and BlueID	Fingerprinting techniques were employed, successfully capturing BLE packets to identify specific Android devices and the device's MAC address
Analog physical layer relay attacks with application to Bluetooth and phase-based ranging [9]	Relay attack	Primary and secondary relay antennas	A relay attack was carried out, successfully opening the smart lock of a car by retransmitting access and extending the unlocking range
Evaluating physical layer ble location tracking attacks on mobile devices [10]	Fingerprinting	BLE Shipset and 20 ESP32 Wi-Fi+BLE y 20 TI CC2640 BLE only chip sets	Security tests were conducted on mobile devices, success fully obtaining the wireless location beacon through the fingerprinting of the devices

Table 2 classifies the main BLE mitigation mechanisms at the application, link, and physical layers. At this stage, the most relevant mitigation proposals and mechanisms addressed in this scientific research have been identified. Table 2 details the most prominent mitigation mechanisms for BLE at several layers:

application, link, and physical. At the link layer, as suggested by [11]-[13], [19] along with [15], [22]-[24], [31], [32], the implementation of security updates and strategies, such as blockchain, is recommended to mitigate attacks, especially at the connection level. Several studies propose raising user awareness to improve secure practices and detect communications from insecure devices. Similarly, at the application layer [33], [34] propose improvements in the security of the interaction between the user's application and the network. However, it is crucial to highlight, for this article, the relevance of the proposals presented in [6], [7], [35], [36], which offer security mechanisms aimed at preventing attacks in the physical layer by improving transmission encryption and generating an alert alarm in the event of anomalies and external manipulation.

Table 2. Research related to mitigation mechanisms in BLE

Reference	Layer OSI model	Attack	Mechanisms security	Limitation	Threat description
[11]-[15]	Link	MITM	Improve user authentication, implement end to end measures, and apply hash code algorithms.	Passive attacks that are difficult to detect, lack of user awareness, and communication with untrusted devices.	Unauthorized interception and manipulation of communications between two devices, posing risks to data integrity and confidentiality.
[21]-[34]	Link	Blueborne, bluetooth, and sniffing	Proposes software updates, a strategy to reject an excessive number of requests, and the implementation of blockchain technology.	Dependency on firmware and software updates, passive and stealthy attacks, difficulty in detecting communications from insecure device.	Remote code execution or passive surveillance through vulnerabilities in BLE stack, enabling attackers to gain access without user interaction.
[35], [36]	Application	Bluejacking and bluesnarfing	Keep the device with the latest updates and reduce the number of unsolicited messages.	Lack of control in receiving unsolicited messages and social ignorance of the risks of not protecting information.	Exploits in application layer services to send unwanted messages (Bluejacking) or extract private data (Bluesnarfing) from mobile Bluetooth devices.
[6]-[10]	Physical	Sniffing and jamming	Improve encryption in transmission, detect jamming such as CRPA, and implement warning alarms to alert the user of anomalies and external manipulations.	The difficulty of detecting a malicious SDR, the anonymity of the attacker, and the lack of tools for visibility and anomaly detection.	Physical layer manipulation that captures BLE signals (sniffing) or disrupts communication (jamming), exploiting the lack of protection at this layer.

## 2.2. Phase B. scenario definition and testing

Three scenarios were defined based on how the mobile device responds to the retransmission of different signals initially transmitted by the earphones. A final scenario was proposed where no signals are retransmitted, and only the content of the advertising channel frames is analyzed. Table 3 presents the equipment used to implement the four proposed scenarios: three replay attack scenarios and one final sniffing scenario. Figures 5-8 show the implemented scenarios.

Table 3. Technical characteristics hardware and software used

Equipment	Characteristics/installation requirements	Objective
HackRF one RTL device	Manufacturer great scott gadgets, operating frequency from 1 MHz to 6 MHz, antenna port (50 mA at 3.3 V), and open-source hardware	Capture and re-transmit the signal with GNU radio also capture BLE packets for sniffing
Laptop with GNU radio software	Operating system 1 windows 11, 16GB RAM, Ryzen 5 processor, and SSD 1 TB storage drive	To capture a signal and then replicate it and thus execute the replay attack
Virtual machine (Oracle VM VirtualBox)	Operating system 2 has Linux 22.04.3 LT with GNU radio v3.10.7.0 and Wireshark 4.2.2	BLE packet sniffing
Mobile devices: device A, B, C, D, E, F	IOS: 16.6.1 and 17.1.1; processor: A13 and A15 Bionic chip, 4 GB RAM, 64 GB storage, and Bluetooth 5.0; Android: 11, 12, 13, and 14; processor: MediaTek Helio P95, Snapdragon 778G, MediaTek Helio G88 and Qualcomm Snapdragon 865, 6 GB RAM, 158 GB storage and Bluetooth 5.0	2 victims device of IOS 4 victims device of Android
Earphones (A) second generation and Earphones (B) first generation and Earphones (C)	MAC (A): 6E:7F:90:2A:E2:F4, Chip H2, and Bluetooth 5.3; MAC (B): 96:65:C6:8C:BC:B6, Chip H1, and Bluetooth 5.0; MAC (C): 6C:D5:86:6A:DB:02 and Bluetooth 5.0	3 supplanted device

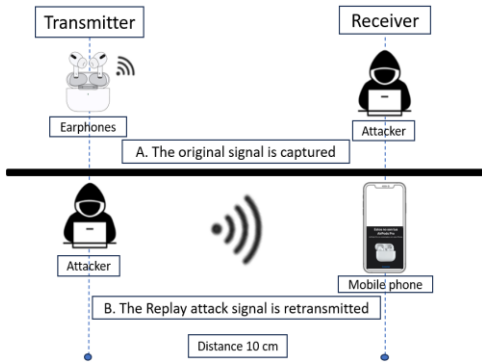


Figure 5. Scenario 1

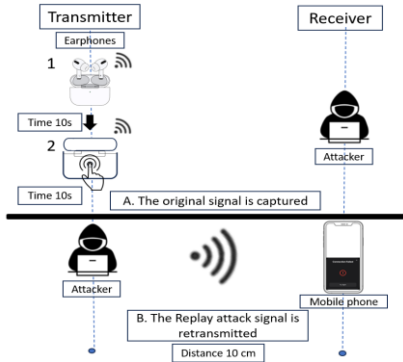


Figure 6. Scenario 2

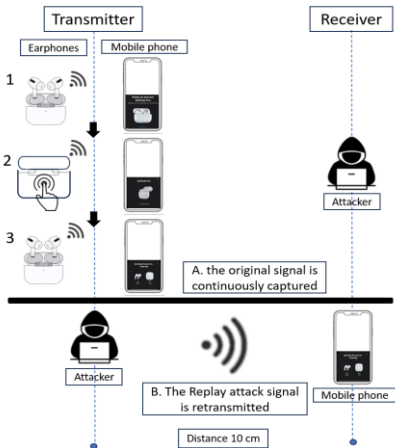


Figure 7. Scenario 3

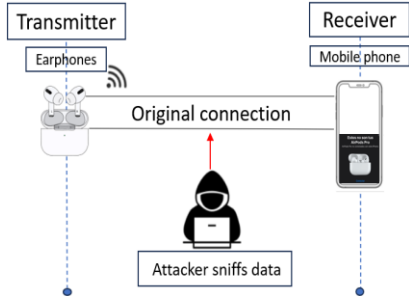


Figure 8. Scenario 4

Table 4 describes the attack methodology employed in this study, highlighting adversary capabilities, key assumptions, attack targets, and specific vulnerabilities exploited on each tested device. The analysis focuses on BLE pairing procedures on three commercial headsets (A, B, and C) and mobile devices. This formal model helps visualize how proximity based SDR and sniffing attacks can compromise integrity, authenticity, and UX without requiring encryption keys or privileged access.

Table 4. Attack methodology used in this study

Attacker capabilities	Assumptions	Objetive	Vulnerability and targeted devices
Access to SDR and sniffing. Short range proximity (less than 5 m).	<ul style="list-style-type: none"><li>No prior access to pairing keys or cryptographic information.</li><li>Communication is not encrypted during initial pairing.</li><li>The attacker can remain undetected during passive monitoring.</li></ul>	<p>Replay attack: deceiving the user by sending previously captured packets to simulate connection attempts, fake pairing processes, and fake device connection messages, triggering UI anomalies.</p> <p>Sniffing attack: passively capturing advertisement and pairing messages to identify devices, infer behavior, and reverse engineer protocol interactions.</p>	<p>Earphones A and B: susceptible to repeated false notifications and UI crashes during playback, but with limited tracking success.</p> <p>Earphones C: intercepted pairing confirmation message; reveals transmission power; allows for proximity attack planning.</p> <p>Mobile devices: affected by pop ups every 5 s, fake pairing processes, and connection messages with devices displaying fake battery levels, affecting the UX.</p>

Scenario 1 (Figure 5) involves capturing and retransmitting the ADV\_NONCONN\_IND message using the HackRF one. This message is broadcast by the IoT earphones when they are opened, allowing the phone to detect them. The radio signal of this message is transmitted through the advertising channels and captured by the HackRF one for 15 s. The capture is performed in a frequency range close to one of the advertising channels, without applying any signal processing before retransmission.



- The capture is activated with the HackRF one.
- The earphones are opened for 15 s, then closed, and the capture ends.
- The attack is executed to retransmit the 15 s signal.
- The mobile phone's Bluetooth must be turned on, even if the earphones are not within the coverage area at the time.

In scenario 2 (Figure 6), it shows a similar attack to scenario 1, the following steps were taken for this:

- A 20 s capture is initiated with the HackRF one.
- The IoT earphones signal is captured again for 10 s, followed by pressing the pairing button on the back of the case for another 10 s.
- Finally, the captured signal is retransmitted in the presence of the Bluetooth enabled phone, while the earphones are out of range.

In scenario 3, the entire pairing process between the device and the mobile phone is captured. The capture begins with the HackRF one, and the pairing procedure is performed a few seconds later, as shown in (Figure 7). The following steps were performed:

- Signal capture is initiated with the HackRF one.
- The IoT earphones are opened, and once the connection message appears on the phone, pairing is initiated following the software instructions.
- The button on the back of the earphones case is pressed.
- Pairing is completed, and the earphones are verified to be connected. The capture ends.
- After this, the attack is executed by retransmitting the signal in the presence of the phone with Bluetooth enabled and the earphones outside the coverage area.

To the scenario 4 (Figure 8), the HackRF is configured to a central frequency near the advertising channels. The software is set up to capture and decode the Bluetooth signal, and then the earphones are opened. The device's MAC address is captured and displayed, along with the information provided in the BLE broadcast packets.

Table 5 shows the frequencies and bandwidths used in each scenario, as well as the devices employed. In the replay attacks, central frequencies close to the advertising channels 37 at 2402 MHz, 38 at 2426 MHz, and 39 at 2480 MHz were chosen in order to capture at least one of these channels.

Table 5. Scenario description: device used (✓), device not used (X)

Scenario	Bandwidth test (MHz)	Frequency center test (MHz)	Earphones			Phone					
			A	B	C	A	B	C	D	E	F
1, 2, 3	12; 10	2405	✓	X	✓	✓	✓	✓	✓	✓	✓
	8 y 6	2423	✓	X	✓	✓	✓	✓	✓	✓	✓
		2477	✓	X	✓	✓	✓	X	X	X	X
4	10	2477	✓	✓	X	X	✓	X	X	X	X

Figure 9 shows how the variation in bandwidth allows for the capture of more or fewer Bluetooth channels near the advertising channel (highlighted in yellow). Some of the frames related to pairing, as well as frames sent when the devices are already connected, are transmitted over channels other than the advertising channels. Capturing multiple channels enables the inclusion of the information transmitted during the pairing process by both devices.

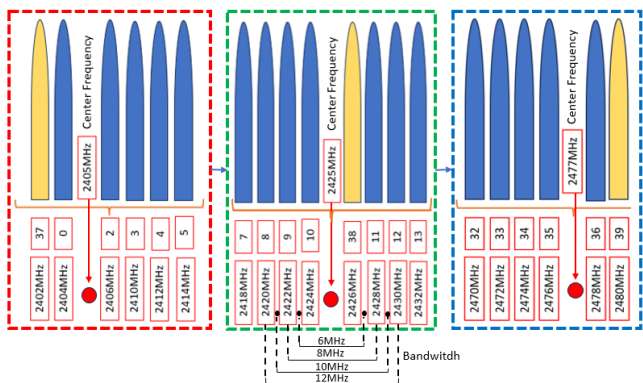


Figure 9. Identification of Bluetooth channels near the advertising channel



### 2.2.1. Phase B. scenarios configuration

For the execution of the replay attack, the GNU radio software is required to configure the radio interface of the HackRF one and manage the transmitted and received signals. Figure 10 shows the block diagram defined for capturing and storing the RF signal transmitted by the earphones for pairing after being converted to baseband. The obtained samples are stored at the sampling rate configured in the variable `samp_rate`. The osmocom source block allows configuring the HackRF one radio interface, and the samples are stored using the file sink block. The stored samples undergo no additional processing beyond what the hardware performs; that is, two channels of samples from the analog to digital conversion channels (I and Q channels) are stored. The QT GUI sink block displays an FFT graph of the signal, but this block can be disabled as the visualization is not critical to the attack. In Figure 11, the file source block is used to read the previously stored signal file to be retransmitted. The radio interface configuration is done with the osmocom sink block, which must have the same parameters used during the capture. The fast multiply const block functions as an amplifier for the signal. For passive sniffing in scenario 4, the ICE9 Bluetooth sniffer software [37] was used. It was configured with a central frequency of 2477 MHz, which is close to the advertising channel 39.

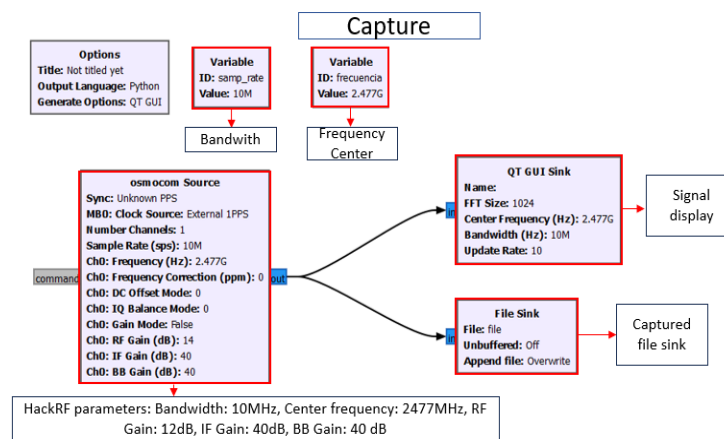


Figure 10. Capture blocks in GNU radio

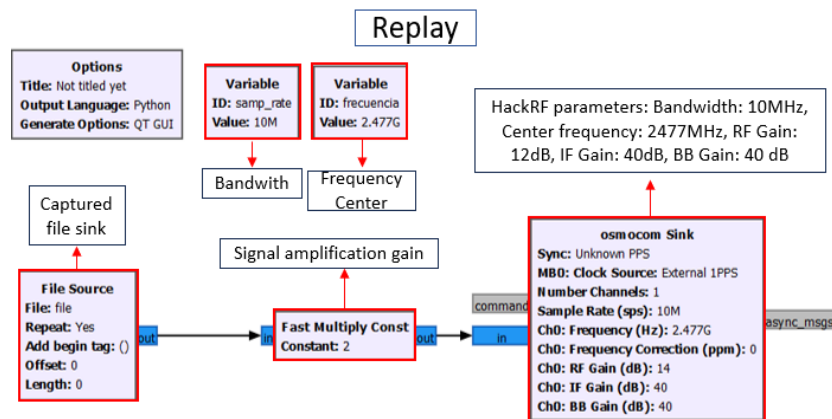


Figure 11. Replay blocks in GNU radio

## 3. RESULTS AND DISCUSSION

Table 6 summarizes the results obtained from the replay attack in each of the proposed scenarios. For device A, 21 successful tests and 15 unsuccessful tests were recorded; for device B, 22 successful tests and 14 unsuccessful tests; for device C, 8 successful tests and 28 unsuccessful tests; for device D, 12 were successful and 24 unsuccessful; for device E, 14 were successful and 22 unsuccessful; and for device F, 16 successful tests and 20 unsuccessful tests. It should be noted that BLE operates on an unlicensed band that is also used by Wi-Fi technology, and interference between these two technologies can cause the attack to fail in some cases.

Table 6. Results of the replay attack tests for the 3 scenarios yes (✓), no (x)

ID_Test	Frequency (MHz)											
	Scenario 1				Scenario 2				Scenario 3			
	12	10	8	6	12	10	8	6	12	10	8	6
DA_1_2405 MHz	✓	✓	X	X	X	✓	X	X	X	X	X	X
DA_2_2425 MHz	✓	✓	✓	✓	✓	✓	✓	X	X	✓	X	X
DA_3_2477 MHz	✓	✓	✓	✓	✓	✓	✓	X	✓	✓	✓	X
DB_1_2405 MHz	✓	✓	✓	X	✓	✓	X	X	X	X	X	X
DB_2_2425 MHz	✓	✓	✓	✓	✓	✓	X	X	✓	✓	X	X
DB_3_2477 MHz	✓	✓	✓	✓	✓	✓	✓	X	✓	✓	X	X
DC_1_2405 MHz	X	✓	X	X	X	✓	X	X	X	X	X	X
DC_2_2425 MHz	X	✓	X	X	X	✓	X	X	X	X	X	X
DC_3_2477 MHz	✓	✓	X	X	✓	✓	X	X	X	X	X	X
DD_1_2405 MHz	X	X	X	X	X	X	X	X	X	X	X	X
DD_2_2425 MHz	✓	✓	✓	X	✓	✓	✓	X	X	X	X	X
DD_3_2477 MHz	✓	✓	✓	X	✓	✓	✓	X	X	X	X	X
DE_1_2405 MHz	✓	✓	X	X	✓	✓	X	X	X	X	X	X
DE_2_2425 MHz	✓	✓	✓	X	✓	✓	✓	X	X	X	X	X
DE_3_2477 MHz	✓	✓	X	X	✓	✓	X	X	X	X	X	X
DF_1_2405 MHz	✓	✓	X	X	✓	✓	X	X	X	X	X	X
DF_2_2425 MHz	✓	✓	✓	X	✓	✓	✓	X	X	X	X	X
DF_3_2477 MHz	✓	✓	✓	X	✓	✓	✓	X	X	X	X	X

\* DA: device A, DB: device B, DC: device C, DE: device E, and DF: device F

Of the tests documented in Table 6, 93 attacks were successful and 123 failed. Many failures occurred on frequencies shared with other technologies such as Wi-Fi, causing interference that, combined with limited bandwidth, made it difficult to fully capture the advertising channels. The results suggest that replay attacks are most effective when using bandwidths above 10 MHz and center frequencies aligned with the BLE advertising channels, although higher bandwidths also require more system resources.

### 3.1. Scenario 1 results and analysis

The replay attack was successful for almost all bandwidths except when a center frequency of 2405 MHz was used with bandwidths of 6 MHz on phone B and 6-8 MHz on phone A. This is because the advertisement channel could not be fully buffered at these bandwidths. For the 12 MHz and 10 MHz bandwidth cases, retransmission of 4 to 6 solicitation messages was completed successfully within 5 to 10 s. With 8 MHz and 6 MHz bandwidths, 9 attempts were successful, sending 1 to 3 solicitation messages within 7 to 13 s, while 3 attempts were unsuccessful. Figure 12 shows the repetitive message displayed on phones B and A. Compared to previous studies such as [6], [11], that mainly focused on signal jamming or MITM simulations, this work provides empirical data on how bandwidth settings directly affect the success of replay attacks. Our findings suggest that HackRF one, despite being a low-cost device, is able to saturate Bluetooth interfaces with repetitive connection requests, underscoring the vulnerability of the BLE protocol to replay based DoS attacks. However, testing was constrained by limited bandwidth options, which might have affected the packet capture success rate. Environmental interference, particularly overlapping Wi-Fi signals, was not fully controlled, which might affect some results.



Figure 12. Scenario 1 results for devices A to F

### 3.2. Scenario 2 results and analysis

A total of 14 successful tests were obtained with phones A and B, successfully sending the connection request signal and completing the pairing process. Figure 13 shows that the user can interact with

the retransmitted signal up to a certain point. The connection error appears after some time when there is no response from the earphones, as it is the false signal retransmitted by the HackRF one. For 8 MHz and 6 MHz, 9 tests were unsuccessful because the bandwidths did not cover the entire advertising channel and did not cover enough data channels with some central frequencies.

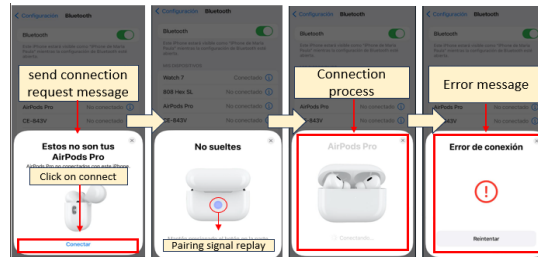


Figure 13. Scenario 2 results on dispositivo A

A total of 25 successful tests were obtained with phones C, D, E, and F, successfully sending the connection request signal and completing the pairing process. The results were the same as those obtained in the tests with phones A and B, with one key difference: on phones C, D, E, and F, there are devices that in their settings allow the user to disable the pairing request animation (C). Figure 14 shows that the user can interact with the retransmitted signal to a certain extent and also shows when the user blocks the request message, limiting it to searching only for Bluetooth devices for pairing, thus preventing unauthorized access to their device.

These findings demonstrate the feasibility of using SDR tools to simulate authentic BLE pairing sequences, deceiving the user and causing unintended system behavior. The attack exploits weaknesses in the BLE protocol's trust model, where the presence of a valid advertising and pairing signal can initiate the connection process, even without a legitimate peripheral responding. This behavior can increase the risk of social engineering or malware distribution, especially if an unauthorized device responds with crafted payloads. However, performance was bandwidth dependent: nine failed tests were performed with 6 and 8 MHz bandwidths, where the advertising and data channels could not be fully captured and replayed. This confirms that incomplete channel coverage reduces the probability of spoofing success and supports the conclusions from scenario 1 about the crucial role of signal width and center frequency. Compared to previous literature [7], [8] that explores passive sniffing or theoretical impersonation, this study provides empirical validation of active pairing impersonation attacks with minimal equipment. However, the experiment did not simulate full endpoint impersonation, such as providing fake service responses or exchanging encrypted data, which would require dynamic interaction with the mobile device.

### 3.3. Scenario 3 results and analysis

A total of 8 successful tests were conducted, successfully replicating or spoofing the pairing messages. In this case, the phone detects a false signal for the battery percentage of the earphones and their case. That is, the phone is led to believe for a time that it is connected to a device, when in fact, it is the HackRF one executing the replay attack. The unsuccessful tests were due to the selected frequencies, where the bandwidths were insufficient to cover all the necessary channels for communication. Figure 15 shows the battery level message provided by the phones. The tests were carried out on phones B and A, with identical results.

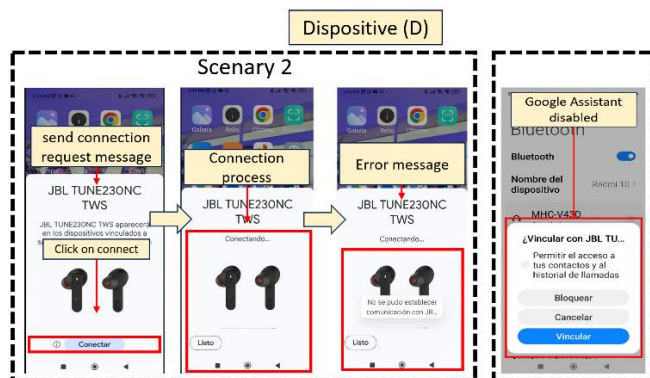


Figure 14. Scenario 2 results

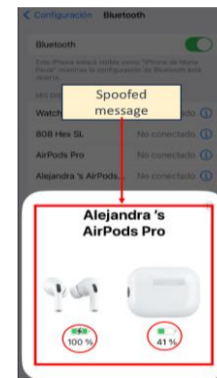


Figure 15. Scenario 3 results dispositivo A

In scenario 3, a more advanced spoofing strategy was tested, transmitting fake pairing messages along with fake battery level indicators for the earphones and their charging case. Using HackRF one, a total of 8 successful tests were performed with phones A and B, where the mobile device displayed battery status information, falsely indicating a connection with Bluetooth audio devices. This result demonstrates a critical vulnerability in the BLE protocol, where the smartphone interprets and displays falsified data from an unauthenticated source. Figure 15 illustrates the battery level display on the smartphone interface, mimicking what would be expected from real IoT Bluetooth earphones. These results are significant as they demonstrate that an attacker can not only activate the pairing interface but also transmit crafted metadata to manipulate the user's perception, making them believe they are connected to a known or trusted device. This goes beyond simple replay of connection attempts and suggests the possibility of deeper social engineering attacks utilizing device identity, battery state, and possibly falsified service capabilities. However, some tests failed, especially when using insufficient bandwidth to capture or replay all necessary BLE channels. This confirms a pattern observed in scenarios 1 and 2: limited bandwidth impacts channel coverage and reduces the fidelity of the replay attack. Compared to existing literature such as [9], [10], [35], [36], which often focuses on device discovery or tracking, this experiment extends the analysis by demonstrating manipulation of dynamic BLE attributes. The ability to inject fake battery data using SDR based replay has not been widely documented, highlighting a novel and practical risk to BLE based user trust systems.

The results indicate that the main cause of failures was related to the overlapping use of the 2.4 GHz band by Wi-Fi and the presence of nearby devices with Bluetooth enabled, which interfered with the capture of specific BLE advertising channels, particularly at narrower bandwidths (6–8 MHz). Devices (C, D, E, and F) exhibited a higher rate of failed attacks due to more robust resistance mechanisms, such as the suppression of false pairing requests and greater susceptibility to Wi-Fi interference at certain bandwidths. In contrast, devices (A and B) consistently displayed pairing requests even under weak or incomplete signals, making them more vulnerable to impersonation attempts.

### 3.4. Scenario 4 results and analysis

The BLE packets from the earphones A were captured, successfully obtaining and observing the source MAC address 6E:7F:90:2A:E2:F4, the destination broadcast address FF:FF:FF:FF:FF:FF, the channel through which the BLE RF packet was sent, which was channel 39, belonging to the advertising channels, and the channel frequency of 2480 MHz. This information enables the proper identification of the earphones (A), as shown in Figure 16.

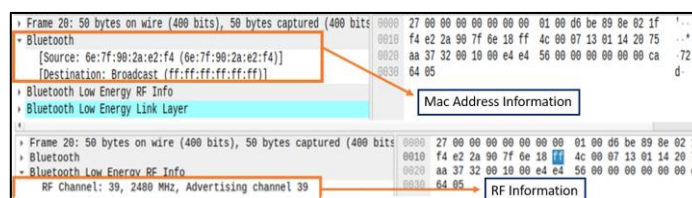


Figure 16. BLE RF level info of captured packet

Figure 17 analyzes the capture obtained by Wireshark of the BLE packet header during the sniffing process, where the earphones A and HackRF one are involved. The access address is identified with a value of 0x8e89bed6, which is a unique address used to establish communication between BLE devices and ensure proper synchronization during data transmission. The PDU Type field 0x2 is identified with an identification value 0010, which corresponds to the ADV\_NONCONN\_IND message generated during the pairing process. Specific data from the manufacturer of the earphones A and B is identified, including the ID 0x004c and CRC 0x5326a0. This information is useful for identifying the key data involved in a BLE pairing process and, in turn, could assist an attacker in launching potential spoofing attacks, compromising the transmitted information and the security of the network.

Once the sniffing was performed with phone C and earphones C, a different message packet type was obtained compared to phones A and B. In this case, the ADV\_IND packets were captured, which announce the presence of the device and provide general information about itself, which can be actively indicated by other receiving devices. This type of packet is essential for the discovery and connection between devices in a BLE network. Figure 18 demonstrates these obtained data. For the C headphones, Figure 19 shows data from manufacturer A with a 16-bit UUID (0x16) and from another manufacturer also with a 16-bit UUID, in addition to revealing the transmission power level (-11 dBm) of the C earphones. In



Figure 20, the capture of a SCAN\_RSP packet is observed. The "SCAN\_RSP" message or packet in BLE is used as a response to a scan request sent by another device. When a BLE device is in active listening mode and detects a scan request from another device, it may respond with a SCAN\_RSP packet. Table 6 summarizes the most relevant data obtained from phone and earphones B, and phone and earphones C when the sniffing process occurs during BLE pairing.

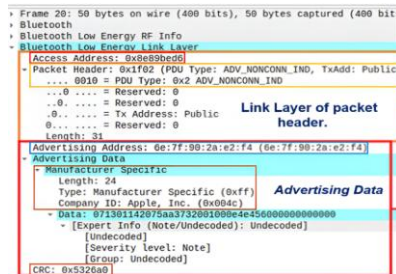


Figure 17. BLE information at the link level

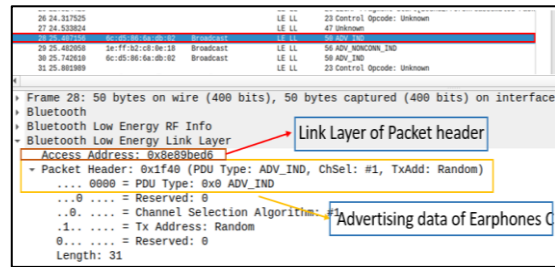


Figure 18. BLE RF level info of captured packet on dispositive I

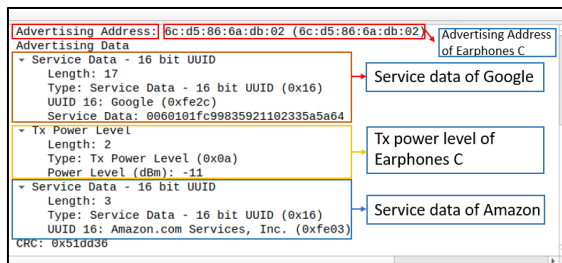


Figure 19. BLE information at the link level of earphone C

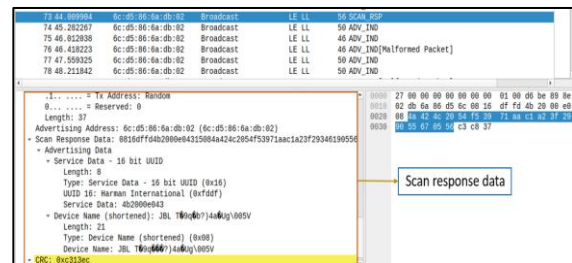


Figure 20. Scan response data BLE of earphones and dispositive F

In Table 7, a comparative analysis is presented between headset B and headset C during the BLE pairing process, highlighting findings obtained through sniffing. In the case of headset C, the pairing confirmation message from the devices ADV\_IND, which represents a significant security risk, as it could facilitate replay attacks or allow reverse engineering of the protocol without the need for active intervention. Furthermore, the transmission power of -11 dBm (TX power) was identified, allowing the approximate distance of the device to be estimated and, therefore, proximity based attacks to be planned more accurately.

Table 7. Information obtained from earphones (B) and (C)

Earphone (B)		Earphone (C)	
Identified parameters	Value	Identified parameters	Value
MAC source	6E:7F:90:2A:E2:F4	MAC source	6C:D5:86:6A:DB:02
RF channel	Advertising 39-frequency 2480 MHz	RF channel	Advertising 39-frequency 2480 MHz
Access address	0x8e89bed6	Access address	0x8e89bed6
Broadcast message type	0010-ADV_NONCONN_IND	Broadcast message type	0000-ADV_IND
CRC	0x5326a0	CRC	(0xfe03)
TX power level	Does not display the transmission power level	TX power level	Power (dBm) -11

#### 4. CONCLUSION

The HackRF one device demonstrated significant potential by allowing the capture and retransmission of signals from both individual and multiple IoT devices, increasing the scale and threat of replay attacks on smartphones. Software related vulnerabilities were also observed, specifically, the inability to disable on screen pairing notifications, which contributes to the success of UX deception. This research experimentally confirmed that BLE connections on mobile devices are vulnerable to replay and sniffing attacks using SDR tools such as HackRF one.

*Replay attacks and sniffing in Bluetooth low energy communications with ... (Juan Sebastian Orozco Duran)*

Various BLE exploitation techniques were evaluated in four defined scenarios: scenario 1 captured and replayed ADV\_NONCONN\_IND messages, generating fake pairing requests; attacks were most effective at 10–12 MHz bandwidths, while lower bandwidths resulted in failures due to partial packet capture. Scenario 2 successfully replicated fake pairing signals, tricking users into interacting with nonexistent devices; 39 successful attacks were recorded, although some devices (C, D, E, and F) showed greater resilience by allowing users to suppress pairing notifications. Scenario 3 successfully replicated pairing messages, including fake battery levels, tricking smartphones into believing they were connected to rogue devices, exposing UX integrity vulnerabilities without requiring OS level access. Scenario 4 demonstrated that passive sniffing can extract sensitive information such as MAC addresses, broadcast types, access addresses, transmit power, vendor ID, and SCAN\_RSP packets data that can be exploited for spoofing, device identification, or targeted jamming. These findings provide strong empirical evidence of the BLE protocol's structural weaknesses against RF based attacks. Even without access to cryptographic keys, attackers can disrupt communications, manipulate interfaces, and deceive users using low cost SDR tools.

This study also allows the identification of certain limitations. The testing environment exhibited uncontrolled electromagnetic interference, such as Wi-Fi signals or nearby BLE devices, and the SDR bandwidth was restricted to the 10-12 MHz range, which could have affected the overall success rate. Furthermore, the work focused exclusively on replay and sniffing attacks, excluding other BLE vulnerabilities such as active jamming or MITM attacks. These limitations open opportunities for future research aimed at expanding the scope of this study.

In conclusion, this research not only reveals critical vulnerabilities in BLE communications but also emphasizes the urgent need to strengthen pairing and authentication mechanisms in wireless systems, particularly in resource-constrained IoT environments. The findings obtained may contribute to the development of future, more secure iterations of the Bluetooth standard and serve as a methodological reference for cybersecurity studies in short-range wireless networks.

## FUNDING INFORMATION

This product is funded by the “Universidad Militar Nueva Granada-Vicerrectoría de Investigaciones”. Product derived from the MAXWELL research seedbed of the GISSIC research group. Universidad Militar Nueva Granada. Year 2025.

## AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Juan Sebastian Orozco Duran	✓		✓	✓	✓	✓	✓		✓	✓	✓			✓
Edith Paola Estupiñan Cuesta		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓
Juan Carlos Martínez Quintero		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓

C : **C**onceptualization

M : **M**ethodology

So : **S**oftware

Va : **V**alidation

Fo : **F**ormal analysis

I : **I**nvestigation

R : **R**esources

D : **D**ata Curation

O : Writing - **O**riginal Draft

E : Writing - Review & **E**ding

Vi : **V**isualization

Su : **S**upervision

P : **P**roject administration

Fu : **F**unding acquisition

## CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

## DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [JSOD], upon reasonable request.

## REFERENCES





- [1] M. Collotta, G. Pau, T. Talty, and O. K. Tonguz, "Bluetooth 5: A Concrete Step Forward toward the IoT," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 125–131, Jul. 2018, doi: 10.1109/MCOM.2018.1700053.
- [2] C. Liu, Y. Zhang, and H. Zhou, "A Comprehensive Study of Bluetooth Low Energy," in *2021 International Conference on Mechanical Automation and Electronic Information Engineering (MAEIE 2021)*, 2021, vol. 2093, pp. 1–9, doi: 10.1088/1742-6596/2093/1/012021.
- [3] M. V., V. R., and A. Shukla, "Bluetooth Vulnerabilities and Security," in *2025 International Conference on Emerging Smart Computing and Informatics (ESCI)*, Pune, India, 2025, pp. 1–6, doi: 10.1109/ESCI63694.2025.10987941.
- [4] J. R. Machado-Fernández, "Software Defined Radio: Basic Principles and Applications," *Revista Facultad de Ingeniería*, vol. 24, no. 38, pp. 79–96, Dec. 2015.
- [5] I. Natgunanathan, N. Fernando, S. W. Loke, and C. Weerasuriya, "Bluetooth Low Energy Mesh: Applications, considerations and current state of the art," *Sensors*, vol. 23, no. 4, pp. 1–25, 2023, doi: 10.3390/s23041826.
- [6] M. TajDini and V. Sokolov, "Penetration Tests for Bluetooth Low Energy and Zigbee using the Software-Defined Radio," *Сучасний Захист Інформації*, no. 1, pp. 82–89, 2018, doi: 10.5281/zenodo.2528810.
- [7] M. TajDini, V. Sokolov, and V. Buriachok, "Men-in-the-middle attack simulation on low energy wireless devices using software define radio," *CEUR Workshop Proceedings*, vol. 2386, pp. 287–296, 2019, doi: 10.2139/ssrn.3455453.
- [8] S. Bräuer, A. Zubow, S. Zehl, M. Roshandel, and S. Mashhadi-Sohi, "On practical selective jamming of Bluetooth Low Energy advertising," in *2016 IEEE Conference on Standards for Communications and Networking, CSCN 2016*, Oct. 2016, pp. 1–6, doi: 10.1109/CSCN.2016.7785169.
- [9] P. Staat, K. Jansen, C. Zenger, H. Elders-Boll, and C. Paar, "Analog Physical-Layer Relay Attacks with Application to Bluetooth and Phase-Based Ranging," in *WiSec 2022 - Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, May 2022, pp. 60–72, doi: 10.1145/3507657.3528536.
- [10] H. Givvehchian *et al.*, "Evaluating Physical-Layer BLE Location Tracking Attacks on Mobile Devices," in *Proceedings - IEEE Symposium on Security and Privacy*, May 2022, pp. 1690–1704, doi: 10.1109/SP46214.2022.9833758.
- [11] D. Z. Sun, Y. Mu, and W. Susilo, "Man-in-the-middle attacks on Secure Simple Pairing in Bluetooth standard V5.0 and its countermeasure," *Personal and Ubiquitous Computing*, vol. 22, no. 1, pp. 55–67, Feb. 2018, doi: 10.1007/s00779-017-1081-6.
- [12] T. Melamed, "An active man-in-The-middle attack on bluetooth smart devices," *International Journal of Safety and Security Engineering*, vol. 8, no. 2, pp. 200–211, Feb. 2018, doi: 10.2495/SAFE-V8-N2-200-211.
- [13] C. Zuo, Z. Lin, H. Wen, and Y. Zhang, "Automatic fingerprinting of vulnerable BLE IoT devices with static uuids from mobile apps," in *Proceedings of the ACM Conference on Computer and Communications Security*, Nov. 2019, pp. 1469–1483, doi: 10.1145/3319535.3354240.
- [14] K. M. Acharige, O. D. P. Albuquerque, M. Fantinato, S. M. Peres, and P. C. K. Hung, "A security study of Bluetooth-powered robot toy," *Journal of Surveillance, Security and Safety*, 2021, doi: 10.20517/jsss.2020.17.
- [15] J. Wang, F. Hu, Y. Zhou, Y. Liu, H. Zhang, and Z. Liu, "BlueDoor: Breaking the secure information flow via BLE vulnerability," in *MobiSys 2020 - Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, Jun. 2020, pp. 286–298, doi: 10.1145/3386901.3389025.
- [16] A. M. Lonzett, P. Cope, J. Campbell, B. J. Mohd, and T. Hayajneh, "Security vulnerabilities in bluetooth technology as used in IoT," *Journal of Sensor and Actuator Networks*, vol. 7, no. 3, pp. 1–26, Jul. 2018, doi: 10.3390/jsan7030028.
- [17] Bluetooth S.I.G., "Bluetooth Core Specification Addendum," *Bluetooth SIG*, vol. 1, no. 1, pp. 1–32, Jul. 2017.
- [18] J. Katsandres, "Bluetooth Low Energy -It Starts with Advertising | Bluetooth® Technology Website," *Bluetooth® Technology Website*, 2017. [Online]. Available: <https://www.bluetooth.com/blog/bluetooth-low-energy-it-starts-with-advertising/>. (Date accessed: Jun. 17, 2025)
- [19] M. A. Al-Shareeda, S. Manickam, S. A. Laghari, and A. Jaisan, "Replay-Attack Detection and Prevention Mechanism in Industry 4.0 Landscape for Secure SECS/GEM Communications," *Sustainability (Switzerland)*, vol. 14, no. 23, pp. 1–15, Nov. 2022, doi: 10.3390/su142315900.
- [20] W. Albazraq, J. Huang, and G. Xing, "A Practical Bluetooth Traffic Sniffing System: Design, Implementation, and Countermeasure," *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 71–84, Feb. 2019, doi: 10.1109/TNET.2018.2880970.
- [21] Y. Qu and P. Chan, "Assessing Vulnerabilities in Bluetooth Low Energy (BLE) Wireless Network Based IoT Systems," in *Proceedings - 2nd IEEE International Conference on Big Data Security on Cloud, IEEE BigDataSecurity 2016, 2nd IEEE International Conference on High Performance and Smart Computing, IEEE HPSC 2016 and IEEE International Conference on Intelligent Data and Security, IEEE IDS 2016*, Apr. 2016, pp. 42–48, doi: 10.1109/BigDataSecurity-HPSC-IDS.2016.63.
- [22] S. S. Hassan, S. Das Bibon, M. S. Hossain, and M. Atiquzzaman, "Security threats in Bluetooth technology," *Computers & Security*, vol. 74, pp. 308–322, May 2018, doi: 10.1016/j.cose.2017.03.008.
- [23] M. Almiani *et al.*, "Bluetooth application-layer packet-filtering for blueborne attack defending," in *2019 4th International Conference on Fog and Mobile Edge Computing, FMEC 2019*, Jun. 2019, pp. 142–148, doi: 10.1109/FMEC.2019.8795354.
- [24] M. Ai *et al.*, "Blacktooth: Breaking through the Defense of Bluetooth in Silence," in *Proceedings of the ACM Conference on Computer and Communications Security*, Nov. 2022, pp. 55–68, doi: 10.1145/3548606.3560668.
- [25] K. Lounis and M. Zulkernine, "Bluetooth Low Energy Makes 'Just Works' Not Work," in *2019 3rd Cyber Security in Networking Conference, CSNet 2019*, Oct. 2019, pp. 99–106, doi: 10.1109/CSNet47905.2019.9108931.
- [26] K. A. Hashim, H. H. Qasim, A. E. Hamzah, O. A. Hasan, and M. Al-Jadiri, "Door lock system based on internet of things and Bluetooth by using Raspberry Pi," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 5, pp. 2753–2762, Oct. 2023, doi: 10.11591/eei.v12i5.5134.
- [27] A. Muñoz, C. Fernández-Gago, and R. López-Villa, "A Test Environment for Wireless Hacking in Domestic IoT Scenarios," *Mobile Networks and Applications*, vol. 28, no. 4, pp. 1255–1264, Aug. 2023, doi: 10.1007/s11036-022-02046-x.
- [28] C. Caballero-Gil, R. Álvarez, C. Hernández-Goya, and J. Molina-Gil, "Research on smart-locks cybersecurity and vulnerabilities," *Wireless Networks*, vol. 30, no. 6, pp. 5905–5917, Aug. 2024, doi: 10.1007/s11276-023-03376-8.
- [29] K. Lounis and M. Zulkernine, "Connection Dumping Vulnerability Affecting Bluetooth Availability," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11391, pp. 188–204, doi: 10.1007/978-3-030-12143-3\_16.
- [30] S. Ditton, A. Tekeoglu, K. Bekiroglu, and S. Srinivasan, "A Proof of Concept Denial of Service Attack against Bluetooth IoT Devices," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2020*, Mar. 2020, pp. 1–6, doi: 10.1109/PerComWorkshops48775.2020.9156126.
- [31] M. Zubair, D. Unal, A. Al-Ali, and A. Shikfa, "Exploiting bluetooth vulnerabilities in e-health IoT devices," in *ACM*





- International Conference Proceeding Series*, Jul. 2019, pp. 1–7, doi: 10.1145/3341325.3342000.
- [32] A. C. T. Santos, J. L. S. Filho, Á. Í. S. Silva, V. Nigam, and I. E. Fonseca, “BLE injection-free attack: a novel attack on bluetooth low energy devices,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 5, pp. 5749–5759, May 2023, doi: 10.1007/s12652-019-01502-z.
- [33] T. Nagrare, P. Sindhwad, and F. Kazi, “BLE Protocol in IoT Devices and Smart Wearable Devices: Security and Privacy Threats,” *arXiv*, 2023, doi: 10.48550/arXiv.2301.03852.
- [34] A. Barua, M. A. Al Alamin, M. S. Hossain, and E. Hossain, “Security and Privacy Threats for Bluetooth Low Energy in IoT and Wearable Devices: A Comprehensive Survey,” *IEEE Open Journal of the Communications Society*, vol. 3, pp. 251–281, 2022, doi: 10.1109/OJCOMS.2022.3149732.
- [35] K. Lounis and M. Zulkernine, “Attacks and Defenses in Short-Range Wireless Technologies for IoT,” *IEEE Access*, vol. 8, pp. 88892–88932, 2020, doi: 10.1109/ACCESS.2020.2993553.
- [36] J. Veijalainen, D. Kozlov, and Y. Ali, “Security and Privacy Threats in IoT Architectures,” in *Proceedings of the 7th International Conference on Body Area Networks*, 2013, doi: 10.4108/icst.bodynets.2012.250550.
- [37] M. Ryan, *ice9-bluetooth-sniffer: Wireshark-compatible all-channel BLE sniffer for bladeRF, with wideband Bluetooth sniffing for HackRF and USRP*. GitHub, 2013, [Online]. Available: <https://github.com/mikeryan/ice9-bluetooth-sniffer>. (Date accessed: Jul. 28, 2025).

## BIOGRAPHIES OF AUTHORS







**Juan Sebastian Orozco Duran**     received the Telecommunications Engineer from Militar Nueva Granada University in 2024. He is currently in his final semester of a master's degree in integrated project management and is currently a research assistant at the same university. He can be contacted at email: [est.juan.orozco1@unimilitar.edu.co](mailto:est.juan.orozco1@unimilitar.edu.co).



**Edith Paola Estupiñan Cuesta**     received the M.Sc. degree in Electronic Engineering from Pontifical Xavierian University in 2013. She is currently professor at Militar Nueva Granada University. Her research interests include mobile network, traffic analysis and data, and management network. She can be contacted at email: [edith.estupinan@unimilitar.edu.co](mailto:edith.estupinan@unimilitar.edu.co).



**Juan Carlos Martínez Quintero**     received the M.Sc. degree in autonomous systems of production from in Universidad Tecnologica de Pereria in 2013. He is currently professor at Militar Nueva Granada University. His research interests include mobile networks, SDR, communication systems, and digital signal processing. He can be contacted at email: [juan.martinezq@unimilitar.edu.co](mailto:juan.martinezq@unimilitar.edu.co).