Indescribably Blue: Bluetooth Low Energy Threat Landscape

Christopher Skallak and Silvie Schmidt
Competence Centre for IT-Security, FH Campus Wien, Favoritenstrasse 226, Vienna, Austria

Keywords: Bluetooth Low Energy, IoT-Security, BLE.

Abstract: This paper elaborates security vulnerabilities of Bluetooth Low Energy. The STRIDE process is used to builld a threat model in order to identify these vulnerabilites. These range from packet sniffing on the physical layer to sophisticated Machine-in-the-Middle attacks that are built upon address spoofing and jamming attacks. The proposed threat model also identifies the optional and mandatory dependencies between the attack vectors. Furthermore, we elaborate the attack vectors aligned to the BLE stack.

1 INTRODUCTION

Bluetooth Low Energy (BLE) upon many other numerous communication protocols, e.g., ZigBee, Internet Protocol Version 6 (IPv6) over Low Power Wireless Personal Area Networks (6LoWPAN), and Long Range Wide Area Network (LoRaWAN), is used in the realm of Internet of Things (IoT) devices for low-power wireless communication. These devices made their way into various domains like the industry with the Industrial Internet of Things (IIoT), healthcare with smartwatches, and transportation (Sarawi et al., 2017). BLE devices do not always use the provided security features. This was demonstrated by Richo Healey and Mike Ryan at the DefCon 23[1] by exploiting the BLE connection of electric skateboards (Healey and Ryan, 2020). They did reverse engineer the BLE connection between a Boosted Board[2] and its remote. Therefore, they were able to control the board with their own script and to take over a connection by jamming it and connecting to the adversary device to the board. Anthony Rose and Ben Ramsey demonstrated this at DefCon 24[3] by exploiting various smart locks controlled with BLE. Their findings show that the Noke Padlock[4] , Masterlock Padlock[5], August Doorlock[6], and Kwikset Kevo Doorlock[7] resisted their penetration testing by using proper AES encryption with truly random nonces, multi-factor authentication, and allowing long passwords with a length of up to 16 and 20 characters. The Quicklock Doorlock[8] & Padlock and iBluLock Padlock[9] did not apply any encryption and sent the password in plain text. Other locks were susceptible to replay attacks like the Elecycle[10] EL797 & EL797G Smart Padlocks, Vians Bluetooth Smart Doorlock[11], and Lagute Sciener Smart Doorlock[12] (Rose and Ramsey, 2020). Consequently, these and various other attacks on the BLE connections put valuable data and BLE devices at risk. Therefore, this paper elaborates various BLE vulnerabilities and creates a threat model using the STRIDE approach. In our outlook we present a proposal for a design of an open-source BLE development and pentesting tool called BLEBerry. This work is based on (Skallak, 2023).

(Barua et al., 2022) offer a comprehensive survery on BLE threats; our work aims to focus on the dependencies between the threats and attack vectors in order to be able to design a user-friendly BLE pentesting tool.

[1]https://defcon.org/html/defcon-23/dc-23-index.html, accessed 2023-06-24

[2]https://boostedusa.com/, accessed 2023-06-24

[3]https://defcon.org/html/defcon-24/dc-24-index.html, accessed 2023-06-24

[4]https://www.janusintl.com/products/noke, accessed 2023-06-24

[5]https://www.masterlock.com/products/bluetooth-electronic-locks, accessed 2023-07-04

[6]https://august.com/, accessed 2023-07-04

[7]https://www.kwikset.com, accessed 2023-07-04

[8]https://www.qicklock.com/, accessed 2023-07-04

[9]https://impowerelitegroup.wixsite.com/iblulock, accessed 2023-07-04

[10]https://www.elecycles.com/bluetooth-smart-lock-797.html, accessed 2023-07-04

[11]https://www.vianslock.com/, accessed 2023-07-04

[12]https://www.sciener.com/, accessed 2023-07-04

Skallak, C. and Schmidt, S.
Indescribably Blue: Bluetooth Low Energy Threat Landscape.
DOI: 10.5220/0012737300003705

In *Proceedings of the 9th International Conference on Internet of Things, Big Data and Security (IoTBDS 2024)*, pages 339-350
ISBN: 978-989-758-699-6; ISSN: 2184-4976

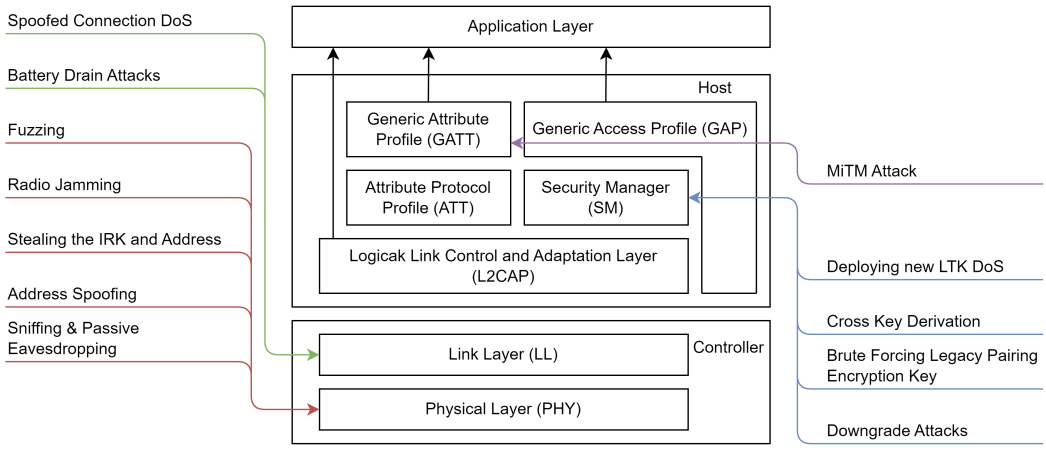

Figure 1: Bluetooth Low Energy Threat Model (Skallak, 2023).

2 THREAT MODEL

Our BLE threat model is based on the STRIDE method; it classifies the threat vectors for BLE. STRIDE is a mnemonic for the following types of attack (Shostack, 2014):

- **S**poofing attacks are used by attackers to disguise themselves as their target. Spoofing attacks on BLE are achieved by altering the Bluetooth device address in the header of the link layer PDUs. A successful spoofing attack can establish spoofed connections, Machine-in-the-Middle attacks (MitM), or send advertisements with counterfeit data disguised as a legitimate device.
- **T**ampering is the act of modifying data. In a BLE attack scenario, this is especially used to modify pairing packets to enable MitM attacks or to alter transmitted data during a MitM attack.
- **R**epudiation describes the denial of malicious actions. A digital cryptographically secure data signing approach or message logging mechanism is applied to prevent repudiation. These measures are not applicable for BLE because the protocol is developed for devices with restricted power and computing resources.
- **I**nformation Disclosure describes the leakage of data without gaining the needed authorization. The BLE standard implements an access control scheme where each attribute holds its access requirements. This is implemented as the so-called properties that indicate that the content of the attribute can be unencrypted, authenticated, encrypted, or authenticated and encrypted. Therefore the developer can decide which attributes are openly accessible to everybody and which ones are sensitive. However, downgrade attacks removed encryption and led to information disclosure.
- **D**enial-of-Service (DoS) attacks prevent the target device from providing its services or preventing a legitimate user from accessing them. BLE is prone to DoS attacks, especially via radio jamming, because all packets are sent over the air.
- **E**levation of Privilege attacks allow the adversary to raise their privilege and interact with services they are not supposed to. Attacks on BLE only use the elevation of privilege by spoofing the device address to bypass the address whitelisting feature.

The BLE threats are categorized in Table 1. Consequently, the attack vecotrs are identified at: Physical Layer, the Link Layer, the Security Manager Protocol, and the Generic Attribute Protocol. They are discussed in Section 3.

Attack Model Limitations

The threats addressed in our threat model only focus on the BLE protocol specification and do not cover the application layer. The implementation of the application layer is outsourced to the manufacturer of the product, which renders it out of scope for this paper. Furthermore, the vulnerabilities of the Bluetooth BR/EDR are not addressed since the two specifications are incompatible except for Key Negotiation of Bluetooth (KNOB) (Antonioli et al., 2019b) Attack and Cross Transport Key Derivation (CTKD) (SIG, 2021). KNOB is a Bluetooth BR/EDR vulnerability adapted to work with the BLE protocol. CTKD derives a BLE key from a Bluetooth BR/EDR key, and vice versa (SIG, 2021). Therefore, a compromized Bluetooth BR/EDR key can be used to create a mali-

Table 1: STRIDE categorization of threats (Skallak, 2023).

Layer	Attack Vector	S	T	R	I	D	E
PHY	Physical Sniffing			S	M		
PHY	Spoofing [S]	M	S	S	I		I
PHY	[S] Advertisement spoofing	M	S	S			
PHY	[S] GATT peripheral spoofing	M	S	S	S		
PHY	[S] GATT central spoofing	M	S	S			I
PHY	Stealing the BD_Address and IRK	M	S	S	S		
PHY	Radio Jamming			I		M	
PHY	Fuzzing				I	I	
LL	Battery drain Attacks [BDA]					M	
LL	[BDA] Connection/Pairing request flooding					M	
LL	[BDA] Battery drain Attacks		I			M	
LL	[BDA] Spoofed connection					M	
SMP	Downgrade Attacks [DA]	S	M	S	I		
SMP	[DA] Pairing Downgrade Attack	S	M	S			
SMP	[DA] Downgrade attack to Just Works	S	M	S			
SMP	[DA] Encryption Key entropy downgrade attack	S	M	S			
SMP	[DA] Downgrade Attack to plain text	S	M	S	S		
SMP	Brute Forcing Legacy Pairing Encryption Key			S	M		S
SMP	Cross Key Derivation (CTKD)	S	M	S		S	S
SMP	Deploying new LTK DoS				M		
GATT	MITM Attack	S	M	S	I		I

M	Main STRIDE category of threat
S	Substitute STRIDE category of threat
I	STRIDE category applies to some specific threat implementations

cious BLE connection. The attacker has no physical access to the victim's device. Therefore, altering the firmware in this attack scenario is not possible. Hardware security is out of scope of this paper.

The threat model is based on the BLE stack with its protocols and layers, as illustrated in Figure 1. The proposed threats and vulnerabilities are located at the lowest entry point they take place. If a threat has dependencies on another one they are positioned on the layer they are performed on not where the prequisite threats are applied. All threats that are mentioned in this Section are discussed in Section 3.

Figure 2 illustrates the dependencies of all threats with each other. The Figure differentiates between mandatory and optimal dependencies. For instance, a Machine-in-the-Middle (MitM) Attack requires address spoofing and radio jamming can be used to raise the chances of a successful spoofed connection but is not mandatory. The theft of the Identity Resolving Key (IRK) results in a special issue where a mandatory dependency loop with address spoofing is created. Impersonating the target Peripheral is required to gather the IRK of the target Central. The IRK is then used to impersonate the target Central and establish a connection with the target Peripheral.

3 ATTACK VECTORS

This Chapter discusses and examines the attack vectors identified in Table 1.

3.1 Physical Layer

The physical layer is the medium the communication protocol uses to communicate over. BLE uses radio waves, especially the 2.4 GHz ISM band, for communication between devices. The radio is a shared medium that all devices can access with an antenna and chip that operates on the frequency range between 2402 and 2480 MHz. Therefore, this layer is vulnerable to jamming attacks and devices that sniff the communication over the radio.

3.1.1 Packet Sniffing and Passive Eaves Dropping

Packet Sniffing collects packets within a network and converts them into a human-readable format for further analysis. Network administrators use traditional Ethernet network sniffing for monitoring traffic, network troubleshooting, and examining proto-

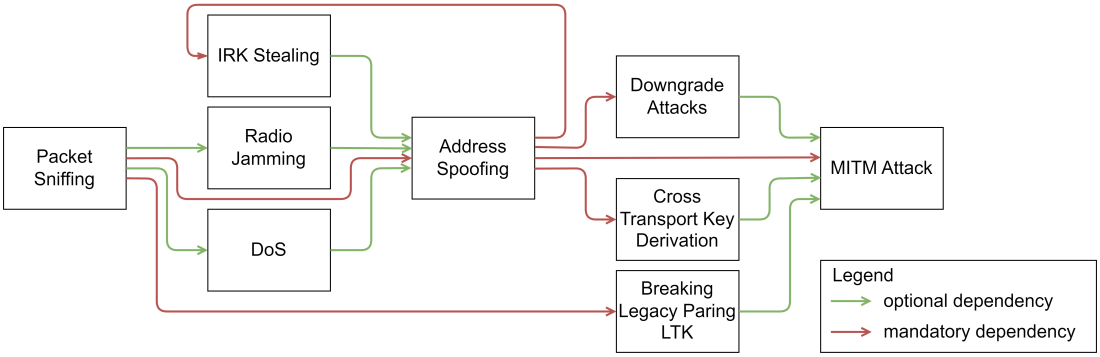

Figure 2: Bluetooth Low Energy Threat Dependencies (Skallak, 2023).

cols, which might send sensitive data in an insecure fashion (Thiyeb et al., 2018). Sniffing BLE Packets is performed by either monitoring the local Bluetooth device or host controller interface with tools, like Wireshark[13], which only allows seeing broadcasted advertisements in the scanning state and data transfers between the local and connected device in the connected state. Specialized hardware solutions, called BLE sniffers, collect messages on the physical layer. That adds the capability to sniff connections between other devices. BLE sniffers can be acquired by different vendors and various price ranges. High-end sniffers which are used in a commercial setting are costly, e.g., Spanalytics PANalyzr[14]. This sniffer can sniff the whole 2.4-2.5 GHz ISM band at the same time and analyze BLE, Bluetooth BR/EDR, and Wi-Fi Packets. There are also low-cost consumer friendly counterparts like:

- Ubertooth One by Great Scott gadgets[15], based on the CC2400 wireless transceiver. Besides packet sniffing, the Ubertooth One can be used for spectrum analysis of the 2.4 GHz band.
- nRF Sniffer[16] by Nordic Semiconductors operated with nordic development kits like the nRF52840 Dongle or the Bluefruit LE Sniffer by Adafruit[17], which sniff connections up to BLE version 4.1.

Low-cost BLE sniffers have the trade-off that they can sniff only one of the 40 BLE channels and can follow only one connection.

The Ettus OctoClock[18] is added to synchronize the two SDRs. In the performance evaluation, they sniffed 24 connections simultaneously, and the system detected all active connections in less than 200 ms. However, the proposed device requires expensive hardware and therefore is not accessible to hobby tinkerers (Cominelli et al., 2020) Sniffing BLE connections cannot be mitigated because the data is sent over the air via radio waves. However, it is possible to prevent the transmitted data from information disclosure by encrypting it. The encryption can be either applied by the security features of BLE by using encrypted writeable or readable characteristics of a GATT server or proprietary implementations which encrypt the transmitted data on the application layer. If the BLE security features are used, the data is encrypted on the link layer, which ensures that all the data is encrypted and only the header of the packet, access address, and CRC are sent in plain text.

3.1.2 [DoS] Radio Jamming

BLE is vulnerable to RF jamming like any wireless communication protocol, e.g., WiFi. With BLE, the three advertisement channels are high-interest targets of jamming attacks to help with a spoofing attack or to disturb Broadcaster and Observer data transmission. Jamming aims to create planned data collisions to flip bits on the receiver end. The flipped bit or bits will lead to a Cyclic Redundancy Checksum (CRC) failure, and the receiver will discard the message before processing. BLE jamming can be distinguished into full band or selective channel jamming. Full band jamming will flood all 40 channels or the three advertisement channels with arbitrary radio signals. This type of jamming has high power consumption and can be easily detected. Selective channel jamming attacks only one channel and is more power ef-

[13]https://www.wireshark.org/, accessed 2023-10-07

[14]https://spanalytics.com/product/panalyzr/, accessed 2023-10-07

[15]https://greatscottgadgets.com/ubertoothone/, accessed 2023-10-07

[16]https://www.nordicsemi.com/Products/Development-hardware/nRF52840-Dongle, accessed 2023-11-08

[17]https://www.adafruit.com/product/2269, accessed 2023-11-08

[18]https://www.ettus.com/all-products/octoclock/, accessed 2023-11-08

ficient and less detectable (Bräuer et al., 2016); the authors proposed a selective reactive jammer. A selective reactive jammer waits for a packet to be sent by the target and then reacts by sending an arbitrary bitstream, also called a jamming signal, on the same channel. The jammer is based on the RedBearLab BLE Nano[19] board, which is equipped with a Nordic nRF51822 System on a chip. This device was chosen because it has a short turnaround time of 140µs from receiving to transmitting, which is required for reactive jamming. It also has the advantage of a small footprint and power efficiency at a small price. The proposed device is separated into a beacon detection and jamming part. Beacon detection receives and decodes packets sent on its current advertising channel, and if the Bluetooth address of the received packet matches the target, it switches to the jamming phase. This phase emits a short jamming signal and hops to the next advertisement channel, and switches back to the detection phase. The channel must be switched because advertising devices use a deterministic channel switching scheme containing channels 37, 38, and 39 by default. Devices can also be configured only to use a subset of these channels, which must be found out in a reconnaissance phase, and the jammer must be configured accordingly.

3.2 Link Layer

The link layer is responsible for packet addressing, obfuscating the real device address with resolvable private addresses, and filtering the connection requests by a whitelist. Attack vectors on this layer range from spoofing addresses, stealing the key to resolve private addresses to Denial of Service attacks.

3.2.1 Address Spoofing

BLE Address spoofing is the act of altering the BLE address to imitate another device. The most straightforward way to spoof the address is if the Bluetooth chip allows changing its address with manufacturer-specific Host Controller Interface (HCI) commands. The GitHub repository of the Linux Bluetooth stack Bluez[20] provides a tool called bdaddr[21] that performs this process for controllers manufactured by Texas Instruments[22], Ericsson[23], Broadcom[24], and ST Microelectronics[25]. Address spoofing lays the foundation for many other attacks, e.g., Machine-in-the-Middle Attacks, Denial of Service, and bypassing the Whitelist. In this paper, BLE Spoofing is separated into three categories: advertisement spoofing, GATT Peripheral spoofing, and GATT Central spoofing.

Advertisement Spoofing
Advertisements are used to indicate that connection establishment with the sending device is possible or for broadcasting public messages. Broadcasted messages usually contain data about the device's services, e.g., its name or sensor data. This data is presented in a format defined by the BLE specification to provide interoperability. Although, some manufacturers implement a proprietary format, like the one used for iBeacon by Apple. Broadcasting data is used for connectionless data transfers between Broadcasters and Observers, e.g., sensor nodes that broadcast their measured data to Observers, which store and visualize it for the user. An attacker can broadcast arbitrary data to the Observers by spoofing the Bluetooth address. The address is spoofed by changing the advertisement packet's source address field to the impersonated Broadcaster device's address. The spoofed messages are used to feed false information to the Observers (Jasek, 2016). Advertisement spoofing is also used to trick Central devices to connect to a spoofed Peripheral. The usual procedure for a Central to connect with a Peripheral starts by listening for advertisements. As soon as the Central receives an advertisement with the address of a desired Peripheral, it sends a connection request. BLE Peripherals are often designed to have low power consumption for extended life capabilities on a small battery. Therefore they often have configured long advertising intervals. An adversary can exploit this by sending advertisements with a spoofed address at a shorter advertising interval. Due to the shorter interval, the adversary device has higher odds to be chosen for connection establishment by the Central. An attacker can achieve even greater odds by either jamming the other device or by connecting to the legitimate Peripheral. Many Peripherals are configured to allow only one connection at a time and therefore stop advertising when a device establishes a connection to it, which can be abused for this attack as well. As soon as the target connects to the adversary device it presents a spoofed GATT profile to the target and responds with arbitrary data if the target wants to read an attribute. The spoofed

[19]https://github.com/RedBearLab/BLENano, accessed 2024-01-08
[20]https://github.com/bluez/bluez, accessed 2024-01-08
[21]https://github.com/bluez/bluez/blob/master/tools/bdaddr.rst, accessed 2024-01-08
[22]https://www.ti.com/, accessed 2024-01-08
[23]https://www.ericsson.com/, accessed 2024-01-08
[24]https://www.broadcom.com/, accessed 2024-01-08
[25]https://www.st.com/, accessed 2024-01-08

GATT profile can be acquired by connecting to the legitimate Peripheral and scanning it beforehand (Jasek, 2016).

GATT Peripheral Spoofing
(Wu et al., 2020) focus on the GATT Peripheral spoofing attack of a beforehand securely paired connection between the target Central and a legitimate Peripheral with their proposed BLE Spoofing Attacks (BLESA). The spoofing attack is performed when the Central attempts to reconnect due to signal loss. The adversary sends spoofed advertisements and hopes the target Central establishes a connection. The adversary then presents the target with a cloned GATT profile with services that do not use encryption and collects the received plain text data. This attack is based on the following two BLE design weaknesses (Wu et al., 2020):

- The encryption and authentication of BLE energy are handled per attribute and are optional due to the specification.
- The Peripheral enforces the security policies, and the Central cannot request encryption or authentication.

A reconnecting Central can follow two different authentication procedures reactive authentication and proactive authentication. Whereas the reactive authentication procedure is always susceptible to BLESA. The proactive authentication procedure is vulnerable but the authors of the paper found that some implementations of the BLE stack are vulnerable because they are not following the specification correctly (Wu et al., 2020).

- Reactive authentication procedure: The Central assumes no security and requests the desired attribute. The Peripheral then indicates to the Central that the used security level is insufficient and requests to raise it. The message transfers can be repeated after adjusting the security level. When the BLESA attack is performed on this authentication procedure, the adversary does not send the insufficient security level error message, and the Central does not adjust the security level. The Central writes data to a sensitive characteristic, e.g., password in plain text, which leads to information disclosure (Wu et al., 2020).
- Proactive authentication procedure: With this procedure the Central sends an enable encryption request before interacting with any attribute. If the Peripheral still has the shared Long Term Key (LTK) the secure data transfer can start. Otherwise, the Peripheral responds with an encryption failed message, and both devices abort the connection and perform a new pairing procedure. The BLESA attack on this procedure responds with a missing LTK error to the encryption request. The Central is supposed to close the connection or initiate another pairing process if the specification was followed correctly. Although, some implementations of the BLE specification do not abort the connection and fall back to a plain text communication and are therefore vulnerable (Wu et al., 2020).

(Wu et al., 2020) recommend the usage of the proactive authentication procedure and fixing the implementation issues to mitigate this attack. In their testing, they discovered that the Linux BLE stack BlueZ[26] uses the reactive procedure and is not vulnerable to BLESA. Android and iOS devices use the proactive procedure but were still vulnerable at the time of their research due to implementation issues (Wu et al., 2020).

GATT Central Spoofing
GATT Central spoofing is required in these specific scenarios:

- The Machine-in-the-Middle attack, where a malicious third party impersonates both Central and Peripheral devices to locate itself in the middle of the connection to eavesdrop on transmitted messages. Machine-in-the-Middle attacks are discussed in further detail in Section 3.4.1 (Wang et al., 2020).
- If the Peripheral uses the whitelist security feature. This feature restricts devices that do not own a whitelisted address from establishing a connection to the Peripheral. Therefore spoofing is used to bypass the whitelist. If the spoofed device also uses the private address feature, the adversary additionally needs to steal the Identity Resolution Key (IRK) first. The process of obtaining the Bluetooth address and IRK is addressed in Section 3.2.2 (Zhang et al., 2020).
- If the target Peripheral has proprietary whitelisting, other security features or services implemented on the application layer, which performs operations based on the device address.

3.2.2 Stealing the IRK and Address

The Identity Resolution Key (IRK) is used to resolve the private address of the device. This private address is a privacy feature to hide the real address and prevent malicious actors from tracking the device by changing it periodically. The IRK is distributed with

[26]https://github.com/bluez/bluez, accessed 2024-01-08

the connection partner device during the pairing procedure to enable it to resolve the address in subsequent sessions. Each holder of the IRK can resolve the address. A malicious actor can abuse this distribution by creating a fake Peripheral that spoofs a legitimate Peripheral the target wants to connect with. Upon connection, the Central expects the adversary device to hold the Long Term Key, which it does not possess. Therefore, the adversary needs to trick the target into initiating a new pairing procedure to replace the Long Term Key of the target. This is accomplished by sending a 'key missing error message' during the connection process, which leads to a fallback to plain text communication. If the target intends to read or write an attribute that has encrypted read or write permissions, the fake device responds with an 'insufficient encryption error message', and the target initiates the re-paring process. During pairing, the target shares its real device address and the IRK with the attacker. With this knowledge, the attacker can impersonate the target device and bypass whitelisting of the legitimate Peripheral (Zhang et al., 2020).

3.2.3 DoS: Battery Drain Attacks

BLE was developed to provide a communication standard for battery-powered devices with a small energy footprint. The devices can be programmed to offer long sleep phases and short awake phases, which can be used to report sensor data. This allows sensor nodes to run for years on one battery and to be placed in hardly accessible areas. Denial of service by draining their battery puts sensor network nodes and the network itself at great risk or can even disrupt the network for the duration it takes to replace the batteries. High battery drain can be achieved by various methods as follows (Uher et al., 2016).

Connection or Pairing Request Flooding
This attack is performed by sending high quantities of pairing requests or connection requests and not performing the pairing or connection process (Sevier and Tekeoglu, 2019).

Denial of Sleep Attacks
This attack is performed by establishing a connection to the device and keeping the connection The BLE specification provides a link layer request packet that can be used for connection-based battery drain attacks. The LL_POWER_CONTROL_REQ can be used to request a transmission power increase, leading to a higher battery drain. The maximum power level can be requested by using the value of 0x7F in the packet. Although the device can deny the request (SIG, 2021).

3.2.4 DoS: Spoofed Connection

A simple Denial of Service against Centrals and Peripherals can be performed by establishing a usual connection. However, it is required to spoof a device and to advertise the cloned GATT profile to trick the Central into connecting to it. The Centrals then regards it as a legitimate connection and stops scanning advertisements. The spoofed device can drop the write requests and respond to read requests with arbitrary data. This prevents the device from using the features of the legit device (Jasek, 2016). The related attack on Peripherals can be performed because not all Peripherals allow multiple connections at a time. These Peripherals stop advertising as soon as the adversary is connected. This prevents the legitimate user from receiving its advertisements and establishing a connection (Lounis and Zulkernine, 2019). Peripherals were designed to allow only one connection to a Central simultaneously, but since BLE version 4.1, this restriction was lifted, and Peripherals allow multiple Centrals to establish a connection. Although this is an optional feature, the usage depends on the device's manufacturer. Therefore, many Peripherals are still vulnerable to this Denial of Service attack (SIG, 2021).

3.2.5 Fuzzing

Fuzzing is usually used to find vulnerabilities in BLE implementations and Bluetooth chips. A fuzzer sends arbitrary packets by sending unsupported values, invalid key sizes, or packets with mutated field lengths. BLE fuzzing can be differentiated into packet fuzzing and attribute fuzzing (Ray et al., 2018), (Garbelini et al., 2020).

Packet Fuzzing allows to interacting with all protocols of BLE. The packet can carry packet fields with mutated lengths or values to trigger non-compliant behavior. Fuzzing packets can result in the following vulnerabilities (Garbelini et al., 2020):

- Device Crashes can be achieved by corrupting the memory of a remote device with a buffer overflow or due to incorrect behavior of protocol execution.
- Device Deadlocks are the result of a hard fault that was not handled properly by restarting the device.
- Bypassing Security was possible by fuzzing Telink Semiconductor devices which led to the installation of a Long Term Key (LTK) of zero length (CVE-2019-19194). The Diffie-Hellman check of secure connection pairing was skipped on Texas Instruments semiconductors (CVE-2020-13593) with a fuzzing attack.

Attribute Fuzzing focuses on fuzzing the GATT service by writing arbitrary values to a characteristic.

3.3 Security Manager

The Security Manager Protocol is responsible for device authentication and the creation of key material. The attacks on this protocol focus on downgrading the used security features of the connection or compromizing keys.

3.3.1 Downgrade Attacks

Downgrade attacks are used to either lower the security to a level that is vulnerable to Machine-in-the-Middle (MitM) and other attacks, e.g., using the Just Works association model, to reduce the entropy of encryption keys or to remove the encryption altogether. Downgrade Attacks are used for spoofing and MitM attacks to enable the attacker to read the sent packets or alter them and inject arbitrary ones.

BLE Pairing Downgrade Attack
BLE has two possible pairing protocols to establish encryption for a connection. The first one was released with BLE and is called legacy pairing. The other one is called secure connections pairing and was introduced with BLE version 4.2 as the replacement for the old one because legacy pairing has a design flaw, as shown in Section 3.3.2. Both of the methods still coexist for backward compatibility. An attacker can use this to downgrade the pairing method to legacy pairing and brute force the keys which are used to encrypt the connections. This attack requires the attacker to perform a Machine-in-the-Middle (MitM) attack while the devices perform the pairing procedure. The malicious device alters the pairing request and response packet to force the devices to use legacy pairing with the passkey association model. Both pairing packets have the same fields and are used to decide the used parameters of the pairing procedure. The most important packet fields for this attack are the ”IO Capability” and ”AuthReq” fields. The IO capability field is set in a way that at least one device contains the value for display only (0x00) and the other device for keyboard only (0x02) value or both devices have the value for keyboard and display (0x04) to force the Passkey Entry pairing method. The ”AuthReq” field holds different flags, of which the secure connection (SC) is the only important one for the attack. The secure connection field indicates if the device can perform secure connection pairing. The secure connection pairing procedure is only used if both devices set the value to 1. Therefore, changing this value to 0 in one of the two packets is required, resulting in a downgrade to legacy pairing. This attack can be driven further by brute-forcing the key as described in Section 3.3.2. This attack can be mitigated by using the secure connection only mode of BLE, which only allows devices for pairing in secure connections mode with the trade-off of losing backward compatibility (SIG, 2021).

Association Model Downgrade Attack to Just Works
This attack requires the attacker to perform a Machine-in-the-Middle (MitM) attack before the pairing process is initiated to alter the pairing request and response packets. With this attack, the adversary party only changes the IO capabilities. At least one of the two packets needs to set the IO capability field to the value (0x03), which indicates that the device has neither a keyboard to insert numbers nor a display to show them. Due to the specification, this results in the usage of the Just Works association model, which Legacy and Secure Connections pairing implement. The Secure Connections variant provides higher security because it replaced the six-digit Temporal Key used by Legacy pairing with a high entropy Diffie-Hellman Key. Secure Connections pairing is secured against passive eavesdropping but is vulnerable to active MitM attacks. On the other hand, the Just Works model of Legacy pairing is vulnerable to both because a fixed value is used for the Temporal Key. This downgrade attack can be mitigated by using the secure connection only mode, which requires an authenticated association model which is only satisfied with Numeric Comparison, Out of Band and Passkey Entry (SIG, 2021).

Encryption Key Entropy Downgrade Attack with Key Negotiation of Bluetooth (KNOB)
The original Key Negotiation of Bluetooth (KNOB) exploit (Antonioli et al., 2019a) was developed to exploit Bluetooth BR/EDR’s Link Manager Protocol (LMP). The attack reduces the key’s entropy for link-layer encryption to one byte. The exploit was modified to also work for the BLE Protocol. BLE uses a different architecture, which moved the key size negotiation to the pairing process with a key size between 7 and 16 bytes. Therefore, BLE’s Long Term Key (LTK) has higher minimum entropy than Bluetooth BR/EDR’s LTK. The key length negotiation takes place in the first phase of pairing, named feature exchange, performed before encryption, and the messages are sent in clear text. An attacker can perform a Machine-in-the-Middle (MitM) attack, as mentioned in section 3.4.1, and alter the key size parameter of the feature exchange message to seven.

The in phase three established LTK is downgraded to an entropy of 7 bytes. All other keys, e.g., the link key, are derived from the LTK, which consequently result with a lower entropy than possible. The AES-CCM specification requires an encryption key with a size of 16 bytes. Therefore, shorter LTKs are padded in at the most significant bytes with zeros to fit this requirement. Once the pairing is completed, the connection is encrypted, restricting the attacker to only eavesdropping and logging the encrypted messages. The attacker then can brute force the LTK and decrypt the sent packets afterward. A brute-forced LTK also allows the attacker to actively eavesdrop on all subsequent connections and perform a MitM attack on them or even establish a spoofed connection with encryption. The tests conducted by Daniele Antonioli et al. (Antonioli et al., 2020a) show that all their tested devices are susceptible to the attack. Even the most secure BLE mode, which is specified to use secure connections with authentication and requires an LTK with an entropy of 16 bytes, can be downgraded to 7 bytes due to their findings (Antonioli et al., 2020a).

Downgrade Attack to Plain Text
This attack is similar to the attack model as described in Section 3.2.1. The attacker spoofs a Peripheral that the Central has paired with in the past. If the Central requests to enable encryption, the adversary responds with an encryption failed message to indicate a missing Long Term Key. The Central is supposed to close the connection or initiate the pairing procedure again if it follows the steps of the BLE specification correctly. Although, some devices perform a fallback to plain text communication due to implementation issues (Wu et al., 2020). Yue Zhang et al.[ZWD+20] used this implementation bug which was existent in Android 9.0 and some other design flaws of androids BLE implementation to create spoofing, passive eavesdropping, and MitM attacks where the communication between the target Central and spoofed Peripheral is downgraded to plain text (Zhang et al., 2020).

3.3.2 Brute Forcing Legacy Pairing Encryption Key

(Kwon et al., 2016) found a major design flaw in the legacy pairing Passkey Entry protocol. The Short Term Key (STK), used to encrypt the communication, is calculated by the Temporal Key (TK) and two random values, each generated by one device of the two devices. The random values are exchanged in plain text during the protocol. Therefore, the only secret parameter is the Temporal Key, a random six-digit value chosen by the user and inserted into both devices. The problem is that the six-digit TK only creates an entropy of around 20 bits, which can be brute-forced. This allows the attacker to calculate the STK and decrypt the exchanged messages. The LTK and additional key material are subsequently exchanged to the STK generation while being encrypted by the STK. Therefore the LTK and STK can be recovered by brute-forcing the passkey if the required messages were recorded. Mike Ryan (Ryan, 2013) has created a tool called crackle[27] to crack the TK and to recover the other keys from a pcap file of the communication establishment which can be recorded with the Ubertooth One (Kwon et al., 2016), (Ryan, 2013). BLE legacy pairing is still part of the specification, even with the introduction of the successor Secure Connections for backward capability and because of the existence of devices that still use BLE version 4.0. An adversary can abuse this to perform a downgrade attack to legacy paring, as shown in Section 3.3.1.

3.3.3 Cross Transport Key Derivation (CTKD)

This threat vector originates in Bluetooth BR/EDR and is used to create compromized LTK for BLE connections using the Cross-Transport Key Derivation (CTKD) protocol. This protocol was introduced in version 4.2 and is used by dual-mode devices to derive a Bluetooth BR/EDR Long Term Key (LTK) from a BLE LTK and vice versa. Before CTKD was implemented, both protocols needed to perform pairing to generate keys, and now with CTKD only one pairing process needs to be performed. CTKD can only be used with devices with a dual controller that supports BLE and Bluetooth BR/EDR, like most smartphones and laptops do. Daniele Antonioli et al. (Antonioli et al., 2020b) used this feature to create the so-called Blur attacks because they blur the boundaries between the two Bluetooth protocols. These attacks can be used for impersonation, Machine-in-the-Middle (MitM) attacks, and to overwrite a trusted LTK (Antonioli et al., 2020b).

CTKD Exploit
The blur attacks utilize the following four design issues of the CTKD feature to overtake a connection by overwriting the shared LTK between the two target identities (Antonioli et al., 2020b):

- **Dual Pairing**
  Both BT BR/EDR and BLE of dual-mode devices are in pair-able states when one of the both is currently in use. They accept pairing requests, al-

[27] http://lacklustre.net/projects/crackle/, accessed 2024-01-08

though they are not discoverable. Because the device is pair-able over both BT BR/EDR and BLE and only one of the both is currently used, an attacker can silently pair with the unused communication protocol using the Just Works pairing method.

- **Asymetric Role Systems**
  BT BR/EDR and BLE use different role systems. BLE implements the fixed roles Central and Peripheral, and BT BR/EDR differentiates between master and slave, but BT BR/EDR roles can be switched anytime, even before the pairing process.
- **Replacing Keys**
  CTKD can overwrite the existing key if the new one has the same or higher strength and protection against Machine-in-the-Middle (MitM). The Identity Resolving Key (IRK), which is used for hiding the real device address, is also sent when a BLE key gets derived from a BR/EDR key.
- **Manipulation of the Association Model**
  Devices forget the used association model after the pairing process. Therefore, a malicious device can connect to the unused protocol with a different association model, e.g., Just Works, and perform CTKD to replace the keys.

This exploit can overtake a secure connection of two paired devices if at least one is a dual-mode device. The attacker connects to one of the two devices by using the not used communication protocol while spoofing the address with the address of the other target. The attacker also claims to have neither input nor output capabilities to force the Just Works association model. If the attackers request to perform the CTKD procedure, the key for the legit connection is replaced. The attacker has overtaken the connection, which means the malicious party is connected to one target, and the legitimate device can not reconnect anymore due to the key miss-match. The attack can be leveled up to a MitM attack by performing the attack with the other device if it also is a dual-mode device (Antonioli et al., 2020b).

3.3.4 Deploying New LTK DoS

Deploying a new malicious LTK can cause a specific Denial of Service on Android 9.0 devices. This DoS exists because the Android applications can not detect a broken link key and do not initiate a new pairing phase to create a new one. Furthermore, the function removeBound() to delete a deprecated key can only be performed with system-level permissions, which means that they can only be deleted manually by removing the bound in the Android system settings. Although, replacing the Long Term Key requires the attacker to connect to the target Peripheral while spoofing the address of the target Central. During the connection setup, the adversary indicates a missing key with a ”key missing error message”. This causes the attacker and target device to do a pairing process and replaces the LTK, which is shared between the target Central and Peripheral. The cross-transport key derivation described in Sections 3.3.3 can be used to replace the key as well (Zhang et al., 2020).

3.4 Generic Attribute Protocol

The attack vector at the Generic Attribute Protocol is the Machine-in-the-Middle (MitM), which requires the malicious actor to spoof the Central and Peripheral to connect to the two targets. This attack is used to eavesdrop on the connection between the targets if the security was downgraded or the key got compromized.

3.4.1 MITM Attack

The specification of BLE (SIG, 2021) built the different association models to protect the devices against Machine-in-the-Middle (MitM) attacks and passive eavesdropping attacks. Passive eavesdropping attacks are sniffing the radio channels and trying to follow the connection. This attack is only able to record the transmitted packets and is not able to alter them. MitM attacks, on the other hand, create a connection with both target devices while spoofing the addresses of the target devices. MitM attacks either relay the transmitted packets, which is called a passive MitM, or alter packets and drop the packet during transmission, which is called an active MitM attack. The legacy pairing association models Just Works, and Passkey Entry do not protect against passive eavesdropping if the attacker sniffs the packets containing the values that are used to calculate the Long Term Key (LTK). The Temporal Key (TK) is the only secret value used to create the LTK. The TK is a six-digit long value chosen by the user, who inserts the value into both devices. This value has low entropy and can be brute forced, as shown in Section 3.3.2. If the Just Works model is used, the key is a fixed value of 0x00, which lets the attacker skip brute-forcing the TK (SIG, 2021). The Secure Connections pairing protocols for Just Works, Passkey Entry, and Numeric Comparison removed the TK from deriving the LTK and only uses the TK for authentication. The LTK is based on Elliptic Curve Diffie-Hellman with the elliptic curve P-256.

This provides passive eavesdropping protection. The Just Work model is still not protected against MitM attacks because it is not authenticated. Secure Connections differentiates between authenticated, e.g., Passkey Entry, Numeric Comparison, and not authenticated, e.g., Just Works association models. The usage of authenticated models is indicated with the MIM protection flag in the authentication requirements field of the pairing request and response packets. The Numeric Comparison model protects against MitM attacks because it shows a random number on the displays of both devices. The connection is compromised with a MitM attack if values do not match. Therefore, the user is responsible to check if the values match. Some lazy users might press the acknowledge button without checking, which nullifies the authentication (SIG, 2021). The Out of Band (OOB) association model of both pairing methods outsources parts of the pairing process to the OOB protocol, e.g., Near Field Communication (NFC). Therefore the MitM and passive eavesdropping protection is dependent on the OOB protocol. If Secure Connections pairing is used, passive eavesdropping protection is OOB independent because the OOB is only used for authentication, and the LTK is based on Elliptic Curve Diffie- Hellman. MitM attacks are mainly based on spoofing and downgrade attacks described in Sections 3.2.1 and 3.3.1. Downgrade attacks use the Just Works model, which is vulnerable to MitM attacks. An attacker connected to the Central and Peripheral with the Just Works model can relay, alter, and drop packets. MitM attacks with the other association models can be performed if the attacker can gather the LTK with the possibilities discussed in the Sections 3.3.2 and 3.3.3. Although to counter MitM attacks pairing is needed, which is not mandatory by BLE to transmit messages. The GATT architecture only requires pairing if the device wants to access an authenticated, encrypted, or encrypted and authenticated attribute. Therefore, the attacker pairs with the Peripheral and then shows the Central a cloned GATT profile with unauthenticated plain text attributes. The malicious identity needs to force Just Works pairing with the Peripheral by indicating that he does not have any input or output capabilities. This results in a encrypted connection between the advesary and the Peripheral and the connection between the Central and the target perfomed as plain text. The adversaty is then able to log, relay, drop and alter packets on the Central side of the MitM (Wang et al., 2020).

4 CONCLUSION

We examined various attacks on BLE and built a threat model in order to identify attack vectors. The research shows that BLE Specification provides security features to create devices with secure BLE data transfers. Nevertehless, the devices can only keep their valuable data confidential if the provided security features are used. Our findings show that the security features are not always properly applied or the security is bypassed due to bugs in the implementation of the BLE specification. In the course of our research we designed a proposal for a BLE penetration testing tool.

4.1 Outlook

The next steps regarding our threat model practical are tests of the addressed attack vectors on BLE devices. Consequently, the knowledge acquired from penetration testing can be used to calculate the severity scores using the Common Vulnerability Scoring System[28]. In order to perform the penetration tests we plan to develop the aforementioned framework which aims to offer user-freindly penetration testing. The BLE Berry tool provides various features like BLE radio sniffing, device scanning, GATT profile scanning, and passive MitM.

REFERENCES

Antonioli, D., Tippenhauer, N. O., and Rasmussen, K. (2019a). The knob is broken: exploiting low entropy in the encryption key negotiation of bluetooth br/edr. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, page 1047–1061, USA. USENIX Association.

Antonioli, D., Tippenhauer, N. O., and Rasmussen, K. (2020a). Key negotiation downgrade attacks on bluetooth and bluetooth low energy. *ACM Trans. Priv. Secur.*, 23(3).

Antonioli, D., Tippenhauer, N. O., Rasmussen, K., and Payer, M. (2020b). Blurtooth: Exploiting cross-transport key derivation in bluetooth classic and bluetooth low energy. *CoRR*, abs/2009.11776.

Antonioli, D., Tippenhauer, N. O., and Rasmussen, K. B. (2019b). The {KNOB} is broken: Exploiting low entropy in the encryption key negotiation of bluetooth {BR/EDR}. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1047–1061. USENIX.

Barua, A., Al Alamin, M. A., Hossain, M. S., and Hossain, E. (2022). Security and privacy threats for bluetooth

[28]https://www.first.org/cvss/, accessed 2024-01-21

low energy in iot and wearable devices: A comprehensive survey. *IEEE Open Journal of the Communications Society*, 3:251–281.

Bräuer, S., Zubow, A., Roshandel, M., Mashhadi Sohi, S., and Zehl, S. (2016). On practical selective jamming of bluetooth low energy advertising.

Cominelli, M., Patras, P., and Gringoli, F. (2020). One gpu to snoop them all: a full-band bluetooth low energy sniffer.

Garbelini, M. E., Wang, C., Chattopadhyay, S., Sun, S., and Kurniawan, E. (2020). Sweyntooth: Unleashing mayhem over bluetooth low energy. In Gavrilovska, A. and Zadok, E., editors, *2020 USENIX Annual Technical Conference, USENIX ATC 2020, July 15-17, 2020*, pages 911–925. USENIX Association.

Healey, R. and Ryan, M. (2020). Hacking electric skateboards: vehicle research for mortals. https://media.defcon.org/DEFSkateboards.pdf, accessed 2023-12-07.

Jasek, S. (2016). Gattacking bluetooth smart devices. https://www.blackhat.com/docs/us-16/materials/us-16-Jasek-GATTacking-Bluetooth-Smart-Devices-Introducing-a-New-BLE-Proxy-Tool.pdf, accessed 2024-02-11.

Kwon, G., Kim, J., Noh, J., and Cho, S. (2016). Bluetooth low energy security vulnerability and improvement method. pages 1–4.

Lounis, K. and Zulkernine, M. (2019). Bluetooth low energy makes "just works" not work. In *3rd Cyber Security in Networking Conference, CSNet 2019, Quito, Ecuador, October 23-25, 2019*, pages 99–106. IEEE.

Ray, A., Raj, V., Oriol, M., Monot, A., and Obermeier, S. (2018). Bluetooth low energy devices security testing framework. In *11th IEEE International Conference on Software Testing, Verification and Validation, ICST 2018, Västerås, Sweden, April 9-13, 2018*, pages 384–393. IEEE Computer Society.

Rose, A. and Ramsey, B. (2020). Picking bluetooth low energy locks from a quarter mile away. https://media.defcon.org/DEF2024/DEFPicking - Bluetooth - Low - Energy - Locks - UPDATED . pdf, accessed on 2023-12-11.

Ryan, M. (2013). Bluetooth: With low energy comes low security. In *7th USENIX Workshop on Offensive Technologies (WOOT 13)*, Washington, D.C. USENIX Association.

Sarawi, S., Anbar, M., Alieyan, K., and Alzubaidi, M. (2017). Internet of things (iot) communication protocols : Review.

Sevier, S. and Tekeoglu, A. (2019). Analyzing the security of bluetooth low energy. pages 1–5.

Shostack, A. (2014). *Threat Modeling: Designing for Security*. Wiley.

SIG, B. (2021). Bluetooth core spec version 5.3. Available at https://www.bluetooth.com/specifications/specs/core-specification-5-3/ (10.01.2023).

Skallak, C. (2023). Ble-berry framework: A framework for bluetooth low energy development and penetration testing. Master's thesis, UAS Campus Wien, 1100 Vienna, Austria.

Thiyeb, I., Saif, A., and Al-Shaibany, N. (2018). Ethical network surveillance using packet sniffing tools: A comparative study. *International Journal of Computer Network and Information Security*, 10:12–22.

Uher, J., Mennecke, R., and Farroha, S. (2016). Denial of sleep attacks in bluetooth low energy wireless sensor networks. pages 1231–1236.

Wang, J., Hu, F., Zhou, Y., Liu, Y., Zhang, H., and Liu, Z. (2020). Bluedoor: breaking the secure information flow via ble vulnerability. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, MobiSys '20, page 286–298, New York, NY, USA. Association for Computing Machinery.

Wu, J., Nan, Y., Kumar, V., Tian, D. J., Bianchi, A., Payer, M., and Xu, D. (2020). BLESA: Spoofing attacks against reconnections in bluetooth low energy. In *14th USENIX Workshop on Offensive Technologies (WOOT 20)*. USENIX Association.

Zhang, Y., Weng, J., Dey, R., Jin, Y., Lin, Z., and Fu, X. (2020). Breaking secure pairing of bluetooth low energy using downgrade attacks. In *Proceedings of the 29th USENIX Conference on Security Symposium*, SEC'20, USA. USENIX Association.