Journal of Interdisciplinary Innovational Letters
Vol 2 No 1 (2025): 55-74

Analyzing Vulnerabilities and Exploitation in Bluetooth Security of Wireless Communication: A Security Framework

Zaeem Nazir[*1], Muhammad Danish[2], Burhan Ali [3], Syed Pervez Hussnain Shah[4], Syed Samee Hasnain[5], Muhammad Akram Mujahid[6]

[1*]Lectu rer, Faculty of Computer Science, University of Narowal, Narowal, Punjab, Pakistan.
[2,3,5] BSCS Scholar, Department of Computer Science, Superior University, Lahore, Punjab, Pakistan.
[4]Lecturer, Department of Computer Science, Lahore Leads University, Lahore, Punjab, Pakistan.
[6]Assistant Professor, Department of Information Science, DSNT, University of Education, Lahore, Punjab, Pakistan.
***Corresponding author:**zaeem.nazir@gmail.com

Abstract

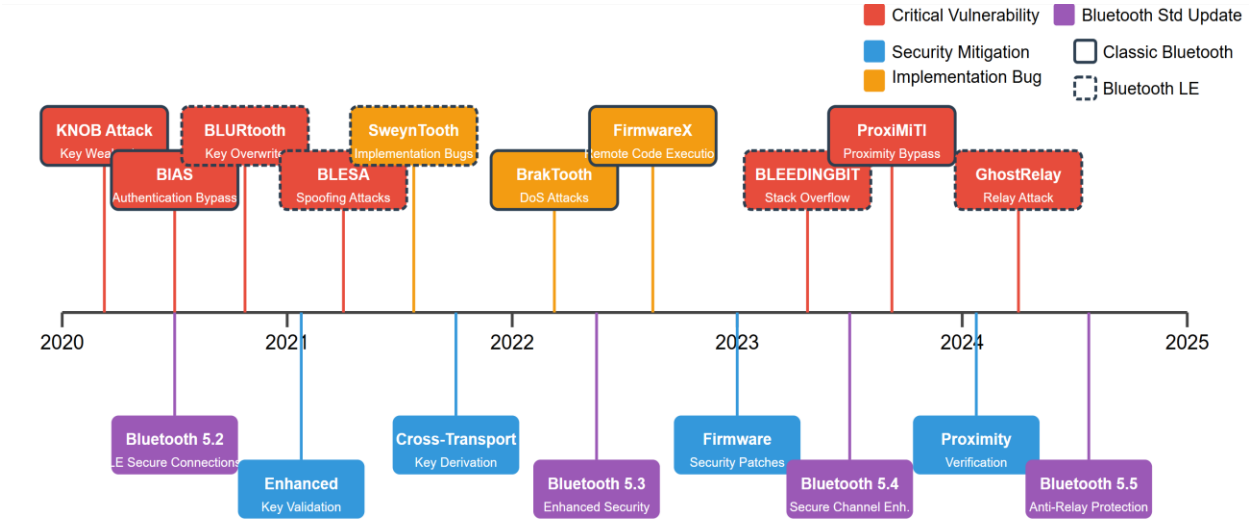
Bluetooth has emerged as a ubiquitous wireless technology in personal, IoT, and industrial devices. During the period 2020–2025, several security vulnerabilities were identified in both Classic Bluetooth (BR/EDR) and Bluetooth Low Energy (BLE) protocols, which support passive eavesdropping, man-in-the-middle (MITM), device impersonation, denial-of-service (DoS), and arbitrary code execution attacks. This survey study provides a current review of specification-level vulnerabilities (e.g., KNOB, BIAS, BLURtooth) and implementation flaws (e.g., SweynTooth, BrakTooth, BlueFrag), describes reported exploitation methods and case studies, and discusses countermeasures developed to counter these threats. We contrast the security designs of Classic and BLE, recognize upcoming trends like tighter key management and increased testing, and suggest coming defenses. We intend to enlighten professionals regarding the modern landscape of Bluetooth security and shape stronger wireless communications protections.

Keywords: Bluetooth; Bluetooth Low Energy; Wireless Security; Vulnerabilities; Exploitation; Man-in-the-Middle; Denial-of-Service; IoT Security; IEEE 802.15.1.

1. Introduction

Bluetooth is the ubiquitous short-range wireless technology in billions of devices ranging from phones and laptops to wearables, medical monitors, cars, and IoT sensors (Prakash et al., 2017). It has two primary variants: Classic Bluetooth (BR/EDR), for constant high-rate links (e.g., streaming audio and human interface devices), and Bluetooth Low Energy (BLE) (Ghori et al., 2020), for power-constrained intermittent connections (e.g., fitness trackers and sensors for environmental sensing). In order to ensure confidentiality, integrity, and authenticity (Danish et al., 2025), Bluetooth utilizes pairing processes, mutual authentication, and link-layer encryption based on Elliptic-Curve Diffie–Hellman (ECDH) and symmetric ciphers.

Figure No 1: Timeline of Bluetooth security (2020–2025), showing critical specification-level vulnerabilities (red), controller/host implementation bugs (orange), Bluetooth Core Specification updates (purple), and auxiliary mitigations (blue)



In spite of all these protections, the 2020–2025-time period saw a waterfall of both specification-level vulnerabilities and implementation flaws that eroded Bluetooth's security. Purposes like KNOB (Key Negotiation of Bluetooth), BIAS (Bluetooth Impersonation AttackS), and BLURtooth revealed flaws in the protocol logic, whereas firmware flaws like SweynTooth, BrakTooth, and BlueFrag facilitated denial-of-service, remote code execution, and keystroke injection on vulnerable devices (Hamid et al., 2023) . To that, the Bluetooth Special Interest Group (SIG) and chipset and operating system (OS) providers introduced a set of countermeasures: ranging from strengthening minimum key lengths and imposing mutual authentication to deploying Bluetooth Core spec updates (5.2–5.5) and vendor-side firmware patches.

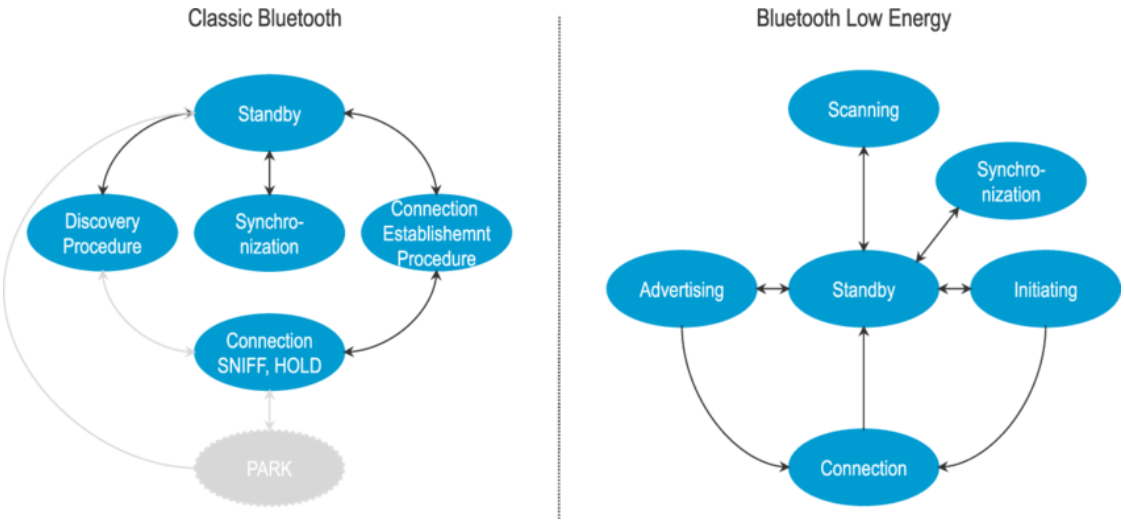
This paper gives a holistic, timely survey of Bluetooth security between 2020 and 2025. We (1) enumerate significant flaws in Classic and BLE, (2) outline reported exploitation methods and case studies, (3) canvass specification revisions and implementation-level protections, and (4) recognize developing architectural trends and future protection mechanisms (Hamid et al., 2023). The rest of this paper follows this structure. Section II introduces the Bluetooth security architecture. Section III explains specification-level and implementation-level vulnerabilities unearthed in 2020–2025. Section IV considers the countermeasures standards updates, firmware/OS patches, and best practices and assesses their effectiveness. Section V explores trends revealed by recent events and indicates possible future protections. Section VI concludes with significant lessons and unresolved challenges (Khaliq et al., 2024).

2. Background: Bluetooth Security Architecture

2.1 Classic vs. BLE Protocols

Classic Bluetooth (BR/EDR) and BLE have some architectural similarities but vary in application and security implementations. Classic uses a fixed 79-channel hop set with greater data rate, generally employed for continuous connections (audio, HID devices), while BLE employs 40 channels with a light design optimized for low power and intermittent connectivity. Classic and BLE have distinct link layer protocols (Link Manager Protocol, LMP for Classic; Link Layer, LL for BLE) and pairing processes, though the Bluetooth Core Specification more closely coordinates their security methodologies (for example, each has "Secure Connections" mode utilizing Elliptic-Curve Diffie–Hellman encryption in recent editions).

Figure No 2: Comparison of Bluetooth Versions



Security Features and Modes Previous versions of Bluetooth (Classic pre-2.1) employed a PIN-based pairing that generated a link key, but it was demonstrated to be vulnerable to eavesdropping

if the PIN was low-entropy. Current Bluetooth employs stronger pairing protocols. In Classic, Secure Simple Pairing (SSP) was added in v2.1, employing modes such as Just Works, Numeric Comparison, Passkey, etc., with elliptic-curve (P-192) Diffie–Hellman key exchange. BLE pioneered its own pairing in v4.0 and subsequently took on LE Secure Connections (LESC) in v4.2, utilizing ECDH with elliptic curve P-256, echoing Classic's Secure Connections (which also transitioned to ECDH P-256 in Bluetooth v4.2). Pairing authentication can be guaranteed through user-verified passkeys or numeric match, with the sole exception being Just Works mode (exposed to MITM by design for not having authentication). Both Classic and BLE provide link-layer traffic encryption: Classic employs the E0 stream cipher with 128-bit keys generated from the pairing link key, while BLE employs AES-CCM with 128-bit session keys (refreshed for each session from a long-term key). Legacy "Secure Simple Pairing" in Classic employed a 16-byte link key (generated using P-192) and encryption keys negotiable to lower sizes for interoperability. BLE's initial pairing (Legacy BLE pairing) may employ a static TK (e.g., "000000") in Just Works and thus be vulnerable to passive eavesdropping if an attacker overheard the pairing exchange (Ibrar et al., 2024) (Malik et al., 2024) (zafar et al., 2023). Table 1 contains a summary comparison.

2.2 Privacy and Identity

Bluetooth Low Energy (BLE) has support for address randomization through Private Resolvable Addresses, where a device's MAC address is updated periodically through an Identity Resolving Key (IRK) shared during pairing (Barua et al., 2022). This mechanism prevents passive tracking by third-party observers. Classic Bluetooth devices, on the other hand, typically employ a static Bluetooth Device Address (BD_ADDR), but some variations may use variable clock offsets in inquiry scans to hide identity. Note that these privacy capabilities do not block active protocol-level attacks; they only slightly make it more difficult for targeted attacks by hiding a device's long-term identifier.

2.3 Host and Controller

The Bluetooth architecture consists of a Host, within the operating system (L2CAP, SDP and higher layers implementation), and a Controller, embedded within the radio firmware (implementation of the link-layer protocols LMP for Classic, LL for BLE—and baseband functionality) (Laghari et al., 2024). These two spaces exchange information via the Host-Controller Interface (HCI). Cryptographic processes e.g., key generation during pairing and link encryption are synchronized between Controller and Host: e.g., in BLE, the Controller encrypts link-layer frames with AES-CCM using session keys provided by the Host. Numerous vulnerabilities result from proprietary Controller firmware implementations of LMP/LL processes, while others take advantage of bugs in the Host's protocol parsers. Interestingly, the BrakTooth series of DoS and RCE vulnerabilities exploited LMP handling in some SoC firmwares (Administrador, 2021), whereas BlueFrag was an out-of-bounds write in Android's Host-side BLE L2CAP handler (*CVE-2020-0022*, n.d.). Controller-side flaws are usually mitigated by firmware

updates which may be challenging to deploy whereas Host-side flaws can typically be fixed through OS patches.

2.4 Threat Model

In our threat model, we consider an attacker within Bluetooth's normal radio range (on the order of tens of meters) who can potentially attack a target by intercepting or manipulating messages during pairing to undermine or bypass authentication; by utilizing malformed Link Manager Protocol (LMP) or Link Layer (LL) packets either pre- or post-pairing to cause denial-of-service or even remote code execution; or by impersonating a legitimate device (through impersonation or man-in-the-middle attacks) to relay, alter, or inject traffic between sincere endpoints.

Table No 1: Classic vs. BLE Security Feature Comparison

Aspect	Classic Bluetooth (BR/EDR)	Bluetooth Low Energy (BLE)
Primary Usage	Continuous high-rate connections (audio, peripherals)	Low-power intermittent connections (sensors, IoT, etc.)
Frequency Channels	79 channels (1 MHz); FHSS at 1 600 hops/s	40 channels (2 MHz); adaptive frequency hopping (AFH)
Pairing Methods	SSP v2.1+: Just Works, Passkey, OOB, Numeric Comparison (ECDH P-192); Legacy PIN (pre-2.1)	LE Legacy Pairing v4.0: Just Works, Passkey, OOB (TK); LE Secure Connections v4.2+: Passkey, Numeric Comparison, OOB (ECDH P-256)
Authentication	Unilateral in legacy pairing (master→slave); mutual in Secure Connections (v4.2+). Just Works has no MITM protection.	Mutual optional: Just Works has no MITM protection; Passkey/Numeric Comparison provides authentication. Secure Connections adds Numeric Comparison for MITM protection.
Encryption	E0 stream cipher; negotiable 1–16 byte key length (128 bit max); key derived from link key.	AES-CCM; fixed 128 bit session keys (STK/LTK) derived from pairing (TK or ECDH + nonces).
Key Management	Long-term Link Key established via pairing; SSP uses ECDH (P-256); legacy unit-key reuse deprecated.	Long-Term Key (LTK) for encrypted sessions, distributed at pairing; devices may store LTK for reconnections; supports IRK (privacy) and CSRK (signing) distribution.
Privacy Feature	No native address randomization (fixed BD_ADDR); upper layers may implement non-static identities.	Private Resolvable Address via IRK; devices periodically change MAC to hinder tracking.
Notable Additions	SSP (v2.1) eliminated fixed-PIN weakness; Secure Connections (v4.2) with ECDH P-256; Cross-Transport Key Derivation (CTKD).	LE Secure Connections (v4.2) with ECDH P-256; LE Privacy (v4.2) adds IRK addresses; Secure Connections Only mode (Mode 1, Level 4).
Vulnerability Examples (2020+)	KNOB (key weakening); BIAS (impersonation); BLURtooth (CTKD overwrite); BrakTooth (LMP bugs).	SweynTooth (BLE LL/L2CAP crashes); BLESA (reconnection spoofing); CVE-2023-45866 (unauthenticated HID injection).

These attackers can operate prior to the establishment of any bond, during an encrypted session (Bhatti et al., 2023; Zia et al., 2025; Tehreem et al., 2025; Riaz et al., 2025), or completely without pairing based on implementation flaws in firmware or host stacks. Table II (Section III) classifies these attack vectors pairing-stage manipulation, link-layer exploitation, and impersonation/MITM and identifies their respective tactics and effects. While Bluetooth's limited range restricts the scale of remote attacks, these proximity-based attacks are still a major threat in densely populated public areas and in targeted situations.

2.5 Legacy and Secure Connections: It should be noted that several contemporary devices implement both legacy and secure connections for compatibility with older systems. For instance, a Bluetooth 5.0 phone would still resort to legacy pairing when connecting to a BT 2.0 headset. This backward compatibility occasionally exposes devices to downgrading attacks. For example, an attacker might try to induce a secure pairing to downgrade to legacy SSP or a weaker key, as with the KNOB and BIAS attacks later described. The evolution of the Bluetooth specification has progressively eliminated known weak mechanisms (e.g., unit keys deprecated, minimum key length recommended), but the existence of legacy support in devices ensures that those legacy weaknesses can be exploited if not adequately mitigated.

In short, Classic and BLE are different in some details of the protocol but both are attacked targets. We discuss next the specific vulnerabilities found in recent years, how they are used to attack, and how they affect Bluetooth security

2.6 Vulnerabilities and Exploitation Techniques (2020–2025)

There have been several Bluetooth vulnerabilities disclosed since 2020 that hit both Classic BR/EDR and BLE protocols. These consist of specification-level issues that break security assumptions (which tend to impact a large number of devices that properly implemented the broken spec), along with implementation bugs in individual vendor stacks. We list the significant vulnerabilities by attack type in Table 2, and provide a detailed description of them below along with concrete-world exploitation scenarios and case studies.

Table No 2: Vulnerabilities and Exploitation Techniques (2020–2025)

Attack Category	Description and Tactics	Example Vulnerabilities (Year)	Impact
Weak Key Negotiation	Forcing devices to agree on a weak encryption key or cipher suite, enabling brute-force decryption.	KNOB (2019/2020): Manipulates LMP negotiation to force 8-bit session keys.	Enables passive eavesdropping on Classic links by brute-forcing the 256-possible key, allowing decryption and packet injection.
Impersonation / MITM	Spoofing identity to pair or connect as a trusted device, often	BIAS (2020): Impersonates previously paired device without key by abusing legacy Secure Connections authentication. BLESA (2020): Abuses	Unauthorized access, interception, or data manipulation. E.g., session takeover via BIAS;

	exploiting lack of mutual authentication.	optional reconnection auth to spoof BLE devices.BlueDucky (CVE-2023-45866, 2023): Unauthenticated fake HID pairing and keystroke injection.	spoofed commands or data theft via BLESA and BlueDucky.
Key Overwrite / Downgrade	Replacing or downgrading an existing trusted key with a weaker one, facilitating further attacks.	BLURtooth (2020): CTKD flaw on dual-mode devices allows BLE Just-Works pairing to overwrite authenticated BR/EDR key.	Elevates attacker privileges by substituting a high-security key with a lower-security one, enabling MITM or unauthorized service access on the overwritten transport.
Denial-of-Service (DoS)	Crashing or deadlocking the Bluetooth stack via malformed packets, flooding, or protocol abuse.	SweynTooth (2020): 12 BLE LL/L2CAP vulnerabilities causing firmware crashes or deadlocks on multiple SoCs.BrakTooth (2021): 16 LMP flaws triggering asserts or infinite loops in Classic controllers.	Disrupts device operation (e.g., audio devices, medical monitors, industrial sensors become unresponsive). Recovery often requires manual restart, enabling persistent service denial.
Arbitrary Code Execution	Exploiting memory-corruption bugs in firmware or OS stacks to run attacker-controlled code on the target device.	BlueFrag (2020): Android L2CAP buffer overflow (CVE-2020-0022) allows zero-click RCE.BrakTooth V15 (2021): ESP32 heap overflow (CVE-2021-28139) enables code injection and NVRAM erasure.	Full device compromise within radio range, including malware deployment on smartphones, IoT takeover, firmware modification, or backdoor installation.

2.7 Specification-Level Vulnerabilities (Classic & BLE)

Specification-level attacks take advantage of logical errors in the Bluetooth standard itself, making any compliant implementation vulnerable until the underlying specification is updated or vendors implement workarounds. In 2019 and 2020, three groundbreaking vulnerabilities—KNOB, BIAS, and BLURtooth—were published, each breaking basic security assumptions in pairing and key management. We outline these below.

2.7.1 KNOB (Key Negotiation of Bluetooth)

The KNOB attack, made public in 2019 and explained in 2020, attacks Classic Bluetooth's link-key negotiation (*Tech Xplore - Technology and Engineering News*, n.d.). The Core Specification allowed BR/EDR devices to negotiate encryption keys as short as one byte (8 bits) for backward compatibility, without protection of the negotiation integrity. An attacker who is present during connection establishment can manipulate Link Manager Protocol (LMP) messages to make both endpoints accept an 8-bit session key. Then, the attacker brute-forces the 256-choice key in real-time, overcoming encryption and allowing for decryption or injection of traffic even on links securely paired in the past. Broadcom, Qualcomm, Apple, Intel, and other top chipsets

were made vulnerable (*Tech Xplore - Technology and Engineering News*, n.d.). In order to counteract KNOB, the Bluetooth SIG amended the Core Specification (erratum to v5.1, included in v5.2) to necessitate a minimum of 56 bit (7 octet) key length and suggested firmware patches to mandate this requirement.

2.7.2 BIAS (Bluetooth Impersonation Attacks)

Revealed in 2020, BIAS exploits two specification weaknesses in historic Secure Connections (Classic through v5.0): (i) unilateral authentication where the slave is authenticated by the master, but not vice versa, and (ii) liberal role switching after link setup (Antonioli et al., 2020). A passive attacker who was not present at the original pairing can masquerade as a prior paired device by authenticating initially as the slave (pretending "Alice" to "Bob") then role-swapping to appear as "Bob" to "Alice," never having to demonstrate knowledge of the long-term key. This provides an opportunity for man-in-the-middle positioning, circumventing mutual authentication assurances. More than thirty devices from leading vendors were identified as vulnerable (Antonioli et al., 2020). Firmware workaround—like preventing role switching prior to both sides authenticating—were implemented subsequent to late 2019, and Bluetooth Core v5.2/5.3 contain specification clarifications to mandate bidirectional authentication.

2.7.3 BLURtooth (CTKD Key Overwrite)

BLURtooth takes advantage of the Cross-Transport Key Derivation (CTKD) feature that was added with v4.2 to synchronize Classic and BLE keys. In CVE-2020-15802, an attacker performs a BLE Just-Works pairing with reduced security and, through CTKD, overwrites a previously stored Secure Connections BR/EDR link key with the unauthenticated BLE key (*CERT/CC Vulnerability Note VU#589825*, n.d.). This downgrade enables subsequent MITM or unauthorized service access on the Classic transport. In September 2020, the SIG released an addendum that banned CTKD overwrites with weaker-security keys and included qualification tests; vendors now disable CTKD unless both transports support authenticated pairing or equivalent key strength.

2.7.4 Blacktooth (Silent Link Establishment)

Most recently, the 2022 Blacktooth study showed how even Secure Connections can be manipulated by chaining corner cases of a protocol to switch "in silence" without any user intervention into pairing and later to privileged profiles (Zhang & Lin, 2022). While no such public firmware exploit exists, Blacktooth emphasizes that specification imprecisions like reconnection optionals on prompting might still leave room for illegal pairing. All of these attacks stem from compliant-with-spec logic errors, not implementation issues, on devices in Bluetooth v4.0–5.2. Their widespread influence led to SIG errata and reinforced qualification requirements. Table III in Section IV lists the corresponding mitigation techniques.

2.8 Implementation Vulnerabilities and Case Studies

In addition to specification errors, various vendor-specific implementation flaws in Bluetooth stacks have been reported since 2020. These vulnerabilities most commonly due to reused chipset SDK code or lack of proper input validation allow Denial-of-Service (DoS), remote code execution, and other attacks. We emphasize the most critical families and their impact in the real world.

2.8.1 SweynTooth (2020)

SweynTooth includes twelve vulnerabilities in the BLE Link Layer and L2CAP implementations of seven or more SoC vendors (TI, NXP, Cypress, Dialog, Microchip, etc.) (*SweynTooth Vulnerabilities | CISA*, 2020), (Administrador, 2021). With specially crafted packets (e.g., CVE-2019-19195, invalid L2CAP fragment; CVE-2019-17517, buffer overflow), an attacker can deadlock or crash the firmware, often needing a manual power cycle. A particularly dangerous flaw (CVE-2019-19194) bypasses Secure Connections pairing, providing unauthorized read/write access to device functions. Affected products from medical wearables to industrial sensors required urgent SDK updates and field firmware patches coordinated by CISA.

2.8.2 BrakTooth (2021)

The BrakTooth family attacks BR/EDR controllers' Link Manager Protocol (LMP). The 13 SoC chipsets from 11 vendors were tested by researchers and identified 16 vulnerabilities (Fatima Yousaf et al., 2024). All of these can be initiated without pairing using malformed LMP packets that produce crashes, firmware hangs, or, in Espressif's ESP32 (CVE-2021-28139), a heap-overflow allowing arbitrary code execution and erasure of NVRAM. Billions of devices laptops to IoT modules remained vulnerable until firmware and driver patches were published by vendors including Intel, Qualcomm, and Espressif.

2.8.3 BlueFrag (CVE-2020-0022)

BlueFrag is an unauthenticated RCE on Android's Bluetooth daemon, fixed in February 2020 (*CVE-2020-0022*, n.d.), (*New "BLESA" Bluetooth Vulnerability Could Affect Billions of IoT Devices, Researchers Warn*, n.d.). One malformed L2CAP packet causes a buffer overflow in reassemble_and_dispatch, allowing zero-click code execution with Bluetooth rights—even for a non-discoverable device. With a CVSS of 8.8, BlueFrag showed how top-level OS stack defects can be just as dangerous as spec problems. Android 8–10 devices were affected until security updates were applied.

2.8.4 Other Noteworthy Flaws

Other implementation vulnerabilities are Linux BlueZ privilege escalation through Unix socket mishandling (CVE-2021-0129) and Android "GREYHOUND" SDP parsing flaws resulting in DoS or potential RCE [12]. Side-channel and privacy research has also shown tracking regardless of BLE address randomization.

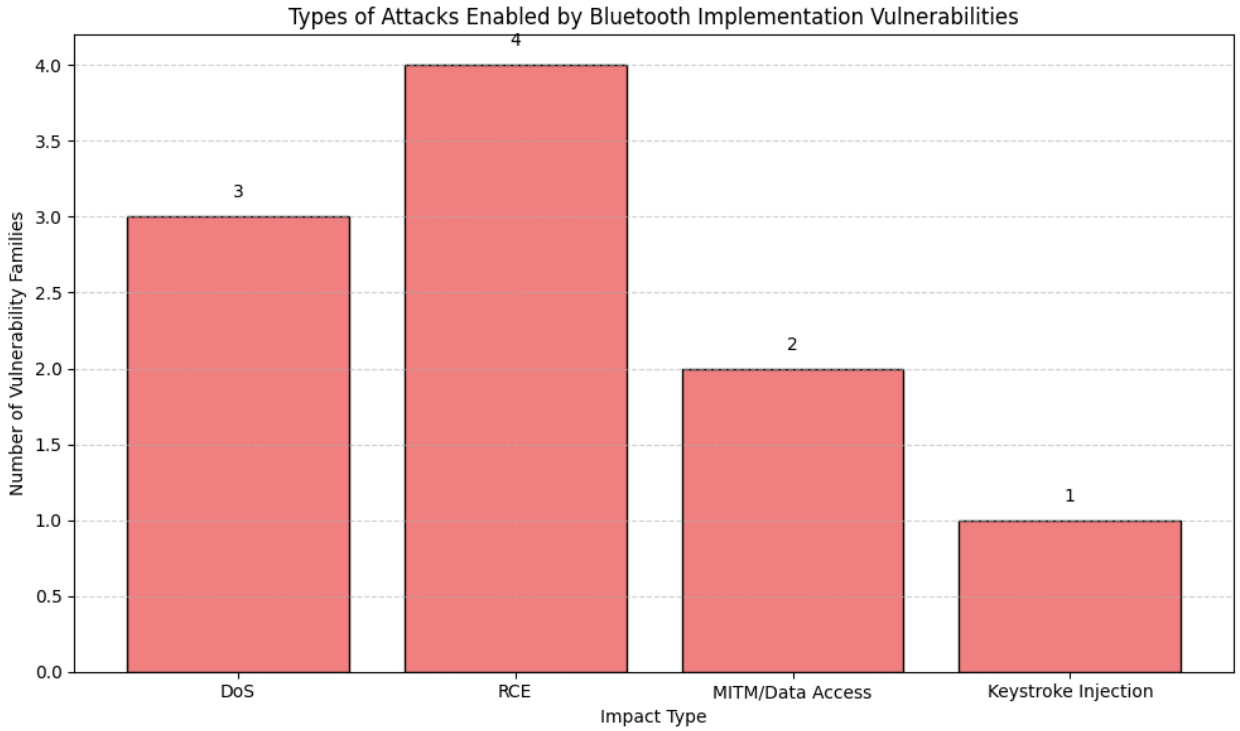
2.9 Case Study—Combined Attacks

Chaining weaknesses can produce devastating exploits. A good illustration is that an attacker applies BIAS to spoof a device and subsequently uses KNOB during session establishment to compel an 8-bit key hence obtaining MITM access and decrypting all the traffic (Antonioli et al., 2020), (*Tech Xplore - Technology and Engineering News*, n.d.).

2.10 Case Study—BlueDucky Keystroke Injection.

CVE-2023-45866 permits an unauthenticated spoofed HID keyboard to silently pair and inject keystrokes (*CVE-2023–45866: A Critical Bluetooth Security Flaw — EXPLOIT | by Hacker's Dump | Medium*, n.d.). Affected platforms (Android, Linux, iOS) fixed their pairing logic to make it user-confirmable, thwarting "wireless Rubber Ducky" attacks that would surreptitiously run arbitrary commands.

Figure No 3: Types of Attacks by Enabled Bluetooth Implementation Vulnerabilities



3. Security Enhancements and Mitigations (2020–2025)

The series of well-publicized Bluetooth vulnerabilities between 2020 and 2025 has prompted a multi-faceted response involving upgrades to the Core Specification, firmware patches by chipset manufacturers, operating-system corrections, and best-practice advice from the Bluetooth SIG. Table III outlines the major threats and their respective mitigations (Hamid et al., 2025) (Raza et al., 2025) (Aamir et al., 2025).

3.1 Specification Updates

Following KNOB, SIG released an erratum to Bluetooth 5.1 (included in 5.2) requiring a minimum key length of 56 bits (7 octets) for BR/EDR links, which effectively eliminated the possibility of negotiating sub-56-bit keys (*Tech Xplore - Technology and Engineering News*, n.d.). Numerous vendors now require full 128-bit keys. Bluetooth 5.3 (mid-2021) added two HCI improvements: a "minimum key size" control to enable the host to reject weak key lengths, and an encryption indicator to inform the host of the negotiated key size. Combined, these enable devices to actively deny weak-key connections.

BIAS triggered more nuanced specification clarifications than a lone revision. Security advisories urged vendors to implement mutual authentication by preventing role switches prior to both sides authenticating Secure Connections. By 5.2/5.3 releases, test procedures officially test for BIAS-like activity, and numerous controllers support a "Secure Connections Only" mode for LE devices, which rejects legacy or unauthenticated link establishment.

BLURtooth initiated an official addendum (September 2020) that forbids CTKD from overwriting a more secure key with a less secure one (*CERT/CC Vulnerability Note VU#589825*, n.d.). Qualification test suites now confirm that dual-mode devices either deactivate CTKD when transports are different in authentication level or necessitate user re-authentication prior to cross-transport key derivation. Outside these focused patches, Bluetooth 5.2 brought Enhanced ATT (EATT), which prevents some DoS attacks by enabling simultaneous ATT transactions, and Bluetooth 5.2/5.3 established Isochronous Channels complete with bonding and broadcast-stream encryption support, optional—that guarantee new functionality is designed secure.

3.2 Firmware and OS Patches

In the implementation space, chipset and OS suppliers have provided patches for nearly all significant vulnerabilities:

SweynTooth. In early 2020, SoC suppliers (TI, NXP, Cypress, Dialog, Microchip, etc.) published SDK updates fixing twelve BLE-link-layer and L2CAP vulnerabilities ranging from buffer overflows to "Zero LTK Installation" bypasses (*SweynTooth Vulnerabilities | CISA*, 2020). OEMs then built these patches into device firmware, frequently through over-the-air updates or field recalls in safety-critical applications.

BrakTooth. After the 2021 revelation of sixteen LMP bugs on thirteen SoCs, Espressif fixed its ESP32 firmware (including the RCE-facilitating heap overflow, CVE-2021-28139), and Intel, Qualcomm, and others provided driver/firmware updates through Windows Update and vendor bulletins (Fatima Yousaf et al., 2024). While some end-of-life chips are unpatchable and therefore vulnerable to ongoing DoS the prompt response highlighted the worth of LMP fuzz testing in development.

Mobile OS Patches. Android's February 2020 security update patched BlueFrag (CVE-2020-0022) by fixing the L2CAP parser in the Bluetooth daemon (*CVE-2020-0022*, n.d.). iOS 13.4 removed the BLESA reconnection vulnerability (CVE-2020-9770) (*New "BLESA" Bluetooth Vulnerability Could Affect Billions of IoT Devices, Researchers Warn*, n.d.), and later Android and Linux patches closed analogous reconnection-authentication vulnerabilities. Windows' BLE stack, as designed, was immune to BLESA.

3.3 Hardening Measures

Various platform providers have added specific patches with architectural hardening. Beginning in Android 11, the Bluetooth stack executes within a very limited sandbox with ASLR enforced. Apple uses its Secure Enclave to persistently store long-term keys on HomePod and later devices, making key extraction impossible even when the primary processor is compromised. These types of steps don't explicitly patch individual vulnerabilities but raise the bar substantially for exploitation.

Figure No 4: Bluetooth Vulnerabilities Reported by Year

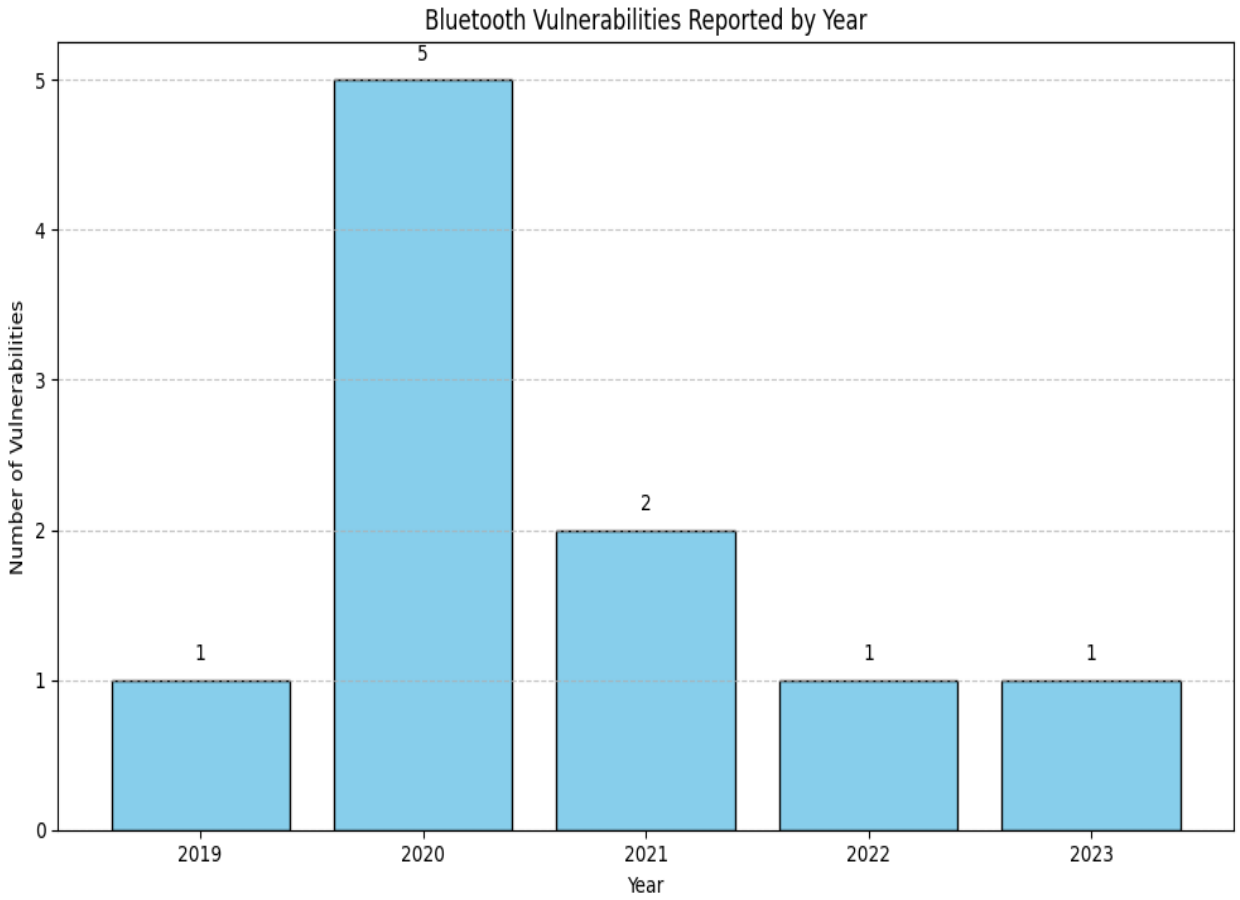


Table No 3: Hardening Measures

Vulnerability (Year)	Affected Area	Mitigation / Response	References
KNOB (2019/2020)	Classic BR/EDR links up through v5.1	Updated spec to require at least a 56-bit key; firmware now rejects smaller keys; test suites check key negotiation.	TechXplore, Argenox
BIAS (2020)	Classic Secure Connections (v4.2–5.0)	Firmware patches enforce full mutual authentication and block role-switches before auth; new devices use "Secure Connections Only."	WeLiveSecurity, Tarlogic
BLURtooth (2020)	Dual-mode devices using CTKD (v4.2–5.0)	Spec clarified that weaker keys cannot overwrite stronger ones; devices either disable CTKD or require matching security levels.	CERT/KB
SweynTooth (2020)	BLE chipsets (link layer and L2CAP)	Chip vendors released SDK updates; OEMs rolled out firmware fixes; more extensive fuzz testing added to development.	CISA, UK NHS Digital
BLESA (2020)	BLE reconnection logic in various OS stacks	Clarified reconnection must be authenticated; iOS 13.4 and Android/Linux updates enforce this; Windows already safe.	Bitdefender
BrakTooth (2021)	Classic BR/EDR controller firmware (LMP)	Espressif, Intel, Qualcomm, etc. pushed firmware and driver updates; new robustness tests added for LMP messages.	The Hacker News, ASSET Group
BlueFrag (2020)	Android's BLE stack (L2CAP parser)	February 2020 Android security patch fixed the overflow; later versions hardened the Bluetooth daemon's memory safety.	Android Open Source, CVEDetails

BlueDucky (CVE-2023-45866)	HID pairing logic in Classic & BLE	Linux and Android now require user confirmation for keyboard pairing; iOS 17 patched the Magic Keyboard issue.	Medium
Blacktooth (2022)	Classic Secure Connections pairing flow	Under study: proposals include always prompting the user for new pairings and adding stricter freshness checks.	KU ITTC
Other Implementation Bugs (2021–2023)	Various OS and firmware components	Patches rolled out in Linux kernel, Android updates, and other OS releases; ongoing improvements to validate inputs.	N/A

4. Trends and Architectural Improvements

A clear trend in Bluetooth's security evolution is the systematic removal or deprecation of legacy, insecure options. For example, Bluetooth 5.3 formally dropped support for Bluetooth 3.0 High Speed—eliminating a poorly used feature with its own attack surface—and the SIG now strongly encourages "Secure Connections Only" devices that refuse legacy pairing methods. In most IoT implementations, accessories allow only LE Secure Connections (excluding "Just Works" except with an out-of-band channel like NFC), thus removing an entire category of MITM threats at the expense of periodic user hassle (e.g., passkey input or QR code scanning).

Privacy and identity management have also become an integral part of security. As a default, newer BLE devices use address randomization (Private Resolvable Addresses), and Bluetooth 5.4's Periodic Advertising with Responses (PAwR) minimizes tracking even further by synchronizing sensor-network communication under randomized IDs. While not explicit mitigations against protocol exploitation, they demonstrate a design approach that integrates privacy and security with new capabilities.

4.1 Industry Response and Testing

To counter 2020–2025 weaknesses, Bluetooth SIG introduced an Automated Testing Interface (ATI) and extended its Qualification test suite to cover minimum-key-length enforcement (KNOB), CTKD verification (BLURtooth), and resilience tests to malformed LMP/LL packets. Consequently, any new module certified would be required to show immunity to such identified attacks to avoid regression of long-fixed bugs.

Security researchers have shared open-source tools InternalBlue, BTFuzz, BlueHydra, and BLEGuard—that allow developers and operators to fuzz their own stacks and listen for out-of-band pairing sequences that signal BIAS or MITM attempts (Barua et al., 2022). These community resources have sped up vulnerability discovery and continuous integration of security assessments into product lifecycles.

4.2 User Awareness and Practices

User-facing mitigations are still an essential layer. Security bulletins now even regularly instruct end users to update firmware, only accept pairing requests from trusted devices, and turn off Bluetooth in untrusted settings. While certain exploits involve no user action, educated users can even prevent situations like default "0000" PIN use or erroneous re-pairing requests—tell-tale signs of spoofing or downgrade attacks.

4.3 Emerging Countermeasures

In the future, a number of forward-thinking defenses are being researched:

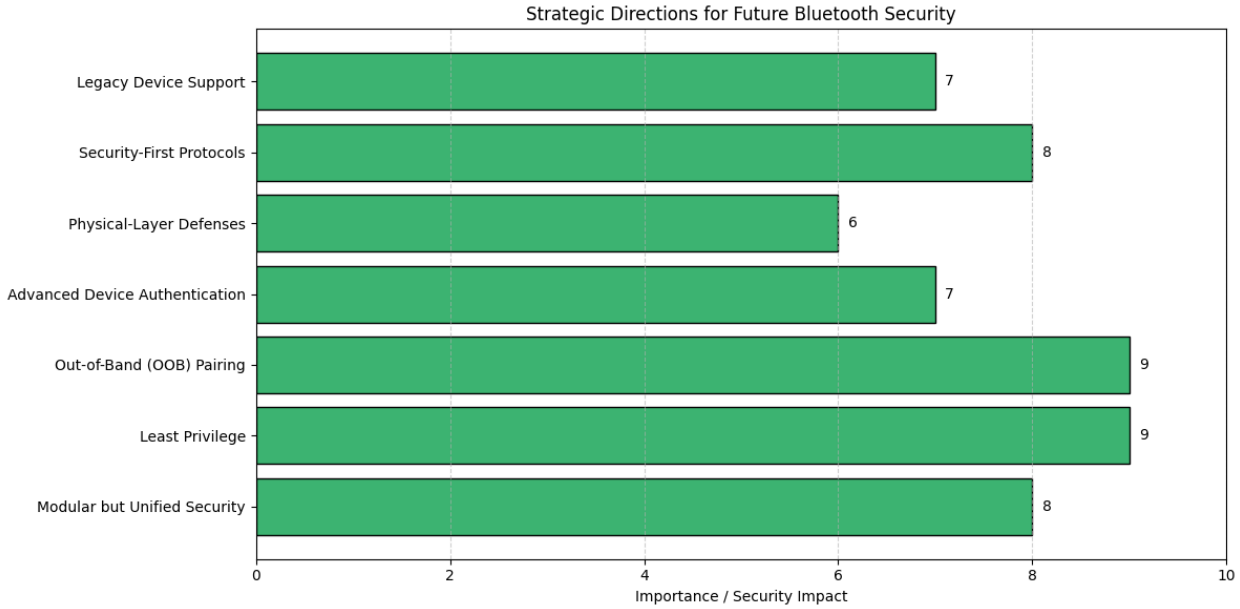
Real-Time Intrusion Detection. There have been suggestions to incorporate anomaly detectors into Bluetooth stacks that mark abrupt key-length reductions or unanticipated role reversals characteristics of KNOB and BIAS and log or abort suspect handshakes. Enhanced Link-Layer Integrity. Engineers have prototyped out-of-band integrity checks similar to WPA3's SAE where a pre-shared fingerprint or certificate excludes downgrade or MITM attacks without degrading user experience.

Firmware Transparency and Virtual Patching. To address BrakTooth, some vendors are making configurable firmware filters visible to prevent out-of-spec LMP messages at the controller level, allowing "virtual patches" to filter out exploit traffic without a complete firmware update. The SIG's Security Expert Group now performs periodic, systematic reviews of the Core Specification, building a positive feedback loop: more scrutiny results in fewer lingering flaws over time. Nevertheless, Bluetooth's legacy-burdened, heterogeneous ecosystem spanning automotive to medical to mesh-network applications means that the attack surface will always be substantial. Ongoing simplification of the protocol (e.g., possible Bluetooth 6 requiring authenticated pairing by default) and careful patch management are necessary to maintain momentum against emerging threats.

4.4 Results and Discussion: Future Security Architectures and Defenses

Industry responses and recent research have clarified some architectural themes in Bluetooth security.

Figure No 5: Strategic Directions for Future Bluetooth Security



Modular but Unified Security: Bluetooth Classic and BLE are more and more converging towards a shared Secure Connections framework, unifying pairing, key exchange, and encryption primitives across transports. This consolidation simplifies security best practices improvements in one area tend to apply directly to the other but risks cross-propagating vulnerabilities, as seen in CTKD/BLURtooth in dual-mode devices. To reconcile consistency with robustness, newer implementations use a "mode separation" strategy: Classic and BLE bonds are treated separately unless both satisfy the same security requirements.

Principle of Least Privilege: OSes are handling Bluetooth as a high-risk subsystem, with stricter isolation and permissions. Android 12, for instance, migrated a large portion of the Bluetooth stack to a low-privilege sandbox and added the "Nearby Devices" runtime permission that keeps arbitrary apps from starting up pairings or accessing user information. Compartmentalizing Bluetooth activity and limiting what processes can communicate with it reduces the attack surface of a compromised stack significantly.

Out-of-Band (OOB) Pairing: Critical use cases like smart locks, payment machines, and medical equipment are moving from "Just Works" to OOB channels (NFC, QR codes) for key exchange. Bluetooth 5.3/5.4 specifically suggest OOB for high-security use cases, as an independent channel cannot be eavesdropped or tampered with over the radio. This transition effectively renders MITM attacks based on purely in-band negotiation ineffective.

Advanced Device Authentication: New proposals foresee a hardware-based identity layer similar to a distributed PKI or blockchain-based device registry in which every Bluetooth radio possesses a verifiable credential. Although such schemes provide strong protection against

impersonation, scaling a worldwide PKI to billions of devices is a major logistical and trust-management problem.

Physical-Layer Defenses: Physical-layer security measures are becoming popular as an add-on to cryptographic security. RF fingerprinting recognizing devices through subtle differences in their transmission hardware and watching for frequency-hopping anomalies can catch cloned or rogue devices even when they perfectly match a valid BD_ADDR. While still at early research stages, these methods may become part of enterprise-grade or defense-focused deployments.

Security-First Protocol Extensions: Bluetooth Mesh, released in 2017, demonstrates a vision-forward design constructed on top of BLE but with its own two-level encryption (network and application keys) and required authentication. Its robust security stance is indicative of lessons from past BLE vulnerabilities and implies that subsequent Bluetooth profiles and extensions will integrate security from the ground up and not as an add-on.

Supporting the Long-Tail of Legacy Devices: An ongoing challenge is the enormous number of legacy or unpatched devices particularly in IoT, industrial, and medical environments where firmware updates are not possible. Until these devices are eventually retired, they continue to be vulnerable to KNOB, BIAS, SweynTooth, and other historical vulnerabilities. Network-level countermeasures, like VLAN segmentation, Bluetooth-over-TLS gateways, or purpose-built wireless intrusion detection systems, are necessary stop-gaps to quarantine risk in scenarios where complete device replacement is infeasible.

In short, the Bluetooth ecosystem is moving towards a layered, defense-in-depth architecture: unified cryptographic primitives, strict privilege separation, OOB pairing, hardware-anchored identities, and physical-layer anomaly detection. The challenge will be continuing to balance interoperability and usability with strong protections especially as new use cases (automotive, medical, mesh networking) require both seamless connectivity and strong security guarantees.

5. Conclusion

In short, the years 2020–2025 were a watershed in the security of Bluetooth. A sequence of specification-level attacks (KNOB, BIAS, BLURtooth) and widespread implementation flaws (SweynTooth, BrakTooth, BlueFrag) highlighted underlying vulnerabilities in negotiation key establishment, authentication, and protocol interpretation in both Classic and BLE transports. Due to the prevalence of Bluetooth, the potential consequences varied from passive eavesdropping to complete remote compromise. The swift, synchronized reaction from Bluetooth SIG errata and new Core Specification mandates (v5.2–5.5) to chipset and platform vendor firmware and OS patches illustrates how the ecosystem will respond when vulnerabilities are laid bare. Important lessons learned are the disproportionate security impacts of protocol design decisions (e.g., negotiable key lengths, non-mutual authentication), the dangers of backward-compatibility downgrade attacks, and the need for extensive fuzz testing against malformed inputs.

Architecturally, Bluetooth is trending towards harder defaults (mandated Secure Connections, minimum-length keys, OOB pairing), more stringent privilege separation, and tougher qualification testing. However, human factors—user consent paradigms and pairing UX—are still essential. Upcoming efforts will probably be focused on more advanced intrusion-detection mechanisms, hardware-anchored identities, and further slimming down legacy features (maybe in a Bluetooth 6 release). In the end, Bluetooth's development in these five years reinforces a larger lesson: strong wireless security requires ongoing protocol hardening, persistent implementation auditing, and an unshakeable commitment to prompt patch deployment. As Classic and BLE support an ever-growing list of applications, vigilant persistence and proactive defenses will be critical to protecting our wireless world.

References

Administrador. (2021, December 14). *Attacks to the Bluetooth Link Manager Protocol with BrakTooth*. Tarlogic Security. https://www.tarlogic.com/blog/attacks-bluetooth-link-manager-braktooth/

Antonioli, D., Tippenhauer, N. O., & Rasmussen, K. (2020). BIAS: Bluetooth Impersonation AttackS. *2020 IEEE Symposium on Security and Privacy (SP)*, 549–562. https://doi.org/10.1109/SP40000.2020.00093

Barua, A., Al Alamin, M. A., Hossain, Md. S., & Hossain, E. (2022). Security and Privacy Threats for Bluetooth Low Energy in IoT and Wearable Devices: A Comprehensive Survey. *IEEE Open Journal of the Communications Society*, *3*, 251–281. https://doi.org/10.1109/OJCOMS.2022.3149732

CERT/CC Vulnerability Note VU#589825. (n.d.). Retrieved July 31, 2025, from https://www.kb.cert.org

CVE-2020-0022: In reassemble_and_dispatch of packet_fragmenter.cc, there is possible out of bou. (n.d.). Retrieved July 31, 2025, from https://www.cvedetails.com/cve/CVE-2020-0022/

CVE-2023–45866: A Critical Bluetooth Security Flaw—EXPLOIT | by Hacker's Dump | Medium. (n.d.). Retrieved July 31, 2025, from https://medium.com/@hackersdump0/cve-2023-45866-a-critical-bluetooth-security-flaw-exploit-d2e0aec149fc

Fatima Yousaf, Ammar Ahmad Khan, & Muhammad Arslan. (2024). Machine Learning-Based Detection of Mirai and Bashlite Botnets in IoT Networks. *JCBI*, *07*(01). https://doi.org/10.56979/701/2024

Ghori, M. R., Wan, T.-C., & Sodhy, G. C. (2020). Bluetooth Low Energy Mesh Networks: Survey of Communication and Security Protocols. *Sensors*, *20*(12), 3590. https://doi.org/10.3390/s20123590

Laghari, S. U. A., Li, W., Manickam, S., Nanda, P., Al-Ani, A. K., & Karuppayah, S. (2024). Securing MQTT Ecosystem: Exploring Vulnerabilities, Mitigations, and Future Trajectories. *IEEE Access*, *12*, 139273–139289. https://doi.org/10.1109/ACCESS.2024.3412030

New "BLESA" Bluetooth Vulnerability Could Affect Billions of IoT Devices, Researchers Warn. (n.d.). Retrieved July 31, 2025, from https://www.bitdefender.com/en-us/blog/hotforsecurity/new-blesa-bluetooth-vulnerability-affect-billions-iot-devices-researchers-warn

Prakash, Y. W., Biradar, V., Vincent, S., Martin, M., & Jadhav, A. (2017). Smart bluetooth low energy security system. *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2141–2146. https://doi.org/10.1109/WiSPNET.2017.8300139

SweynTooth Vulnerabilities | CISA. (2020, March 4). https://www.cisa.gov/news-events/ics-alerts/ics-alert-20-063-01

Tech Xplore—Technology and Engineering news. (n.d.). Retrieved July 31, 2025, from https://techxplore.com/

Zhang, Y., & Lin, Z. (2022). When Good Becomes Evil: Tracking Bluetooth Low Energy Devices via Allowlist-based Side Channel and Its Countermeasure. *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 3181–3194. https://doi.org/10.1145/3548606.3559372

Aamir, N., Raza, A., Iqbal, M. W., Hamid, K., Nazir, Z., Asif, A., Hussain, S., & Muhammad, H. (2025). Topic Modeling Empowered by a Deep Learning Framework Integrating BERTopic, XLM-R, and GPT. *Journal of Computing & Biomedical Informatics*, *08*, 1–18. https://doi.org/10.56979/802/2025

Bhatti, S., Hamid, K., Bashir, A., zafar, zishan, raza, ahmad, & Iqbal, M. waseem. (2023). *SOLUTIONS, COUNTERMEASURES, AND MITIGATION METHODS FOR THE RISE OF AUTOMOTIVE HACKING*. *56*, 77–99. https://doi.org/10.17605/OSF.IO/UG6VD

Danish, M., Shahid, S., Ghafar, A., Hamid, K., Ali, N., Ghani, A., Ibrar, M., & Mandan, S. (2025). *Security of Next-Generation Networks: A Hybrid Approach Using ML-Algorithm and Game Theory with SDWSN*. *3*, 18–36. https://doi.org/10.63075/wdpwrr31

Hamid, K., Danish, M., Asif, A., Khan, Y., Iqbal, M. W., Ali, U., & Ibrar, M. (2025). Empowering Robust Security Measures in Node.js-Based REST APIs by JWT Tokens and Password Hashing: Safeguarding Cyber World. *Annual Methodological Archive Research Review*, *3*. https://doi.org/10.63075/w2nam443

Hamid, K., Iqbal, M. W., Aqeel, M., Liu, X., & Arif, M. (2023). Analysis of Techniques for Detection and Removal of Zero-Day Attacks (ZDA). In G. Wang, K.-K. R. Choo, J. Wu, & E. Damiani (Eds.), *Ubiquitous Security* (pp. 248–262). Springer Nature. https://doi.org/10.1007/978-981-99-0272-9_17

Hamid, K., Iqbal, M. W., Aqeel, M., Rana, T. A., & Arif, M. (2023). Cyber Security: Analysis for Detection and Removal of Zero-Day Attacks (ZDA). In *Artificial Intelligence & Blockchain in Cyber Physical Systems*. CRC Press.

Ibrar, M., Riaz, S., Khan, Y., Asif, A., Hamid, K., Iqbal, M. W., & Asim, M. (2024). Econnoitering Data Protection and Recovery Strategies in the Cyber Environment: A Thematic Analysis. *International Journal for Electronic Crime Investigation*, *8*. https://doi.org/10.54692/ijeci.2024.0804216

Khaliq, K., Rahim, N., Hamid, K., Ibrar, M., Ahmad, U., & Ullah, M. (2024). *Ransomware Attacks: Tools and Techniques for Detection* (p. 5). https://doi.org/10.1109/ICCR61006.2024.10532926

Malik, N., Delshadi, A., Ibrar, M., Hamid, K., Aamir, M., Ahmed, F., & Ahmad, G. (2024). *Behavior and Characteristics of Ransomware—A Survey* (p. 05). https://doi.org/10.1109/ICCR61006.2024.10532983

Raza, A., Hussain, N., Hamid, K., Liaqat, H., Mujahid, M., Gul, S., & Iqbal, M. W. (2025). *Automatic Abstractive Summarization of Text: Harnessing the Power of Large Language Models and Deep Learning Article Details ABSTRACT*. *3*, 26–43. https://doi.org/10.63075/82cejv76

Riaz, S., Asif, A., Khan, Y., Ibrar, M., Afzal, S., Hamid, K., Gul, S., & Iqbal, M. W. (2025). Software Development Empowered and Secured by Integrating A DevSecOps Design. *Journal of Computing & Biomedical Informatics*, 02. https://doi.org/10.56979/802/2025

Tehreem, U., Iqbal, M. W., Hamid, K., & Saeed, M. M. (2025). Analysis and Development of Kids' Cell Activities Monitoring App. In M. Arif, A. Jaffar, & O. Geman (Eds.), *Computing and Emerging Technologies* (pp. 405–420). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-77617-5_32

zafar, zishan, Hamid, K., Kafayat, M., Iqbal, M. waseem, Nazir, Z., & Ghani, A. (2023). AI-Based Cryptographical Framework Empowered Network Security. *Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)*, *42*, 497–510. https://doi.org/10.17605/OSF.IO/W69VT

Zia, S., Iqbal, M. W., Noor, M., Aqeel, M., & Hamid, K. (2025). User Experience (UX) Enrichment Through Digital Branding. In M. Arif, A. Jaffar, & O. Geman (Eds.), *Computing and Emerging Technologies* (pp. 37–50). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-77617-5_4