Bluetooth LE Spam with ESP32 running Marauder and Bruce

Blagoj Nenovski[1]

[1]*University St Kliment Ohridski, 1Maj nn, 7000 Bitola, R. North Macedonia*

blagoj.nenovski@uklo.edu.mk

Abstract:
Bluetooth Low Energy (BLE) advertising is a fundamental mechanism that enables fast device discovery and connection. However, this same mechanism can be exploited for spam attacks that can overwhelm users with repeated pairing prompts, confuse them, or cause denial of service. This paper evaluates BLE spam using two ESP32-based Cheap Yellow Display (CYD) devices running two open-source penetration testing firmware images: Marauder and Bruce. These firmware images include multiple BLE advertising attacks such as AppleJuice, SourApple, Samsung Spam, Google Fast Pair and Microsoft Swift Pair. The tests were conducted in a controlled environment with black box experiments testing against iOS, Android/Samsung and Windows devices running different operating systems and software versions. Results show that Apple devices running iOS 26 do not crash under BLE spam but still display persistent pairing prompts when Bluetooth is enabled before or during the attack. Modern Samsung devices largely ignore or show only a single prompt, while older models remain vulnerable to persistent spam. Windows 11 devices are consistently susceptible to Swift Pair spam when notifications are enabled and Windows 10 behavior depends on the configuration and patch status. Detection experiments highlight Android smartphones with suitable scanning apps as the most practical means for detecting active BLE spam sources.

Keywords:
BLE spam, ESP32, CYD, Marauder, Bruce

1. Introduction

ESP32 is a low-cost, low-power system-on-chip with integrated Wi-Fi and Bluetooth [1]. It is powered by a dual core CPU and has suitable RAM capacity for IoT usage. Having a peripheral set with GPIO, SPI, I²C, UART, ADC/DAC and PWM makes it flexible for various sensors, radios and human interfaces. It supports multiple programming ecosystems starting with the native ESP-IDF, then Arduino core, MicroPython etc. One of the key advantages of using ESP32 is the strong maker community coupled with the industry ecosystem with various modules, shields, libraries and code examples. Its low cost and strong community support make it widely used in IoT, wearables, DIY radios and embedded devices. ESP32-Cheap-Yellow-Display [2] or, for short, CYD is an ESP32 with a built-in display with a touch screen. The very first mainstream device was ESP32-2432S028R which the community addressed as Cheap Yellow Display or CYD. The device integrates an ESP32 with Wi-Fi and Bluetooth, and features a 2.8-inch 320×240 LCD display with resistive touch screen, USB for power and programming, SD card slot, LED and additional pins. It is available for around $15 on sites like AliExpress [3]. Besides the original CYD there are variations of the product with different screen sizes, including capacitive touch and dual-USB variants.

There are many legitimate use cases for ESP32 devices such as IoT device debugging, beacon and location systems applications [4], microcontroller projects with a custom UI, as a handheld field tool for wireless troubleshooting and custom projects, but CYD can also run penetration testing / security firmware images like: Marauder [5] and Bruce [6]. Both include a category for BLE spam attacks which are the focus of this paper. BLE spam can cause user confusion, panic and frustration to the end user. This can result in increased likelihood of misclicks, lowered trust in legitimate prompts and also battery drain. Most notable examples in this category are the multiple iOS attacks that caused denial of service by using crafted Bluetooth packets [7].

This paper utilized two CYD (one running Marauder and one Bruce) and black-box manual observation testing. CYD devices were used to test BLE spam attacks against multiple devices, each running different operating systems and versions. This approach allows assessing the real world impact to determine how devices behave in practice. It creates the initial picture of which devices show no response, which show brief prompts and which are susceptible to persistent spam. The focus is on the user experience level and how the spam can be perceived by an ordinary user. Even though both Marauder and Bruce are available as open source, the code itself cannot confirm how successful the attacks against various devices are.

2. Previous work

There is work that conceptualizes, designs, and evaluates a novel framework to defend BLE peripherals against low level BLE attacks [8]. There is also work that addresses the vulnerability of the Just Works pairing method in BLE through a case study involving a smart light bulb where the authors intercept BLE exchanges between two entities and propose an algorithm to enhance the security of the Just Works pairing mechanism [9]. ESP32 for Wi-Fi security is addressed by Tiavri et al. with developing a comprehensive ESP32-based Wi-Fi penetration testing tool that incorporates various attack methodologies including de-authentication, WPA/WPA2 handshake capture, PMKID extraction and rogue access point creation [10]. There is more work available for Wi-Fi security with ESP32 such as a work that presents a simulation of an attack scenario on the commoditized ESP32 utilized for drones during their OTA update process where the authors demonstrate Wi-Fi cracking, ARP spoofing, TCP SYN flooding techniques and postponing the OTA update procedure on an ESP32 Drone [11]. Another work proposes an application layer security model [12] for BLE devices that was implemented and tested on ESP32 with Ubertooth One sniffing that ensures authenticity, confidentiality, integrity and non-repudiation against eavesdropping, impersonation and replay attacks. Closest to this paper comes the work where they reverse engineer the hardware and software components dedicated to Bluetooth Low Energy (BLE) on the ESP32 and ANT protocol on Nordic Semiconductors nRF chips and then implement multiple attacks on the repurposed ESP32 by targeting various wireless protocols including ones not natively supported by the chip [13].

3. Methodology

Using a web flashing tool, the first CYD was flashed with Marauder [14] and the second Bruce [15]. These CYD devices were used to perform tests with controlled BLE advertisement packets. When running the tests the two CYD devices were powered by a portable battery and the targets were positioned at a distance of 1–1.5 m. The tests began with the Marauder CYD, followed by tests using the CYD running the Bruce firmware. The target devices were in a default state with Bluetooth turned on and with no prior pairing.

Each CYD kept the configured firmware advertising routines for the five BLE spamming modes: SourApple, AppleJuice, Samsung, Google/Android and Windows/Swift Pair. The key firmware behaviors used during tests such as the MAC randomization per burst, name randomization for Swift Pair, TX power and advertising bursts were preserved during all runs. Since the Marauder firmware does not display spam iterations, the tests were run for five minutes for each of the BLE spam attacks offered. For Bruce each of the tests was run for at least 300 iterations for each of the available BLE spam attacks. Each test was repeated three times to account for randomness.

Target devices included six Android devices, two iPhones and two Windows laptops running Windows 10 and Windows 11. Manual observation tests measured whether the target device detected the spoofed advertisement and prompted a pairing UI.

This paper and work was conducted ethically and only in a highly controlled environment. All target devices were owned by the researcher or used with explicit owner consent. The tests done included no public spaces and no uninvolved third party devices were exposed. Experiments were limited in duration to avoid prolonged interference and stopped immediately once determining the test results.

4. Results and discussion

Apple devices

The most known BLE spam attack is the one where a Flipper Zero was used to perform DoS on iOS devices [16]. The security update that came with iOS 17.2 eliminated the denial of service aspect. Devices running the latest iOS version 26 were tested against BLE spam attacks included in Marauder and Bruce in order to determine how iPhones react to the SourApple attack. The SourApple attack is included in both firmware images while the Bruce firmware has an additional attack called AppleJuice.

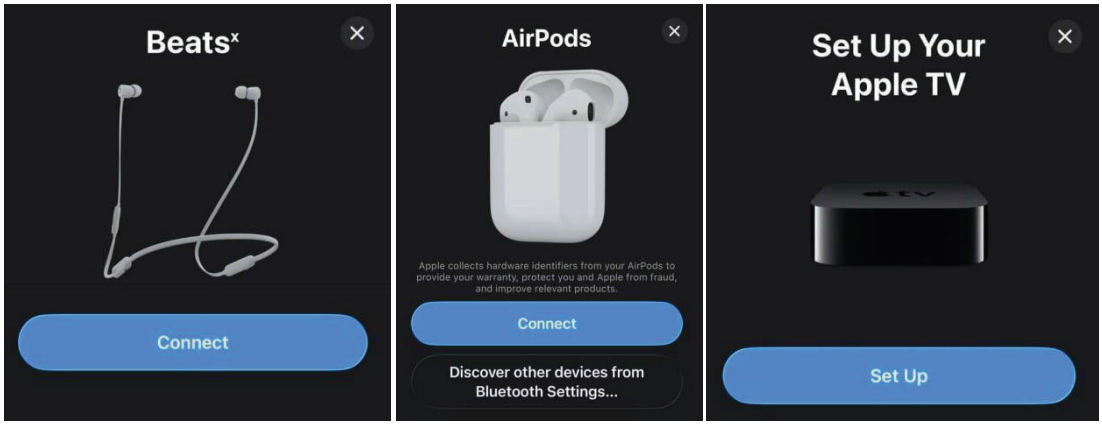

Figure 1: Different connection prompts on iPhone

The black-box testing conducted on two iPhones (13 and 16 Pro) showed that even with the latest iOS 26, iPhones prompted for connection when exposed to the BLE advertisements. The iPhones did not crash but there was different behavior depending on when Bluetooth was turned on. When idle with Bluetooth on, both iPhones responded to very few BLE advertisements. When activating Bluetooth close to or at the time of the attack both iPhones displayed persistent prompts for connection.

Android / Samsung devices

For testing the BLE spam attacks against Android, various Samsung devices with different versions of Android, OneUI and Android security patch versions were used. These Samsung devices were tested with both the Samsung BLE and Android/Google BLE spam attacks.

Table 1:
Samsung devices tested

Device	Release year	Android	OneUI	Android security patch	Marauder Samsung BLE Spam	Marauder Google BLE spam	Bruce Samsung Spam	Bruce Android Spam
Z Fold 6	2024	16	8.0	Sep. 2025	U	U	U	S
Tab A9+	2023	15	7.0	Jul. 2025	U	U	U	S
A13	2022	14	6.1	Jan. 2025	U	U	U	S
A52s	2021	14	6.1	Jun. 2025	U	U	U	S
A71	2020	13	5.1	Feb. 2024	P	U	S	S
J4+	2018	9	1.0	Dec. 2020	P	U	U	U

Table 1 lists six Samsung (Android) devices that vary by model, Android version, OneUI version and Android security patch. From the results we can see that the devices Z Fold 6, Tab A9+, A13 and A52s were unaffected by Bruce's Samsung Spam and Marauder's Google BLE spam and Samsung

BLE spam. Each of these devices showed a single brief prompt when exposed to Bruce's Android spam. These prompts lacked persistency so they cannot be classified as spam. A71 displayed brief prompts under both of the Bruce's attacks and was unaffected by Marauder's Google BLE Spam. A71 displayed persistent prompts when exposed to Marauder's Samsung BLE Spam both on the home screen as well as when apps were opened. J4+ was unaffected by all attacks, other than Marauder's Samsung BLE spam. In this case J4+ showed persistent prompts on the home screen, but it differed from A71 in that when apps were opened, instead of connections prompts it displayed notifications. The notifications were more rapid compared to the home screen prompts. Different connection prompts for J4+, A71 and Z Fold 6 are shown with Figure 2.

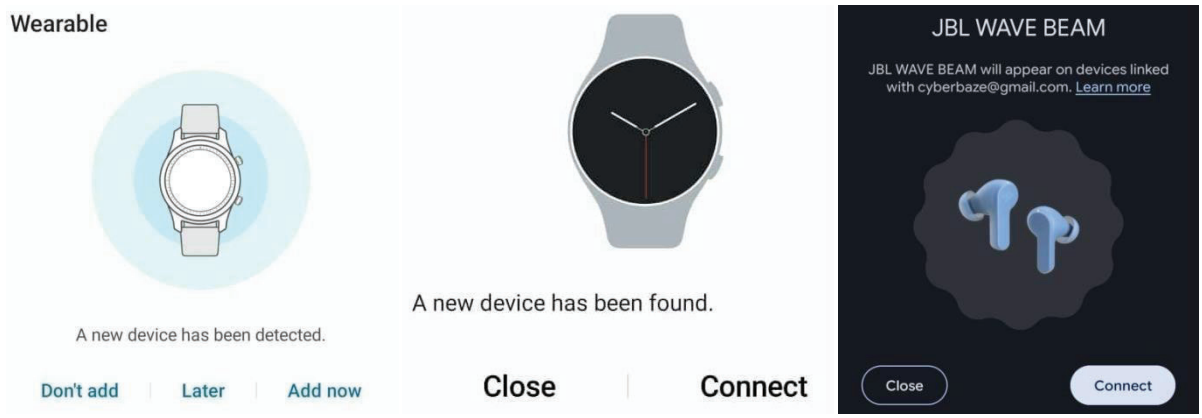

Figure 2: Connection prompt for: J4+; A71 and brief prompt for Z Fold 6

The results in Table 1 show that modern Samsung devices largely ignore BLE spam, or they show only a single brief prompt, while older models that are no longer supported such as A71 and J4+ are susceptible to persistent prompts when exposed to continuous advertising.

Windows Swift Pair

Windows Swift Pair is a feature in Windows 11 and Windows 10 that enables users to quickly pair supported Bluetooth devices with their computer through a notification-based workflow, thereby eliminating the need to manually navigate the Settings menu and search for devices.

Swift pair was first released with Windows 10 version 1803 and it was not turned on automatically for users [17]. Microsoft stated that the decision was made when they learned continuously monitoring Bluetooth Low Energy advertisements caused some radios to improperly handle Wi-Fi activity when on the same radio.

Bruce's Windows Spam and Marauder's SwiftPair Spam were tested against two Windows laptops, one running Windows 11 Pro version 24H2 (OS build 26100.6584) and another running Windows 10 Home version 22H2 (OS build 19045.6332).

In the case of Windows 11, both Marauder and Bruce BLE spam initiated persistent notification prompts to connect to a new device. The prompts displayed random names with 1-10 characters.

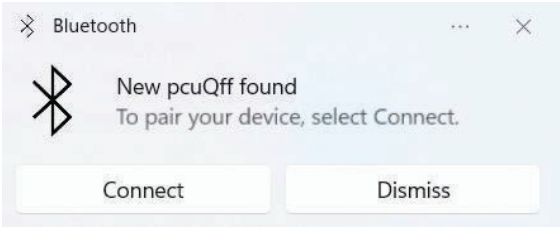

Figure 3: Windows 11 pairing notification

In contrast, the Windows 10 device was not showing notifications. Checking the Bluetooth settings revealed that "Show notifications to connect using Swift Pair" was disabled. Enabling this setting resulted in the same type of prompts as with the Windows 11 laptop.

The default setting for “Show notifications to connect using Swift Pair” plays a key role in determining whether a Windows device is vulnerable. This setting can vary starting with the OEM configuration, the version of the operating system as well as the hardware and the driver’s capability. In enterprise environments, company policies may override these factors.

There is also the factor of users changing the setting for showing notifications to connect using Swift Pair. With Windows 11 this can be changed in Settings > Bluetooth & devices > Devices under the toggle “Show notifications to connect using Swift Pair” and in Windows 10 Swift Pair can be enabled or disabled in Settings > Bluetooth & other devices using the checkbox “Show notifications to connect using Swift Pair”.

Patching Windows can be challenging due to the very nature of the operating system, as it supports a fragmented ecosystem with different Bluetooth radios, drivers and OEM configurations. Patching may also require driver updates which are delivered by third parties. Supporting various components also puts pressure on Microsoft for backwards compatibility by preserving legacy Bluetooth behaviors. Patching must be supported by user behavior for updating their operating system. Some users may continue to use Windows 10 even after end of support by October 2025. User behavior can delay or even disable Windows updates, making the user’s computer more vulnerable. In the enterprise there can be cases where companies may delay applying patches due to compatibility. All of this limits how aggressive a Windows patch can be when compared to simpler ecosystems like those with Apple and Samsung.

Detecting a BLE spam attack

BLE spam attacks flood the environment with continuous fake advertisements, making the identification and detection of the device a challenging process. Detection requires both hardware and software that can capture and display BLE signals. Three different detection approaches were evaluated: using a laptop, a CYD device and an Android smartphone, each offering different levels of usability and accuracy. The main challenge is that BLE spam packets are sent in short bursts with different identifiers. The best detection approach should balance portability, scanning speed and the ability to visualize BLE advertisements based on the signal strength (RSSI).

Laptop

Detecting a BLE spam attack with a laptop can seem like the preferred option mainly due to its processing power and the versatility of the operating system and application software but it has its own challenges. Most laptops have low sensitivity integrated Bluetooth radios that are designed for short range connections and not for continuous scanning. Detection may require an external Bluetooth sniffer or dongle such as Ubertooth One, Nordic nRF52840 or an adapter that is capable of capturing HCI. Even when setup with the appropriate software, roaming with a laptop to detect a BLE spamming device is unintuitive and impractical due to the form factor, dimensions and the weight of the laptop.

CYD device

Using the same CYD devices for detecting a BLE spam device is also not the best option. Marauder firmware includes a “Bluetooth sniffer” that displays the RSSI and the MAC address of nearby devices while Bruce offers BLE scan with only the MAC address. Both of these functionalities can list devices periodically and are not intended for high speed monitoring due to the low scan intervals and the display list intervals compared to the burst of spam advertisements. This approach would also require the person that detects to have a CYD or similar ESP32 device.

Android smartphone

In the case of a BLE spam attack, using one or more Android smartphones can prove to be the most effective approach. Most modern Android smartphones have BLE radios with scanning capabilities already built in, thus it eliminates the need for additional hardware. There are multiple apps available on the Play Store that allow BLE scanning and provide a visual representation of BLE advertisements based on their RSSI. This approach also allows for crowd detection where multiple people can collectively scan and detect the presence and locate a BLE spamming device.

To detect a BLE spam attack the app needs to be able to process and display the most recent BLE advertisements in a way that allows the end user to intuitively detect the position of the spamming device. One app is Bluetooth Finder & BLE Scanner [18] which is good for determining the distance between the user and a device that is continuously advertising via BLE but not suitable for multiple

short bursts of advertisements. There are apps that sort by signal strength such as with BLE Hero [19] and UFind: BLE & Bluetooth Tracker [20] but these apps position the past advertisements on the top due to the signal strength which can turn out to be overwhelming when detecting a spamming device. Then there is nRF Connect for Mobile [21] that has the option to show RSSI graph which in theory would allow the user to determine the distance of the device emitting the BLE advertisements, while in practice the BLE spamming is not displayed on the graph due to the multiple short bursts of advertisement.

BLE Scanner (Connect & Notify) [22] is the most suitable Android app for detecting a BLE spamming device. This app has a "Proximity" feature that places devices on "radar" visualizing the devices based on the proximity, as determined by the signal strength of the advertisements. In this mode devices are displayed in the form of a circle with the first two characters of their names (or N/A when names are not available) in the first row and the signal strength in the second row. The signal strength determines the position of devices/advertisements in three ranges: Far, Near and Immediate. A user can move within an area until large numbers of devices are placed in the Immediate category and by which the user can locate the BLE spamming device.

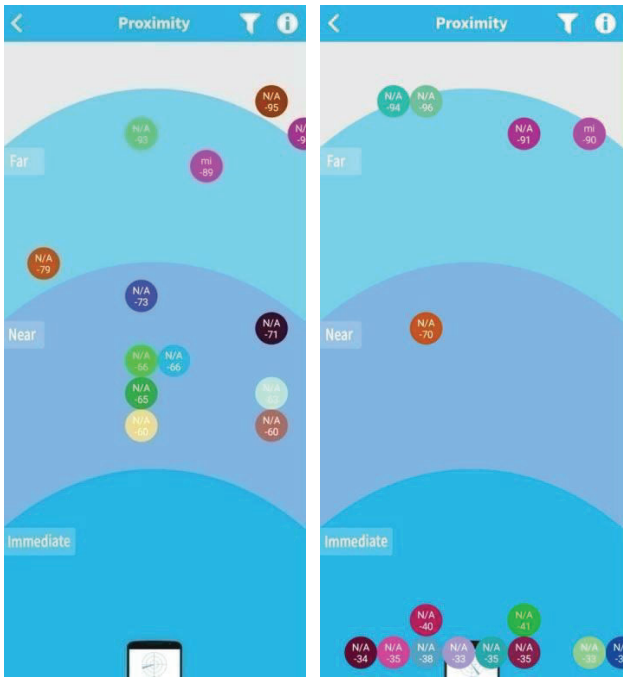

Figure 4: Left: BLE advertisements, right: BLE advertisements in immediate distance

5. Conclusions and future work

ESP32 devices like the CYD are inexpensive, widely available and can be repurposed for BLE penetration testing with firmware images like Marauder and Bruce. These small devices make it simple to generate realistic BLE advertising that can make users overwhelmed or confused and even cause denial of service in the past. Controlled black box tests showed that iPhones running the latest version iOS 26 do not crash under BLE spam advertising but still display persistent connection prompts especially when Bluetooth is actively turned on before or during the attack. Modern Samsung devices mostly ignore or show only brief non-persistent prompts. Older Samsung and other Android devices remain the most susceptible to persistent prompts and repeated notifications such as the case with Samsung A71 and Samsung J4+. The setting for showing notifications to pair with Swift Pair determines whether Windows devices show persistent prompts.

In 2025, BLE spam is primarily a usability and trust issue rather than an immediate code execution exploit in mainstream operating systems. The real cost is user confusion, panic, accidental interactions and potential battery drain.

Analysis of different detection approaches shows that using one or more Android smartphones is the most effective way to locate a BLE spamming device.

As a recommendation, users should turn off Bluetooth when not needed and keep their devices regularly updated. Software updates and their frequency are of key importance when countering these attacks. Looking at the update support, iOS 26 is compatible with previous phones starting with iPhone 11 [23], a device that was launched in 2019. Samsung has update cycle for devices across three categories: models with monthly security updates, quarterly security updates and biannual security updates [24]. Real-world tests showed that devices with discontinued software updates performed the worst. Microsoft's support for Windows 10 ends October 2025 [25].

This work used black-box manual observation with two CYD devices and a limited set of target devices. The results reflect the tested devices and the operating system combinations. Different hardware radios, OEM firmware or regional software builds may behave differently. The manual logging approach limits repeatability and packet level attribution.

Future work can include in-depth code analysis with a white-box approach and study both Marauder and Bruce codebases. This can map how each BLE spam attack is constructed at the packet level. Future work can also do comparative analysis of the same BLE attacks available in the Marauder and Bruce firmware images such as SourApple, Samsung BLE spam and SwiftPair, to identify the differences in packet structure, randomness and effectiveness.

Additional research can be done on automated testing framework which could automate the spam transmission rather than relying on manual logging. Future work can also develop individual detection mechanisms such as the notification system in Windows, where once detected could provide the end user with an option to silence that specific type of notification on a user-determined period.

References:

[1] Espressif, "ESP32 Series Datasheet Including," 2024. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (accessed Jul. 14, 2025).

[2] witnessmenow, "GitHub - witnessmenow/ESP32-Cheap-Yellow-Display: Building a community around a cheap ESP32 Display with a touch screen," GitHub, 2023. https://github.com/witnessmenow/ESP32-Cheap-Yellow-Display (accessed Jul. 12, 2025).

[3] "ESP32 MCU 2.8 Inch Smart Display for Arduino LVGL WIFI Bluetooth Touch WROOM 240*320 Screen LCD TFT Module with Free Tutorials - AliExpress," aliexpress., 2025. https://www.aliexpress.com/item/1005004961285750.html (accessed Jul. 14, 2025).

[4] S. Sophia, B. M. Shankar, K. Akshya, A. C. Arunachalam, V. T. Y. Avanthika and S. Deepak, "Bluetooth Low Energy based Indoor Positioning System using ESP32," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2021, pp. 1698-1702, doi: 10.1109/ICIRCA51532.2021.9544975.

[5] justcallmekoko, "GitHub - justcallmekoko/ESP32Marauder: A suite of WiFi/Bluetooth offensive and defensive tools for the ESP32," GitHub, https://github.com/justcallmekoko/ESP32Marauder (accessed Jul. 16, 2025).

[6] pr3y, "GitHub - pr3y/Bruce: Predatory ESP32 Firmware," GitHub, https://github.com/pr3y/Bruce (accessed Jul. 16, 2025).

[7] "CVE Website," Cve.org, 2024. https://www.cve.org/CVERecord?id=CVE-2023-42941

[8] G. Benita, L. Sestrem, M. E. Garbelini, S. Chattopadhyay, S. Sun and E. Kurniawan, "VaktBLE: A Benevolent Man-in-the-Middle Bridge to Guard against Malevolent BLE Connections," 2024 Annual Computer Security Applications Conference (ACSAC), Honolulu, HI, USA, 2024, pp. 621-635, doi: 10.1109/ACSAC63791.2024.00059.

[9] M. Y. Sadaoui, O. Salem and A. Mehaoua, "Security Assessment of Bluetooth Just Works Pairing Method: Vulnerabilities and Enhancements," 2023 IEEE International Conference on E-health Networking, Application & Services (Healthcom), Chongqing, China, 2023, pp. 107-112, doi: 10.1109/Healthcom56612.2023.10472370.

[10] H. Tiwari, A. Tomar, S. Patil, S. Patil, J. Gangane and S. Kate, "Slipper Zero: Exploring Wi-Fi Security Vulnerabilities and Attack Implementations on ESP32 Microcontrollers," 2024 Global Conference on Wireless and Optical Technologies (GCWOT), Malaga, Spain, 2024, pp. 1-7, doi: 10.1109/GCWOT63882.2024.10805625.

[11] J. Baek, J. Jang, and S. Kim, “A Study on Vulnerability Analysis and Memory Forensics of ESP32,” Journal of Internet Computing and Services, vol. 25, no. 3, pp. 1–8, Jun. 2024.
[12] F. de Vito, Í. A. de Sousa Tacca, G. P. Aquino and E. C. V. Boas, "A Novel Secure Communication Scheme for Bluetooth Low Energy Devices," 2024 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS), Bali, Indonesia, 2024, pp. 137-143, doi: 10.1109/IoTaIS64014.2024.10799446.
[13] R. Cayre, D. Cauquil and A. Francillon, "ESPwn32: Hacking with ESP32 System-on-Chips," 2023 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 2023, pp. 311-325, doi: 10.1109/SPW59333.2023.00033.
[14] “CYM WebFlasher,” Github.io, https://fr4nkfletcher.github.io/Adafruit_WebSerial_ESPTool/ (accessed Jul. 22, 2025).
[15] “Bruce Firmware,” Bruce.computer, https://bruce.computer/flasher (accessed Jul. 14, 2025).
[16] C. Reynolds, “Crashing iPhones with Flipper Zero and The Story Behind CVE-2023-42941,”, Jan. 20, 2024. https://ecto-1a.github.io/AppleJuice_CVE/ (accessed Jul. 16, 2025).
[17] “Swift Pair,” Microsoft.com, Jun. 10, 2022. https://learn.microsoft.com/en-us/windows-hardware/design/component-guidelines/bluetooth-swift-pair (accessed Jul. 16, 2025).
[18] G. studio apps, “Bluetooth Finder & BLE Scanner,” Google.com, 2021. https://play.google.com/store/apps/details?id=com.kraph.bledevice (accessed Jul. 26, 2025).
[19] onceLabs, “BLE Hero,” https://play.google.com/store/apps/details?id=com.oncelabs.blehero (accessed Jul. 26, 2025).
[20] DreamTeam Mobile, “UFind: BLE & Bluetooth Tracker“, https://play.google.com/store/apps/details?id=com.dreamteammobile.ufind (accessed Jul. 26, 2025).
[21] N. S. ASA, “nRF Connect for Mobile,” https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp (accessed Jul. 26, 2025).
[22] Bluepixel Technologies, “BLE Scanner (Connect & Notify)“ https://play.google.com/store/apps/details?id=com.macdom.ble.blescanner (accessed Jul. 26, 2025).
[23] “iPhone models compatible with iOS 26,” Apple Support, 2025. https://support.apple.com/en-mk/guide/iphone/iphe3fa5df43/ios (accessed Jul. 29, 2025).
[24] “Samsung Mobile Security,” Samsungmobile.com, https://security.samsungmobile.com/workScope.smsb (accessed Jul. 29, 2025).
[25] Microsoft, “Windows 10 support ends on October 14, 2025 - Microsoft Support,” Microsoft.com, 2025. https://support.microsoft.com/en-us/windows/windows-10-support-ends-on-october-14-2025-2ca8b313-1946-43d3-b55c-2b95b107f281 (accessed Jul. 29, 2025).