BRAKTOOTH: Causing Havoc on Bluetooth Link Manager

VULNERABILITY DISCLOSURE REPORT

By

Matheus E. Garbelini (SUTD)
Sudipta Chattopadhyay (SUTD)
Vaibhav Bedi (SUTD)
*Sumei Sun (I2R, A*STAR)*
*Ernest Kurniawan (I2R, A*STAR)*

Supported by

NRF NATIONAL SATELLITE OF EXCELLENCE IN TRUSTWORTHY SOFTWARE SYSTEMS
KEYSIGHT TECHNOLOGIES

August 31, 2021

BRAKTOOTH: Causing Havoc on Bluetooth Link Manager

Matheus E. Garbelini[1]; Sudipta Chattopadhyay[1]; Vaibhav Bedi[1]; Sumei Sun[2]; Ernest Kurniawan[2]
[1]*Singapore University of Technology and Design* [2]*I2R, A*STAR*

1 Introduction

Bluetooth Classic (BT) protocol is a widely used wireless protocol in laptops, handheld devices, and audio devices. BT main procedures are shown in Figure 1 for reference. In the past few years, Bluetooth has come under scrutiny due to the discovery of several critical vulnerabilities. In this report, we disclose BRAKTOOTH, a family of new security vulnerabilities in commercial BT stacks that range from **denial of service (DoS)** via firmware **crashes** and **deadlocks** in commodity hardware to **arbitrary code execution (ACE)** in certain IoTs. As of today, we have evaluated 13 BT devices from 11 vendors. *We have discovered a total of 16 new security vulnerabilities, with 20 common vulnerability exposures (CVEs) already assigned and four (4) vulnerabilities are pending CVE assignment from Intel and Qualcomm.*

All the vulnerabilities are already reported to the respective vendors, with several vulnerabilities already patched and the rest being in the process of replication and patching. Moreover, four of the BRAKTOOTH vulnerabilities have received bug bounty from Espressif System and Xiaomi. *An exploration on Bluetooth listing* [28] *reveals that* BRAKTOOTH *affects over 1400 product listings*. BRAKTOOTH exposes fundamental attack vectors in the closed BT stack.

As the BT stack is often shared across many products, it is highly probable that many other products (beyond the ≈1400 entries observed in Bluetooth listing) are affected by BRAKTOOTH. Therefore, we suggest vendors producing BT system-on-chips (SoCs), BT modules or BT end products to use the BRAKTOOTH proof-of-concept (PoC) code to validate their BT stack implementation. The availability of the BRAKTOOTH PoC is discussed at the end of this report.

Why BrakTooth

The code name *BrakTooth* is the combination of two words: 1) Brak and 2) Tooth. While the word *Tooth* is clearly pointing towards Bluetooth targets, the word *Brak* is Norwegian and translates to *crash* in English. The BrakTooth family of vulnerabilities affect Bluetooth enabled devices by continuously crashing or deadlocking them, while some result in more serious consequences such as arbitrary code execution.

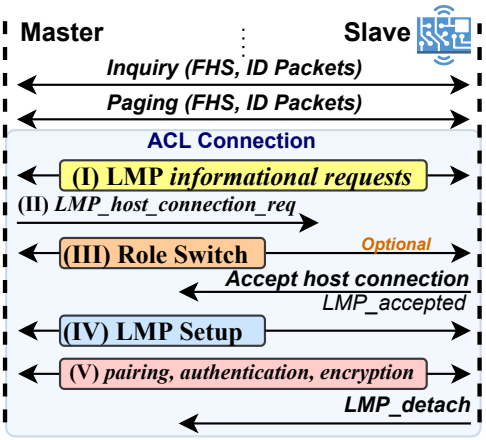

Figure 1: An Illustration of the BT connection procedure. FHS stands for *Frequency Hopping Synchronization*, ID stands for *Identity*, LMP stands for *Link Manager Protocol* and ACL stands for *Asynchronous Connection Less*.

2 Attack Scenario Overview

Figure 2 showcases the generic scenario in which BRAKTOOTH attacks are performed. The attacker only requires (1) a cheap ESP32 development kit (ESP-WROVER-KIT [31]) with a custom (non-compliant) LMP firmware and (2) a PC to run the PoC tool. The PoC tool communicates with the ESP32 board via serial port (*/dev/ttyUSB1*) and launches the attacks according to the specified target BDAddress (<*target bdaddr*>) and exploit name parameter (<*exploit_name*>).

Furthermore, the PoC tool logs over-the-air (OTA) packets and checks the health of the target by getting a paging timeout (no response) or alternatively getting status directly from the target via a serial port, ssh connection, etc.

Due to some vendors having a fixed release schedule, we will release the Proof of Concept (PoC) tool publicly only at

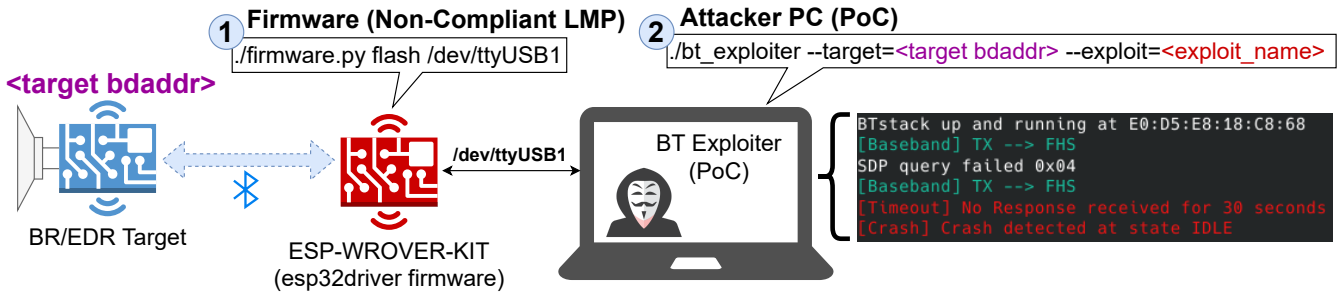

Figure 2: An Illustration of BRAKTOOTH attack scenario

Table 1: Devices used for evaluation. The sample code is provided by vendor to test the development board. This is not applicable (N.A) on products running a fixed application.

BT SoC Vendor	BT SoC	Dev. Kit / Product	Sample Code
Bluetooth 5.2			
Intel	AX200	Laptop Forge15-R	N.A
Qualcomm	WCN3990	Xioami Pocophone F1	N.A
Bluetooth 5.1			
Texas Instruments	CC2564C	CC256XCQFN-EM	SPPDMMultiDemo
Zhuhai Jieli Technology	AC6366C	AC6366C_DEMO_V1.0	app_keyboard
Bluetooth 5.0			
Cypress	CYW20735B1	CYW920735Q60EVB-01	rfcomm_serial_port
Bluetrum Technology	AB5301A	AB32VG1	Default
Zhuhai Jieli Technology	AC6925C	XY-WRBT Module	N.A
Actions Technology	ATS281X	Xiaomi MDZ-36-DB	N.A
Bluetooth 4.2			
Zhuhai Jieli Technology	AC6905X	BT Audio Receiver	N.A
Espressif Systems	ESP32	ESP-WROVER-KIT	bt_spp_acceptor
Bluetooth 4.1			
Harman International	JX25X	JBL TUNE500BT	N.A
Bluetooth 4.0			
Qualcomm	CSR 8811	Laird DVK-BT900-SA	vspspp.server.at
Bluetooth 3.0 + HS			
Silabs	WT32i	DKWT32I-A	ai-6.3.0-1149

the end of **October 2021**. This is the time when we expect most vendors to have released their patches. In the meantime, any BT semiconductor or module vendor can acquire the PoC by filling up a simple form at the BRAKTOOTH PoC website.

3 Affected BT BR/EDR chipsets

31/08/2021: As of today, vulnerabilities V6, V15 and V16 are pending CVE trackers. These vulnerabilities affect Qualcomm and Intel which are vendors that can issue CVEs by themselves (CVE Numbering Authorities - CNA). However, their default policy is to issue CVEs only after a patch has been distributed. Therefore, Table 2 will be updated accordingly once fixes for V6, V15 and V16 are available. Regarding the availability of patches, when our team had contacted Intel, we were informed that their next patch is scheduled to be released by the end of October 2021. Similarly, we are also informed by Qualcomm that their patch for V15 will take another few months to be propagated.

A summary of BRAKTOOTH appears in Table 2. In each row, we use the prefix **V** to identify a security vulnerability and **A** to indicate an anomalous behaviour (i.e., faulty target responses) that deviates from the *Core Specifications* [27]. Moreover, Table 2 outlines the respective CVEs, affected devices, protocol layers, and the violated compliance. In summary, we discovered 16 new security vulnerabilities. For all the discovered vulnerabilities, we have followed a responsible disclosure process. *Specifically, we have reached out to the affected vendors and we have provided them at least* **90 days** *until the public disclosure of this report. In most cases, we have also actively helped the vendors to reproduce the reported attacks at their end.* Several vendors (e.g. Cypress, Bluetrum, Espressif) have already produced patches with many in the process of producing them (e.g. Qualcomm, Intel). The status of patches is discussed in Section 5.

The *impact* of our discovered vulnerabilities is categorized into (I) *crashes* and (II) *deadlocks*. *Crashes* generally trigger a fatal assertion, segmentation faults due to a buffer or heap overflow within the SoC firmware. *Deadlocks*, in contrast, lead the target device to a condition in which no further BT communication is possible. This may happen due to the *paging scan* being forcibly disabled (**V16**), state machine corruption on **V6** or entirely disabling BT functionality via arbitrary code execution (ACE) on **V1**. Our results affect popular BT vendors (i.e, Intel, Qualcomm, Cypress, Texas Instruments) and relatively less known (i.e., Bluetrum, Jieli Technology, Harman), which are still employed in many consumers products such as BT speakers, keyboards, toys, etc.

V1 affects ESP32, which is used in many products ranging from consumer electronics to industrial equipment such as programmable logic controllers (PLCs). Hence, the impact is significant, as the attacker only requires knowledge of the target *BDAddress* to launch the attack. *Indeed, all the vulnerabilities* **V1-V16** *can be triggered without any previous pairing or authentication*. Moreover, the impact of **V1-V16** reaches beyond the devices listed in Table 2, since any other BT product employing an affected SoC is also vulnerable.

Multiple Link Manager Protocol (LMP) flooding attacks (e.g., **V4**, **V12**) and **V15** were detected across SoCs from different BT vendors. Since the affected vendors are majors in their fields (i.e., Intel & Qualcomm), it indicates that there is a lack of flexible tools for over-the-air testing even in 2021.

Table 2: Summary of new vulnerabilities and other anomalies found (**Vx:** Vulnerability, **Ax:** Non-compliance)

Anomalies	CVE ID(s)	Device(s)	State(s)	Target Layer(s)	Impact Type	Compliance Violated
8.1 **V1** Feature Pages Execution	CVE-2021-28139	ESP-WROVER-KIT	*Feature Exchange*	LMP	ACE / Deadlock	[V.1] Part E, Sec. 2.7
8.2 **V2** Truncated SCO Link Request	CVE-2021-34144	AC6366C_DEMO_V1.0	***After Paging***	LMP	Deadlock	[V.2] Part E, Sec. 2.7
8.3 **V3** Duplicated IOCAP	CVE-2021-28136	ESP-WROVER-KIT	*Bounding*	LMP	Crash	[V.2] Part C, Sec. 4.2.7.1
8.4 **V4** Feature Resp. Flooding	CVE-2021-28135 CVE-2021-28155 CVE-2021-31717	ESP-WROVER-KIT JBL TUNE500BT Xiaomi MDZ-36-DB	***After Paging***	LMP	Crash	[V.1] Part E, Sec. 2.7
8.5 **V5** LMP Auto Rate Overflow	CVE-2021-31609 CVE-2021-31612	DKWT32I-A BT Audio Receiver	Data Rate Change	Baseband	Crash	[V.2] Part B, Sec. 6.6.2
8.6 **V6** LMP 2-DH1 Overflow	Pending	DVK-BT900-SA	After EDR Change	Baseband	Deadlock	[V.2] Part C, Sec. 2.3
8.7 **V7** LMP DM1 Overflow	CVE-2021-34150	AB32VG1	***Many***	Baseband	Deadlock	[V.2] Part B, Sec. 6.5.4.1
8.8 **V8** Truncated LMP Accepted	CVE-2021-31613	BT Audio Receiver XY-WRBT Module	***Many***	LMP	Crash	[V.2] Part C, Sec. 5.1
8.9 **V9** Invalid Setup Complete	CVE-2021-31611	BT Audio Receiver XY-WRBT Module	*Feature Exchange*	LMP	Deadlock	[V.1] Part E, Sec. 2.7
8.10 **V10** Host Conn. Flooding	CVE-2021-31785	Xiaomi MDZ-36-DB	*Host Connection*	LMP	Deadlock	[V.1] Part E, Sec. 2.7
8.11 **V11** Same Host Connection	CVE-2021-31786	Xiaomi MDZ-36-DB	*Host Connection*	LMP	Deadlock	[V.1] Part E, Sec. 2.7
8.12 **V12** AU Rand Flooding	CVE-2021-31610 CVE-2021-34149 CVE-2021-34146 CVE-2021-34143	AB32VG1 CC256XCQFN-EM CYW920735Q60EVB AC6366C_DEMO_V1.0	***After Paging***	LMP	Crash Deadlock	[V.1] Part E, Sec. 2.7
8.13 **V13** Invalid Max Slot Type	CVE-2021-34145	CYW920735Q60EVB	*After Setup Complete*	Baseband	Crash	[V.1] Part E, Sec. 2.7
8.14 **V14** Max Slot Length Overflow	CVE-2021-34148	CYW920735Q60EVB	*After Setup Complete*	Baseband	Crash	[V.1] Part E, Sec. 2.7
8.15 **V15** Invalid Timing Accuracy	CVE-2021-34147 Pending Pending	CYW920735Q60EVB Pocophone F1 (WCN3990) Intel AX200	*Timing Accuracy*	LMP, Baseband	Crash	[V.1] Part E, Sec. 2.7
8.16 **V16** Paging Scan Deadlock	Pending	Intel AX200	*After Host Connection*	LMP, Baseband	Deadlock	[V.1] Part E, Sec. 2.7
A1 Accepts Lower LMP Length	N.A	**All tested devices**	***Many***	Baseband	Non-Compliance	[V.2] Part C, Sec. 5.1
A2 Accepts Higher LMP Length	N.A	**All, except ESP32**	***Many***	Baseband	Non-Compliance	[V.2] Part C, Sec. 5.1
A3 Multiple Encryption Start	N.A	Xiaomi MDZ-36-DB	*After Encryption Start*	LMP	Non-Compliance	[V.2] Part C, Sec. 4.12
A4 Ignore Role Switch Reject	N.A	Pocophone F1 (WCN3990)	*Role Switch*	LMP	Non-Compliance	[V.2] Part C, Sec. 4.4.2
A5 Invalid Response	N.A	Intel AX200 DVK-BT900-SA	*Feature Exchange*	LMP	Non-Compliance	[V.2] Part C, Sec. 4.3.4
A6 Ignore Encryption Stop	N.A	CYW920735Q60EVB	*After Encryption Start*	LMP	Non-Compliance	[V.2] Part C, Sec. 4.2.5.4

Besides, the *Core Specifications* only allows a limited "LMP test mode" [27] that restricts the SoC to operate with few LMP procedures.

4 Impact of BrakTooth

We created different concrete attacks leveraging the BrakTooth vulnerabilities. In the following, we show three such sample attacks that launch arbitrary code execution (ACE) or Denial of Service (DoS) on target devices.

4.1 Arbitrary Code Execution in IoTs

The most critical vulnerability (**V1** in Table 2 - 8.1) affects *ESP32* SoC [30], which is used in many Wi-Fi and Bluetooth IoT appliances such as Industry Automation, Smart Home, Fitness, etc. The attack is illustrated in Figure 3. A lack of out-of-bounds check in ESP32 *BT Library* [9] allows the reception of a *mutated LMP_feature_response_ext*. This results in the injection of eight bytes of arbitrary data outside the bounds of *Extended Feature Page Table* ("E. Features Table" in Figure 3). An attacker, which knows the firmware layout of a target device, can write a known function address (*JMP Addr.*) to the offset pointed by *Features Page* ("Feat. Page" in the *LMP_feature_response_ext* packet) field. It turns out that the *BT Library* stores some callback pointers within the out-of-bounds *Features Page* offset and such a callback is eventually invoked during the BT connection. While exploiting this vulnerability, we forced *ESP32* into erasing its NVRAM data (normally written during product manufacturing) by setting *JMP Addr.* to the address of *nvs_flash_erase*. Such erase function is always included in *ESP32* SDK [7] and therefore, it is present in any *ESP32* firmware. Similarly, disabling BT or BLE can be done via *esp_bt_controller_disable* and Wi-Fi via *disable_wifi_agc*. Additionally, general-purpose input/output (GPIO) can be controlled if the attacker knows addresses to functions controlling actuators attached to *ESP32*. As expected, this has serious implications if such an attack is applied to Bluetooth-enabled Smart Home products.

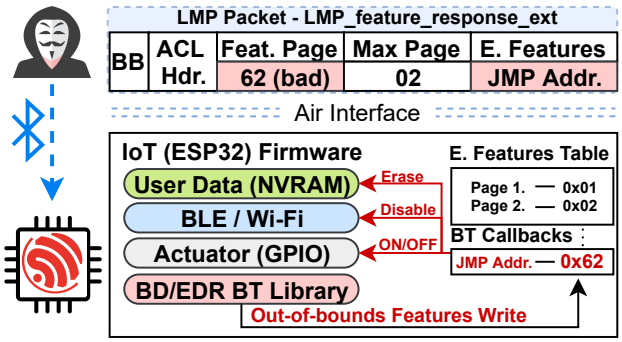

Figure 3: An Illustration of CVE-2021-28138.

4.2 DoS in Laptops & Smartphones

We discuss two sample DoS attacks discovered on laptops and smartphones employing *Intel AX200* SoCs [16] and *Qualcomm WCN3990* SoC [29]. Due to the number of smartphones and laptops vulnerable to such attacks, and the common use of BT connectivity during video conference calls and music streaming, updating the affected devices is essential.

The first DoS (**Invalid Timing Accuracy** - 8.15) is due to a failure in the SoC to free resources upon receiving an invalid *LMP_timing_accuracy_response* from a BT slave. The attacker can exhaust the SoC by (a) *paging*, (b) sending the malformed packet, and (c) disconnecting without sending *LMP_detach*. These steps are repeated with a different BT address (i.e., BDAddress) until the SoC is exhausted from accepting new connections. On exhaustion, the SoC fails to recover itself and disrupts current active connections, triggering firmware crashes sporadically. This vulnerability allows an attacker to forcibly disconnect *slave* BT devices currently connected to *AX200* under Windows or Linux Laptops. Similarly, Android phones such as Pocophone F1 and Oppo Reno 5G [21] experience BT disruptions. For example, users connected to BT Headsets experience audio to be continuously "cut" during the attack. This also results in firmware crashes and restart of Android BT Service. In all cases, the OS tries to recover connectivity which can be continuously disrupted.

The second DoS (**Paging Scan Disable** - 8.16) affects only devices using Intel *AX200* and it is triggered when an oversized *LMP_timing_accuracy_request* (>17 bytes) is sent to *AX200 slave*. This temporarily corrupts *AX200* firmware, which responds incorrectly during a subsequent BT connection and eventually disables the *paging scan* procedure (cf., Figure 1). Thus, scanning *AX200* works, but no connection is established from an external BT device. Therefore, this attack can be used to trick an user to connect to the attacker's BT hardware instead of the legitimate target since **AX200** paging scan is disabled. Indeed, the user needs to manually re-enable BT to restore functionality. The attack is also used to disconnect certain *master* BT devices connected to a vulnerable Laptop. This leads to sporadic BT firmware crashes.

4.3 Freezing Audio Products

Many vulnerabilities were discovered while testing with a BT Speaker (*Mi Portable Bluetooth Speaker - MDZ-36-DB* [35]), BT Headphone (*JBL TUNE 500BT* [17]) and BT Audio Modules (XY-WRBT [24] and an unbranded BT Audio Receiver). The discovered vulnerabilities arise from failures when sending oversized LMP packets (**LMP Auto Rate Overflow** - 8.5), truncated packets (**Truncated LMP Accepted** - 8.8), starting procedures out-of-order (**Invalid Setup Complete** - 8.9) and finally by flooding LMP packets (**Feature Response Flooding** - 8.4).

The vulnerabilities can *"freeze" Xiaomi MDZ-36-DB* and completely shut down *JBL TUNE 500BT*. This requires the user to *manually* turn on the unresponsive devices. Since both devices accept multiple BT connections, an attack can be triggered while the user is playing some music. As an exception, *XY-WRBT* and *BT Audio Receiver* accept only one connection, which avoids an attack to be launched during an active BT connection with the user. Nevertheless, different products employing the same SoC may enable multiple BT connections depending on the product requirements.

Although issues were found in SoCs targeted to Audio products, the BT Implementation can be reused in a number of SoCs destined to different BT products. Hence, our discoveries are not limited to the type of products discussed in this section, but rather to the LMP stack implementation.

4.4 Estimating the Scope of BrakTooth

In this section, we measure the potential impact of BrakTooth vulnerabilities. To this end, we estimate the number of product listings employing the affected BT chipsets. Each listing may contain one or multiple (up to hundreds) products from the same company. We get such an estimated number by querying the Bluetooth Listings Website [28]. In particular, we use the qualification IDs (QIDs) of each affected SoCs to get their corresponding products listings. The total number of potentially affected product listings is illustrated in Table 3. In summary, as of August 23, 2021, the total number of listings affected is over 1400. We note that even if the number of listings for a major vendor such as Intel seems low, its listings have up to hundreds of product models referenced by popular PC brands such as HP, Asustek, Dell, etc.

Table 3: Number of product listings with respect to BT SoC.

Vendor	SoC	QID(s)	Listing(s) Count
Intel	AX200	127398	17*
Texas Instruments	CC2564C	87924, 126789, 117855	14
Cypress	CYW20735B1	169362, 112768	1
Bluetrum Technology	AB32VG1	115952, 131655	201
Zhuhai Jieli Technology	AC6905X	91274	341
Zhuhai Jieli Technology	AC6925C	110839	376
Zhuhai Jieli Technology	AC6366C	136145	173
Actions Technology	ATS281X	124400, 124265	47
Qualcomm	WCN3990/8	96248, 116819	22
Qualcomm	CSR8811/CSR8510	30846, 58778, 70941	92
Espressif Systems	ESP32	**116661 (ESP32 Dual-Mode Stack)**	28
Harman International	JX25X	84803, 62232	63
Silabs	WT32i	49552	48
Total Listings			**1423**

It is worthwhile to mention that while we can get the total number of registered products within Bluetooth Listings website for a certain chipset QID, it only reveals a lower-bound of the exact total number of listings employing a particular chipset. This is because product vendors may choose to customize the design. In this case, the QID of the employed chipset or BT stack is written to a field called *Combined Designs*. Unfortunately, the Bluetooth Listings site do not cross-reference *Combined Designs* when searching for a QID, which makes it almost impossible to get the total number of

products employing a particular chipset. Furthermore, products employing modules derived from chipsets such as ESP32, CSR8811, etc are also not shown during the search of a chipset QID.

In conclusion, the total number of individual product models that are potentially affected by BRAKTOOTH *could be an order of magnitude higher than the listings estimated in Table 3.*

Table 4: Sample products using SoCs affected by BRAKTOOTH. The declaration ID references each product on the Bluetooth Listing search website [28].

Product Vendor	Product Type	Product Model	SoC Model	Declaration ID
Microsoft	Laptop	Surface Laptop 3 Surface Go 2 Surface Pro 7 Surface Book 3	Intel AX200	D048122
Dell	Desktop PC / Laptop	Optiplex 5070 Alienware M17 R3 **(Many more)**	Intel AX200	D044215
Sony	Smartphone	Xperia XZ2	WCN3990	D048452
Oppo	Smartphone	Reno 5G CH1921	WCN3998	D044072
Ericsson	Home Entertainment Hub	KDE20102	CSR8510	D054397
Volvo Technology	Automotive Infotainment	Volvo FH **(Many More)**	CSR8811/510	D053903
Hella GmbH	Electronic Control Unit	PMP3	CC2564C	D044076
Walmart Stores	Audio	Disco Lamp Speaker **(Many More)**	Jieli AC63XX	D049113
Walmart Stores	Audio	Small rugged speaker **(Many More)**	ATS281X	D048582
Panasonic	Audio	Sound Bar SC-HTB100	ATS281X	D053923
Becker Avionics	Aircraft Entertainment System	AMU6500 **(More)**	WT32i	D044945
u-box	Industrial IoT Module	NINA-W106	ESP32	D050596
Koyo Electronics	PLC (Industrial Automation)	C2-02CPU	ESP32	D050601

From the listings shown in Table 3, we capture some interesting products employing the vulnerable chipsets and list them in Table 4. The observed types of products range from audio appliances (BT Speakers, headsets, ambience, etc) to personal PC/laptop computers, smartphones, and surprisingly even automotive multimedia electronic control units (**PMP3**), automotive infotainment systems (**Volvo FH**) and in-flight audio systems such as **AMU6500**. Notably **Volvo FH** and **AMU6500** are employing **Qualcomm CSR8811/510** and **Silabs WT32i** chipsets, respectively. As discussed in Section 5, the Qualcomm CSR8811/510 is unlikely to receive a patch (affected by **V6**) and we are not able to receive a response from Silicon Labs regarding their status of the investigation on Silabs WT32i (affected by **V5**).

4.5 Product Design Considerations

It is important to clarify that any product employing a vulnerable Bluetooth chipset, is not necessarily insecure (nevertheless, affected due to BT connectivity being impaired). The overall security of an end-product, which has an internal chipset with firmware flaws, depends on how much the product relies on such a vulnerable chipset for its main functionality.

Figure 4 showcases common BT product design strategies and their dependency on the BT chipset or stack.

1. In the **Isolated Design**, products may have their main processor application (*Product SW*) communicating with a standalone BT chipset via a simplistic interface (e.g., serial AT commands, SPI/I2C transfers, etc). Such products may not need targeted handling of vulnerabilities in the BT stack and therefore are less impacted by a vulnerable BT chipset. This means that BT vulnerabilities are not likely to affect the Product SW or at least, the design allows to fully isolate attacks coming from a vulnerable chipset. **WT32i** is an example of such a *BT Standalone* chipset.

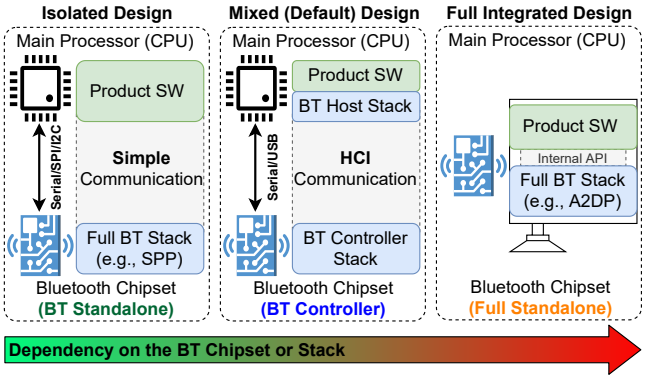

Figure 4: Examples of Bluetooth Product Design

2. In between, a **Mixed Design**, which shares the BT stack between the main processor and the BT controller, are somewhat susceptible to attacks on the specific part of the BT stack. Since BRAKTOOTH affects the BT Baseband or LMP implementation, the host stack is not affected directly. However, since both the *BT Controller Stack* and the *BT Host Stack* are continuously communicating depending on what happens inside the *BT Controller*, issues on the *Product SW* could arise if the HCI interface hangs. HCI interface may also misbehave in terms of whether the *BT Controller* provides expected responses or not. Therefore, it is up to the product software designer to investigate and handle all potential issues that can arise in the *BT host stack*. In general, this involves significant expertise and cannot be easily derived. The Mixed Design choice was unfortunately observed to affect products that rely on MSP432 as the main processor and **Texas Instruments CC2564C** as the BT controller when we had tested a sample code from TI's BT stack (CC2564CMSP432BTBLESW1 (v4.2.1.1) + Service pack v1.4). In this context, attack V5 (8.5) causes a temporary hang on MSP432, which could cause an impact on the underlying product. In contrast, operating systems in smartphones and laptops have a mature mechanism to handle BT controller hardware errors originated from attacks in the BT controller. See Section 4.2 where we illustrate the impact of BRAKTOOTH attacks on smartphones and laptops.

3. Lastly, we have the **Fully Integrated Design**, which is an obvious choice for cost reduction and is widely deployed for IoTs, speakers, headphones, and computer accessories. This design is susceptible to BRAKTOOTH hangs, crashes, and RCE directly on the main product functionalities. Such is due to both the *Product SW* and *BT Stack* being coexistent on the

Table 5: Patching status, Vulnerabilities and SDK/Firmware version of affected devices.
* Contact vendor to acquire the patch.

SoC or Module Vendor	BT SoC	Firmware or SDK Ver.	Vuln. / Anomalies	Patch Status
Espressif Systems	ESP32	esp-idf-4.4	V1,V3-4 / A1	[10] **Available**
Intel	AX200	Linux - ibt-12-16.ddc Windows - 22.40.0	V15-16 / A1-2, A5	Patch in progress
Qualcomm	WCN3990/8	crbtfw21.tlv, patch 0x0002	V15 / A1-2,A4	Patch in progress
Qualcomm	CSR8811/CSR8510	v9.1.12.14	V6 / A1-2	No fix
Texas Instruments	CC2564C	cc256xc_bt_sp_v1.4	V12 / A1-2	No fix
Infineon (Cypress)	CYW20735B1	WICED SDK 2.9.0	V12-15 / A2,A6	**Available ***
Bluetrum Technology	AB5301A	V06X_S6645 (LMP Subver. 3)	V7,V12 / A1-2	**Available ***
Zhuhai Jieli Technology	AC6925C	*unspecified* (LMP Subver. 12576)	V8-9 / A1-2	Investigation in progress
Zhuhai Jieli Technology	AC6905X	*unspecified* (LMP Subver. 12576)	V5,V8-9 / A1-2	Investigation in progress
Zhuhai Jieli Technology	AC6366C	fw-AC63_BT_SDK 0.9.0	V2,V12 / A1-2	Patch in progress
Actions Technology	ATS281X	*unspecified* (LMP Subver. 5200)	V4,V10-11 / A1-2	Investigation in progress
Harman International	JX25X	*unspecified* (LMP Subver. 5063)	V4 / A1-2	Pending
Silabs	WT32i	iWRAP 6.3.0 build 1149	V5 / A1-2	Pending

same hardware. This is exemplified by V1 attack (8.1), which affects ESP32 running Bluedroid BR/EDR stack.

To conclude, the assessment of a product based on its dependency on the BT chipset is a fundamental approach to understand and calculate the final risk factors. *Nevertheless, since* BRAKTOOTH *affects the lower layers of several BT implementations, at least BT connectivity is impaired regardless of the design choice*. Therefore, this should be taken into consideration when evaluating the product risks. Specifically, it needs to be evaluated whether a stable BT connectivity is always required for the assessed product to properly function.

5 BT Firmware Patches

Table 5 captures the affected vendors and the status of their investigation. We categorize the status of the investigation in the following forms:

- ***Available***: The vendor has replicated the vulnerability and a patch is available.
- ***Patch in progress***: The vendor has successfully replicated the vulnerability and a patch for the same will be available soon.
- ***Investigation in progress***: The vendor is currently investigating the security issue and our team is assisting them.
- ***No fix***: The vendor has successfully replicated the issue, but there is no plan to release a patch.
- ***Pending***: The vendor hardly communicated with the team and the status of their investigation is unclear at best.

We observe that **Espressif Systems**, **Infineon** (former Cypress) and **Bluetrum** have so far produced the patches. All other vendors are currently at the stage of investigation or patch development of the reported issues. We are working with the vendors to help them reproduce the issues in their systems such that patches are available to the public.

The vendor **Texas Instruments** has successfully replicated the security issue, however, at this stage has no plan for producing a patch. In particular, according to the Texas Instruments PSIRT team, they will consider producing a patch only if demanded by customers.

Our team approached Qualcomm to inquire whether a patch would be available for the affected devices. We were informed that they are working on a fix for WCN3990/8 and that the security issue reported in Qualcomm CSR8811A08 [23] has been fixed since 2011 *only* for ROM Versions A12 and beyond. However, new products in 2021 are still being listed to use CSR8811A08, which has no plan to be fixed. Moreover, a patch for the issue on CSR8510A10 [22] that causes **V6** (see Section 8.6) is not possible for CSR8510A10 due to the lack of ROM patch space.

Zhuhai Jieli Technology confirmed that a fix for AC6366C [33] would be released soon, but they did not confirm whether a patch would also be available to their other audio devices AC6925C and AC6905X. Finally, we notice for certain vendors (e.g. Harman International, Silabs), the investigation status is unclear, as indicated by the *Pending* status in Table 5.

6 Sniffing BT BR/EDR in less than $15

As part of our work of reverse engineering ESP32 BT stack, we are releasing to the community a low-cost BT Classic (BR/EDR) Active Sniffer which is available at the following URL:

https://github.com/Matheus-Garbelini/esp32_bluetooth_classic_sniffer

At the time of writing, this is the cheapest BR/EDR active sniffer, we are aware of due to the low price of ESP32 boards. ESP32-PICO-KIT can be purchased for $**14.80** [20], but it is possible to find alternative ESP32 boards on Aliexpress for as low as $**4** [8].

Note that differently than **passive** sniffers, which does not interact with the network, the ESP32 **active** sniffer must act as either a BT Master or Slave device within the BT piconet. As exemplified in Figure 5, the sniffer logs BT BR/EDR OTA protocols such as Baseband header, FHS, LMP, and ACL packets. The sniffer cannot be used to inject packets at the moment due to the PoC embargo. The embargo will be lifted at the end of October 2021 and our full PoC tool will be made available to the public for research and reproduction.

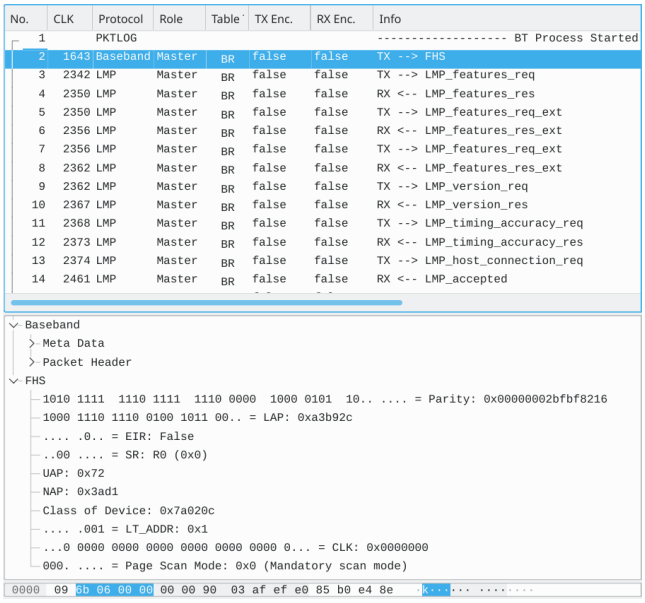

Figure 5: Active BR/EDR Sniffing of piconet. The sniffer captures Baseband + FHS and LMP frames.

7 Reflection

In recent years, Bluetooth has come under scrutiny due to several design and implementation level vulnerabilities, such as KNOB [2], BIAS [1], SweynTooth [13], BLESA [34], BlueMirror [5], etc. Our previous disclosure of SweynTooth [13] vulnerabilities (2020) has clearly indicated the lack of basic tests in Bluetooth certification to validate the security of Bluetooth Low Energy (BLE) devices. The BRAKTOOTH family of vulnerabilities revisits and reasserts this issue in the case of the older, but yet heavily used Bluetooth classic (BR/EDR) protocol implementations. Additionally, during the responsible disclosure period, we had the following observations that might shed light on the future research in Bluetooth security.

Replication difficulty: Bluetooth Core Specifications allows a limited "LMP test mode" [27]. This limits the vendors to operate with only a few LMP procedures and therefore, the vendors are unlikely to have full control over the messages exchanged over *all LMP procedures* without expensive certification hardware. All BRAKTOOTH vulnerabilities **V1**-**V16** involve specific mutation or duplication to create potentially unexpected scenarios in the Bluetooth link manager (see Section 8 for details). We postulate that due to the lack of tools (or flexibility thereof) to control all LMP procedures, BT devices were not tested thoroughly on such uncommon scenarios. This is also evident during our communication even with popular BT vendors (e.g. Intel, Cypress, and Qualcomm). Specifically, in contrast to our disclosure of SweynTooth [13] vulnerabilities, we experienced the following crucial differences: 1) The replication of BRAKTOOTH vulnerabilities involves significantly more interaction with the vendors. In the past, our experience with BLE vendors suggests that they were able to use in-house tools (without using our custom fuzzer) to replicate the SweynTooth vulnerabilities. However, while replicating BRAKTOOTH, vendors had to use the exact setup used to discover BRAKTOOTH vulnerabilities. This involves the usage of the ESP32 device, the custom LMP firmware and the BRAKTOOTH PoC tool (see Figure 2). We believe such is the case due to the lack of BT tooling for comprehensively testing the security of Bluetooth link manager implementations under unexpected packet exchanges. Thankfully, our PoC brings the control over LMP to the host, dismissing the need to change the BT firmware just to create specific test cases. This approach was inspired from our previous work with the "SweynTooth Non-Compliant BLE Controller" [13].

Downstream dependency: During the responsible disclosure period, we experienced that it is impossible for certain BT module vendors to produce a patch due to their dependency on downstream vendors. Specifically, this was observed while reporting the vulnerability **V6** (**LMP 2-DH1 Overflow**) for Laird BT900 and BT820 SoC devices. A discussion with Laird Technologies, Inc. revealed that the underlying LMP Stack used by both BT900 (*CSR8811*) and BT820 (*CSR8510*) devices is Stonestreet One's Bluetopia stack (currently acquired by Qualcomm) and the rectification of **V6** requires the support of Qualcomm. After a discussion between our team and Laird Connectivity, it was revealed (from Laird Connectivity contacting Qualcomm internal support forum) that the CSR8811 has no patch space available and therefore, the vulnerability **V6** is not possible to fix. In other words, any devices using *CSR8811*, including but not limited to BT900 and BT820 SoC devices, are likely to remain vulnerable forever. This clearly shows the complexity of resolving IoT vulnerabilities due to the downstream dependencies on multiple vendors. The implication of such a problem is not trivial to gauge. Considering that certain devices may remain vulnerable for a prolonged time, it is advised that the IoT module and product vendors conduct a thorough risk assessment. This is

particularly important if the target module or product uses a vulnerable component where patching is difficult due to software lock-down (similar to *CSR8811*).

Reverse engineering effort: During our research on Bluetooth Link Manager testing, we clearly observe the gap of security testing tools to validate arbitrary BT devices. While there exists emulation-based approaches [25], these works need to spend significant effort in reverse engineering the target devices (if at all possible). The absence of any open source and feature-complete Bluetooth LMP stack further complicates the security testing problem. During our research, we have designed a fuzzing interface by reverse-engineering the ESP32 BT stack which is located in the static library *libbtdm_app.a* [9] and ESP32 ROM memory. The reverse engineering effort was utilized to design a (non-compliant) LMP firmware that can be used off-the-shelf to validate any devices with BR/EDR functionality. While the reverse-engineering effort of a commodity BT stack was significant, we believe that the effort was fruitful and necessary in comprehensively analysing the security of any BT implementations in the wild. As discussed in the preceding paragraphs, we also experienced that even major industries lack flexible tools for comprehensively testing BT Link Managers.

8 Vulnerabilities Description

In this section, we provide a detailed description of each vulnerability, the affected system-on-chip (SoC) models and the SDKs where applicable. Some vulnerabilities were discovered when testing developments kits and others were detected by testing final products (e.g. speakers or headphones).

8.1 V1: Feature Pages Execution (CVE-2021-28139)

The Bluetooth Classic implementation in Espressif ESP-IDF 4.4 and earlier [7] does not properly restrict the Feature Page upon reception of an LMP Feature Response Extended packet, allowing attackers in radio range to trigger arbitrary code execution (ACE) in ESP32 via a crafted Extended Features bitfield payload. As shown in Figure 6, the ACE vulnerability is triggered by simply sending *LMP_feature_res_ext* with an invalid feature page of 0*x*62 after *LMP Setup* procedure (c.f., Figure 1). Moreover, the contents of extended features in the *LMP_feature_res_ext* can be used to execute code at an arbitrary address within the ESP32 firmware. For example, as described in Section 4.1, such an address may point to the erase function (*nvs_flash_erase*) and disabling Wi-Fi (*disable_wifi_agc*), among others.

Impact: The attacker is able to execute arbitrary functions implemented in the target ESP32's firmware. Once the attacker acquires a dump of ESP32 firmware and the memory layout is known, functions at arbitrary addresses can be called. This allows the attacker to erase user data (NVRAM), call I/O functions that may control actuators, etc. Triggering firmware crashes or a BT deadlock is also possible during repeated attack attempts by executing invalid function addresses. In case of a deadlock, the user needs to manually reset ESP32 to restore BT communications.

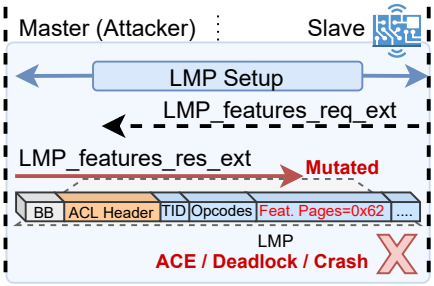

Figure 6: Feature Page Execution

8.2 V2: Truncated SCO Link Request (CVE-2021-34144)

The Bluetooth Classic implementation in the Zhuhai Jieli AC6366C BT SDK 0.9.1 and earlier [33] does not properly handle the reception of truncated LMP_SCO_Link_Request packets while no other BT connections are active. This allows attackers in radio range to prevent new BT connections (disabling the AC6366C inquiry and page scan procedures) via a crafted LMP packet. The user needs to manually perform a power cycle (restart) of the device to restore BT communication. Following Figure 7, the attack can be launched by sending an *LMP_SCO_Link_Request* with *ACL length=2* instead of its normal size of 7.

Impact: The attacker is able to prevent external parties to connect to the device, requiring the user to manually perform a power cycle on the device to resume normal connectivity. Attack risk is reduced if the user is already connected to AC6366C or similar chipsets that only allow one active BT connection at a time.

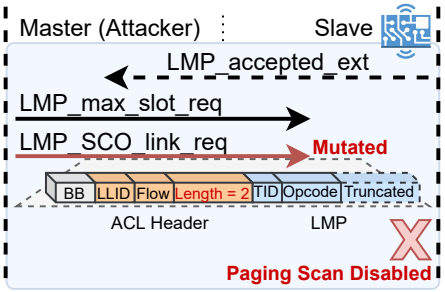

Figure 7: Truncated SCO Link Request

8.3 V3: Duplicated IOCAP (CVE-2021-28136)

The Bluetooth Classic implementation in Espressif ESP-IDF 4.4 and earlier [7] does not properly handle the reception of multiple *LMP_IO_Capability_req* packets during the *pairing process* (see Figure 8). This allows attackers in radio range to trigger memory corruption (and consequently a crash) in ESP32 via a replayed (duplicated) LMP packet. The issue was raised due to the sudden disconnection of ESP32 while its bluedroid host stack was still in the pairing process. The patch provided by Espressif rectifies this issue by handling a sudden disconnection during the pairing procedure (c.f., Table 5).

Impact: The attacker can exploit this vulnerability to promptly cause firmware crashes and therefore maintain a DoS while the attack is taking place. Since ESP32 restarts its firmware by default upon receiving a fault, no user interaction is needed to restore BT communication after the attack is stopped.

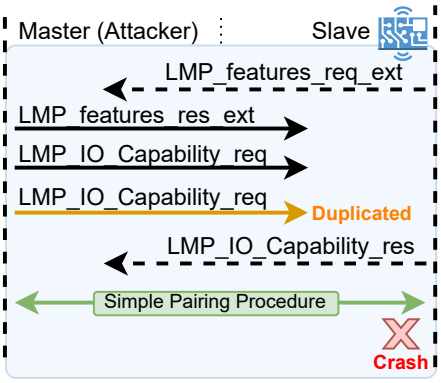

Figure 8: Duplicated IOCAP

8.4 V4: Feature Response Flooding (CVE-2021-28135/28155/31717)

The Bluetooth Classic implementation in Espressif ESP-IDF 4.4 and earlier [7] does not properly handle the reception of continuous unsolicited LMP responses. This allows attackers in radio range to trigger a denial of service (crash) in ESP32 by flooding the target device with LMP Feature Response data. As shown in Figure 9, *LMP_features_res* packets are sent within a BT transmission slot of 1.25ms, which effectively floods the Slave with unsolicited responses. This attack vector is also effective against a range of other BT chipsets such as **JBL TUNE500BT** [17] and **Xiaomi MDZ-36-DB (Actions Semi. ATS2815)** [32, 35]. See Table 2 for all the targets affected by V4.

Impact: The attack triggers a firmware crash on ESP32 and a shutdown on **JBL TUNE500BT** and **Xiaomi MDZ-36-DB**. In the case of a shutdown, the user needs to manually power on the device again to restore communication. Furthermore, since all targets accept multiple BT connections, the attack can also be triggered while the target is already connected to another device. For example, it is possible to immediately stop audio from playing on JBL TUNE500BT and Xiaomi MDZ-36-DB. The attacker, however, needs to know the BDAddress of the target to launch the attack.

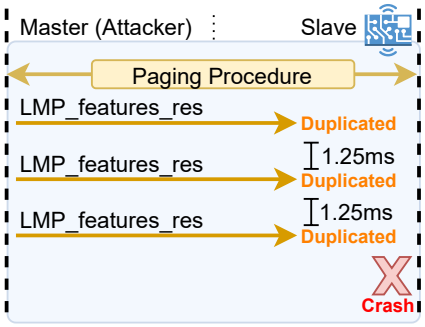

Figure 9: Feature Response Flooding

8.5 V5: LMP Auto Rate Overflow (CVE-2021-31609/31612)

The Bluetooth Classic implementation in Silicon Labs iWRAP 6.3.0 and earlier [18] does not properly handle the reception of an oversized LMP packet greater than 17 bytes, allowing attackers in radio range to trigger a crash in WT32i [19] via a crafted LMP packet. As shown in Figure 10, the attack is triggered by sending an LMP with an ACL size higher than 17 bytes. Normally this is not allowed by the BT Core Specification [26], but ESP32 radio can indeed send such oversized LMP packets over the DM1 transport channel. Our proof of concept (PoC) firmware bypasses some length checks on the transmission path to make this possible. This attack vector also works on some **Jieli AC6905X (BT Audio Receiver)**. Note that this attack requires sending this malformed packet many times over many reconnections to trigger a firmware crash. Particularly, a firmware crash on WT32i may take around 3-5 minutes to be triggered.

Impact: The attack causes a firmware crash on both **WT32i** and **AC6905X**. This can be used to cause a sporadic denial of service (DoS) attacks on such devices.

8.6 V6: LMP 2-DH1 Overflow (Pending CVE)

The Bluetooth Classic implementation on Laird CSR8811 A08 [23] and CSR8510 A10 [22] SoCs allows an LMP length overflow over 2-DH1, resulting in a Deadlock (state machine or packet handler corruption) outcome. Figure 11 captures the sequence of messages exchanged to expose this vulnerability.

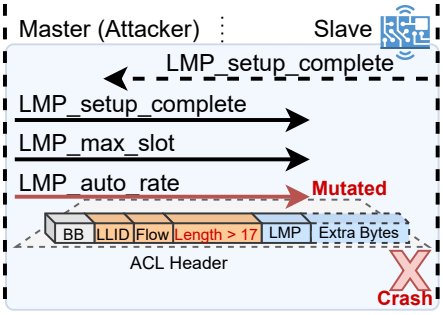

Figure 10: LMP Auto Rate Overflow

In particular, the attack is performed by changing the transport type of the normal LMP packet to *2-DH1* and then adding more bytes to the LMP payload. This, for example, results in a total ACL length of 27.

Impact: The attack causes a BT deadlock on Laird CSR8811 and a firmware crash on CSR8510 operating as a standard BT dongle. While the dongle firmware crash is automatically recovered by the OS, the deadlock requires the user to perform a power cycle to restore normal BT communication. The attack does not affect Bluetooth Low Energy (BLE) communication in CSR8811.

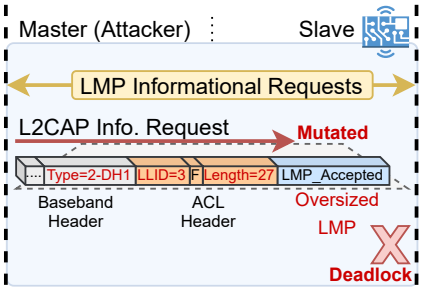

Figure 11: LMP 2-DH1 Overflow

8.7 V7: LMP DM1 Overflow (CVE-2021-34150)

The Bluetooth Classic implementation on Bluetrum AB5301A [4] (tested with AB32VG1 dev. kit [3]) with *unspecified* firmware versions does not properly handle the reception of oversized DM1 LMP packets. This allows attackers in radio range to prevent new BT connections (disabling the AB5301A inquiry and page scan procedures) via a crafted LMP packet. Following Figure 12, an arbitrary LMP packet (*LMP_packet_type_table_req*) is sent with an ACL length of 31, which is much higher than the expected 17 byte limit for DM1 transport. Due to a lack of check of DM1 length, an undefined behaviour on the target baseband implementation was observed. This eventually disables the paging scan procedure and hence blocks new connections. If the user is already connected to AB5301A and the attack is started (multiple BT connections are allowed), the active connection is not disrupted, but new connections or reconnection is not possible. As compared to **V5** (Section 8.5), this attack works with any overflown LMP packet rather than a specific LMP packet.

Impact: The impact of this attack is similar to **V2** (Section 8.2), in which connections to the target are not accepted due to the page scan procedure being disabled after the attack (see Figure 1). Therefore, the user needs to manually perform a power cycle (restart) of the device to restore normal BT connectivity.

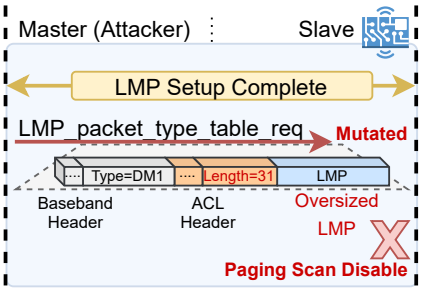

Figure 12: LMP DM1 Overflow

8.8 V8: Truncated LMP Accepted (CVE-2021-31613)

The Bluetooth Classic implementation on Zhuhai Jieli AC690X and AC692X devices with *unknown* firmware does not properly handle the reception of a truncated LMP packet during LMP auto rate procedure, allowing attackers in radio range to immediately crash (and restart) a device via a crafted LMP packet. As depicted in Figure 13, an attacker sends a malformed *LMP_accepted* just after the LMP Setup procedure. This malformed packet has an ACL size of one byte, rather than the expected 2 bytes for the opcode *LMP_accepted*. The implementation of the target BT device does not properly reject such packets with a truncated size. This eventually results in a firmware crash, which, in turn, triggers a prompt restart.

As the BT implementation of the target is closed source, the exact root cause of this vulnerability is not known. However, it is common that LMP procedures involving slot and rate adjustments are usually implemented in an interrupt handler different than other non-real-time LMP packets, such as feature requests and version requests. Such a handler may miss some checks which are present on the other handlers, explaining why some overflow/underflow attacks only work for certain LMP packet opcodes. This split in the LMP handler is present, for instance, in ESP32 BT implementation.

Impact: The impact is a denial of service due to a firmware crash & restart. The overall risk of such an attack is reduced if

AC690X and AC692X do not allow multiple BT connections. In such a scenario, the attacker could only trigger a firmware crash when no user is connected to AC690X or AC692X.

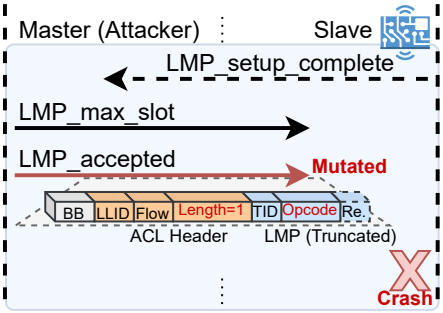

Figure 13: Truncated LMP Accepted

8.9 V9: Invalid Setup Complete (CVE-2021-31611)

The Bluetooth Classic implementation on Zhuhai Jieli AC690X and AC692X devices does not properly handle the reception of an out-of-order *LMP Setup* procedure (c.f., Figure 1) followed by a malformed LMP packet.

This scenario is illustrated in Figure 14. The attacker sends an unexpected (duplicated) *LMP_setup_complete*, followed by a malformed *LMP_features_req_ext* with an invalid LMP opcode such as $0x64$. The behaviour seen in Figure 14 allows attackers in radio range to deadlock a device via injecting crafted LMP packets. Moreover, the user needs to manually reboot the device to restore communication.

Impact: An attacker can exploit this vulnerability to perform a DoS. The overall risk of such an attack is reduced if AC690X and AC692X do not allow multiple BT connections.

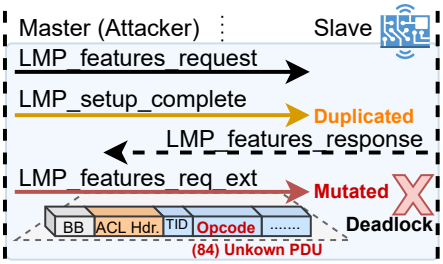

Figure 14: Invalid Setup Complete

8.10 V10: Host Connection Flooding (CVE-2021-31785)

The Bluetooth Classic implementation on Actions ATS2815/ATS2819 [32] chipsets does not properly handle the reception of multiple *LMP_host_connection_req* as shown in Figure 15. This allows attackers in radio range to trigger a denial of service (deadlock) if no user is connected to the target. Manual user intervention is required to restart the device and restore Bluetooth communication.

Impact: An attacker in radio range can exploit the vulnerability to perform DoS and shutdown or restart products using chipsets and products based on Actions ATS2815/ATS2819. However, the impact of the attack is reduced as the vulnerability is only exploited when no other user device is connected to the target.

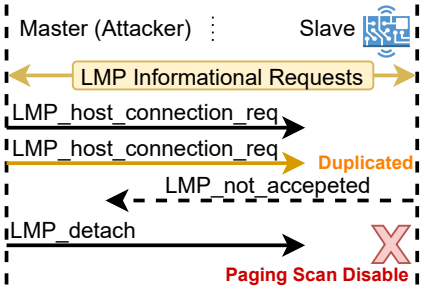

Figure 15: Host Connection Flooding

8.11 V11: Same Host Connection (CVE-2021-31786)

The Bluetooth Classic Audio implementation on products based on Actions ATS2815/ATS2819 [32] chipset may not properly handle a connection attempt from a host with the same BDAddress as the currently connected BT host. As illustrated in Figure 16, by simply connecting to the target device (BT Speaker) with a forged BDAddress that matches the BDAddress of the originally connected host (Device A - *E0:D4:E8:19:C7:69*), the target device triggers a disconnection and eventually deadlocks. This requires the user to reboot the target device to restore BT functionality.

When reaching Actions Semi during the disclosure period, the company has informed us that they recommend the product vendors to only allow one audio BT connection. However, when we tested an ATS chipset based product *Xiaomi MDZ-36-DB*, we observed that multiple BT connections are allowed, albeit only one device can play audio at a time. Normally, connections with repeated host BDAddress are rejected on *LMP Setup* procedure (c.f., Figure 1). However, this procedure is successful when testing *Xiaomi MDZ-36-DB* with repeated host BDAddress, leading the audio device to an undefined state.

Impact: An attacker in radio range can exploit this vulnerable behaviour to perform DoS and intentionally deadlock the target device.

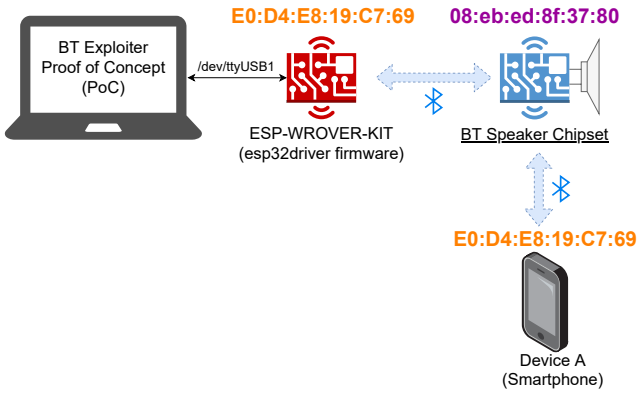

Figure 16: Same Host Connection

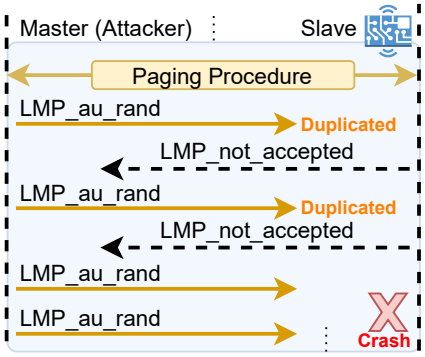

Figure 17: LMP AU Rand Flooding

8.12 V12: LMP AU Rand Flooding (CVE-2021-31610/34149/34146/34143)

The Bluetooth Classic implementation on several chipsets does not properly handle the reception of continuous unsolicited LMP responses that trigger a heap overflow within the BT firmware. This allows attackers in radio range to trigger a denial of service (either restart or deadlock the device) by flooding a device with *LMP_AU_rand* packet as shown in Figure 17. Only **CC2564C** [14] from *Texas Instruments* enters a deadlock state, which requires user intervention to recover. Additionally, use of *TI Dual-mode Bluetooth Stack for MSP432* [15] may temporarily hang the host MCU (MSP432) during the attack.

Many major chipset vendors are affected by this attack as shown in Table 2. This attack is similar to Feature Response Flooding (Section 8.4), indicating that flooding testing with packets from certain LMP procedures may not have been well tested. This may arise because the *Core Specifications* only allows a limited LMP testing mode [27] that restricts the SoC to work with only a few LMP packets. This, in turn, may restrict the test flexibility for BT vendors. Specifically, vendors may have a limited capacity of tests that they can perform with their production BT firmware, such as flooding or sending out-of-order packets during normal LMP procedures.

Impact: An attacker in radio range can exploit this vulnerable behaviour to perform DoS and intentionally crash or deadlock the target device.

8.13 V13: LMP Invalid Max Slot Type (CVE-2021-34145)

The Bluetooth Classic implementation in the Cypress WICED BT stack 2.9.0 [12] and earlier for CYW20735B1 [11] devices does not properly handle the reception of *LMP_max_slot* with an invalid Baseband packet type and *LT_ADDRESS* (see *Type* and *LT_ADDR* fields in Figure 18). The vulnerability is triggered after completion of the LMP setup procedure, allowing attackers in radio range to trigger a denial of service (firmware crash) via a crafted LMP packet.

Impact: An attacker in radio range can exploit this vulnerable behaviour to perform DoS and intentionally crash the target device.

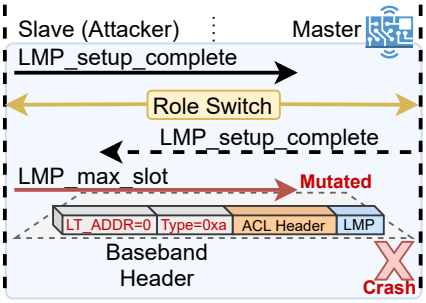

Figure 18: Invalid Max Slot Type

8.14 V14: Max Slot Length Overflow (CVE-2021-34148)

The Bluetooth Classic implementation in the Cypress WICED BT stack 2.9.0 [12] and earlier for CYW20735B1 [11] devices does not properly handle the reception of *LMP_max_slot* with a higher ACL Length (*Length=31* as shown in Figure 19) after completion of the LMP setup procedure. This allows attackers in radio range to trigger a denial of service (firmware crash) via a crafted LMP packet.

Impact: An attacker in radio range can exploit this vulnerability to perform DoS and intentionally crash the target device.

8.15 V15: Invalid Timing Accuracy (CVE-2021-34147/Pending/Pending)

The Bluetooth Classic implementations in the Cypress WICED BT stack 2.9.0 [12] and earlier for CYW20735B1 [11] devices, **Intel AX200** [16] and **Qualcomm WCN3990**

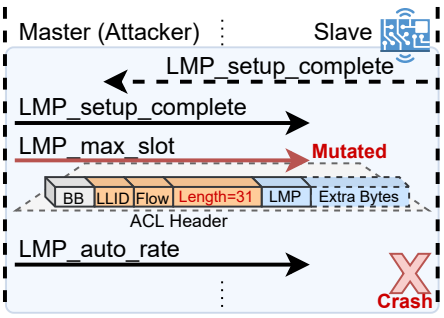

Figure 19: Max Slot Length Overflow

[29] chipsets do not properly handle the reception of a malformed LMP timing accuracy response followed by multiple re-connections to the target link slave. This allows attackers to exhaust device BT resources. Eventually, the attacker can trigger a crash or BT disturbance via multiple attempts of sending a crafted *LMP_timing_acc_response* (i.e. LMP timing accuracy response) followed by a sudden reconnection to the target with a random BDAddress.

As shown in Figure 20, the attacker needs to perform a loop of reconnection and injection of the malformed *LMP_timing_acc_response* until the target chipset gets unstable. This either triggers a firmware crash (in the case of WCN3990 and CYW20735B1) or disconnects other active BT devices (in the case of Intel AX200). The faster the reconnection is performed, the easier is for the attack to succeed in disturbing other BT devices connected to the target chipset.

Impact: This attack causes a denial of service (DoS) attack. Specifically, while the attack is in progress, the target devices are unable to use the Bluetooth service normally.

The impact of this attack does not persist after the attack stops. This is because the target normally tries to recover the BT connection by performing a re-connection to previously disconnected devices (e.g., Intel AX200 under Linux or Windows). In the case of a WCN3990 firmware crash, the Android OS restarts the Bluetooth daemon and re-uploads a firmware image to WCN3990. Finally, for CYW20735B1, it restarts automatically upon any fault due to its watchdog timer being enabled by default. Nevertheless, in all cases, it is not possible to normally use the device while the attack is in progress, as the BT connection can be disrupted continuously. The attacker only needs to know the BDAddress of the target device and no authentication is required to launch the attack.

8.16 V16: Paging Scan Disable (Pending CVE)

This vulnerability is depicted in Figure 21. As shown in Figure 21, sending an invalid packet during the LMP timing accuracy procedure (i.e. packet *LMP_timing_acc_request*), followed by a forced re-connection with the same BDAddress

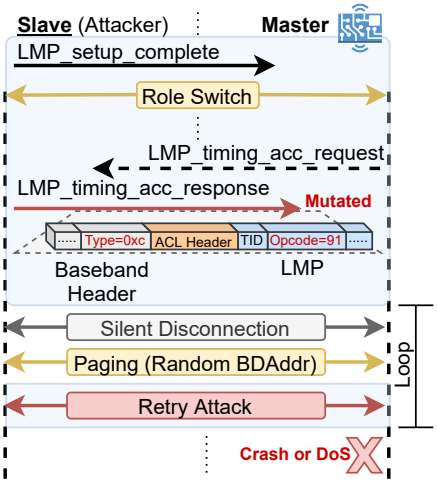

Figure 20: Invalid Timing Accuracy

(any arbitrary BDAdress of choice of the attacker), leads **Intel AX200** to reject any externally initiated BT connections for an undetermined amount of time. This persists even after the attack stops and requires user intervention to recover ***AX200*** normal functionalities.

Figure 21 illustrates the attack, which, similarly to attack **V15** (Section 8.15), involves a loop of sending the malformed packet throughout re-connections to trigger the vulnerability. In contrast to V15, the attack is triggered before role switch procedure (see Figure 1), requires the same BDAddress when initiating re-connections and injects a malformed *LMP_timing_acc_request* with an oversized LMP length greater than 17 bytes. As a reminder, the maximum length limit of an LMP packet under DM1 channel is 17 bytes, which suggests that oversized LMP packets are not correctly handled by **AX200** under a sudden disconnection scenario.

Furthermore, during the next re-connection, **AX200** sends an out-of-order response which does not correspond to the original request. For instance, during the second re-connection involved in the scenario captured in Figure 21, **AX200** sends *LMP_version_res* when receiving a feature request from the attacker. This depicts anomaly **A5** as listed in Table 2.

Finally, as a second side effect of the vulnerability, the user may not be able to initiate as much BR/EDR connections as AX200 originally supports even after the attack stops. For example, the user is able to connect a maximum of only one or two BR/EDR devices depending on when the vulnerability is triggered. More specifically, if the vulnerability is triggered when AX200 is not connected to any device, then the user can only connect to one device. Otherwise, if **AX200** is already connected to a device when the vulnerability is triggered, then **AX200** can only connect to two devices.

Impact: Once the attack is triggered and successful, the attacker can cause DoS via the following **AX200** behaviours: **(I)** paging scan is disabled, which prevents any external de-

vice to connect to the target even if the BDAddress is known. This behaviour can be used to trick an user to connect to the attacker's BT hardware instead of the legitimate target since **AX200** paging scan is disabled; **(II)** Multiple active BT connections cannot be performed from the target. The user requires to manually restart the Bluetooth service to restore normal BT behaviour. On Linux and Windows, BT is recovered by disabling and enabling Bluetooth via their respective configuration menu.

Firmware crashes may be sporadically triggered on **AX200** during the attacks, but no specific scenario was found to reliably trigger such crashes all the time.

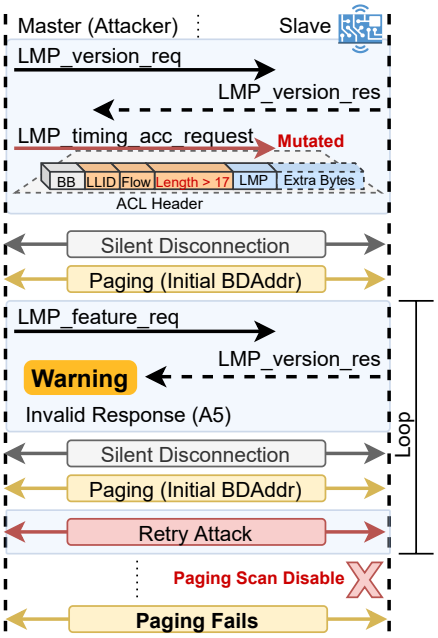

Figure 21: Paging Scan Disable

PoC Tool Availability

BRAKTOOTH proof-of-concept (PoC) tool is available to download for vendors producing BT SoCs, modules and products. To download the BRAKTOOTH proof-of-concept (PoC) tool, kindly fill up the following simple form: BRAKTOOTH PoC. The form requires certain basic information (job role, organization, and valid email) to be provided. The detailed instructions to download and launch the exploits on a target device will be sent (to the provided email) once the required information is given. We encourage vendors producing BT SoCs, BT modules, and BT end products to use the PoC and validate against BRAKTOOTH attacks. We will be glad to help with replication and validation. Send all questions related to BRAKTOOTH to ask@braktooth.com.

Acknowledgments

This research was partially supported by NRF National Satellite of Excellence in Trustworthy Software Systems. This research is also successfully translated to a Keysight penetration testing tool with the generous funding and support provided by Keysight Technologies. We also want to thank all the involved vendors for their support during the coordination process. We thank Ezekiel O. Soremekun for coining the term BrakTooth. We thank Rushati Chakraborty for creating the BrakTooth title design. Additionally, special thanks to Olof Astrand for his support on the open-source Tensilica Xtensa module for Ghidra [6] and Kwok Chuo Willis Yip for the ESP32 attack video.

References

[1] Daniele Antonioli, Nils Ole Tippenhauer, and Kasper Rasmussen. BIAS: bluetooth impersonation attacks. In *2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020*, pages 549–562. IEEE, 2020.

[2] Daniele Antonioli, Nils Ole Tippenhauer, and Kasper B. Rasmussen. 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, Santa Clara, CA, August 2019. USENIX Association.

[3] Bluetrum. AB32VG1 Development Kit Quick Start (CN). https://ab32vg1-example.readthedocs.io/zh/latest/introduction.html, 2021.

[4] Bluetrum. AB5301A SoC Product Description Website (CN). http://www.bluetrum.com/product/ab5301a.html, 2021.

[5] Tristan Claverie and Jose Lopes Esteves. Bluemirror: Reflections on bluetooth pairing and provisioning protocols. In *2021 IEEE Security and Privacy Workshops (SPW)*, pages 339–351, 2021.

[6] Ebiroll. Fork of Tensilica Xtensa module for Ghidra. https://github.com/Ebiroll/ghidra-xtensa, 2021.

[7] Espressif. Espressif IoT Development Framework. https://github.com/espressif/esp-idf, 2020.

[8] Espressif. Cheap ESP32 Development Kit. https://www.aliexpress.com/item/1005001757645011.html, 2021.

[9] Espressif. ESP32 BT/BLE Stack Libraries. https://github.com/espressif/esp32-bt-lib, 2021.

[10] Espressif. ESP32 ESP-IDF BrakTooth Patches (commit #bf71f49). https://github.com/espressif/esp-idf/tree/bf71f494a165aba5e5365e17e1e258598d9fc172, 2021.

[11] Infineon (former Cypress). CYW20735B1 Datasheet. https://www.cypress.com/file/298426/download, 2021.

[12] Infineon (former Cypress). WICED Software. https://www.cypress.com/products/wiced-software, 2021.

[13] Matheus E. Garbelini, Chundong Wang, Sudipta Chattopadhyay, Sun Sumei, and Ernest Kurniawan. Sweyntooth: Unleashing mayhem over bluetooth low energy. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, pages 911–925. USENIX Association, July 2020.

[14] Texas Instruments. CC2564C Product Details. https://www.ti.com/product/CC2564C, 2021.

[15] Texas Instruments. CC2564C TI Dual-mode Bluetooth Stack on MSP432 MCUs. https://www.ti.com/tool/CC2564CMSP432BTBLESW, 2021. Service pack must be downloaded separatelly.

[16] Intel. Intel Wi-Fi 6 AX200 SoC Product Details. https://ark.intel.com/content/www/us/en/ark/products/189347/intel-wi-fi-6-ax200-gig.html, 2021.

[17] JBL. JBL TUNE 500BT Wireless on-ear Headphone Product Details. https://www.jbl.com.sg/over-ear-headphones/JBL+TUNE500BT.html, 2021.

[18] Silicon Labs. Bluegiga iWRAP Bluetooth Classic Software Stack. https://www.silabs.com/developers/bluegiga-iwrap-bluetooth-classic-software-stack, 2021.

[19] Silicon Labs. WT32I-A Product Description. https://www.silabs.com/wireless/bluetooth/bluegiga-classic-legacy-modules/device.wt32i-a, 2021.

[20] Mouser. Official ESP32 Development Kit ESP32-PICO-KIT. https://www.mouser.sg/ProductDetail/Espressif-Systems/ESP32-PICO-KIT?qs=MLItCLRbWsyoLrlknFRqcQ%3D%3D, 2021.

[21] Oppo. Oppo Reno 5G (CPH1921) Product Details. https://www.oppo.com/en/smartphone-reno-5g/specs/, 2019.

[22] Qualcomm. CSR8510 Bluetooth SoC Product Details. https://www.qualcomm.com/products/csr8510, 2021.

[23] Qualcomm. CSR8811 Bluetooth SoC Product Details. https://www.qualcomm.com/products/csr8811, 2021.

[24] DF Robot. XY-WRBT Bluetooth 5.0 Audio Receiver Product Details. https://www.dfrobot.com/product-2084.html, 2021.

[25] Jan Ruge, Jiska Classen, Francesco Gringoli, and Matthias Hollick. Frankenstein: Advanced wireless fuzzing to exploit new bluetooth escalation targets. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 19–36. USENIX Association, August 2020.

[26] Bluetooth SIG. Bluetooth certification guideline: Qualify your product. https://www.bluetooth.com/develop-with-bluetooth/qualification-listing/, 2019.

[27] Bluetooth SIG. Bluetooth Core Specification v5.2, December 2019. https://www.bluetooth.com/specifications/bluetooth-core-specification.

[28] Bluetooth SIG. View previously qualified designs and declared products, January 2020. https://launchstudio.bluetooth.com/Listings/Search.

[29] Bluetooth SIG. WCN3990 Bluetooth Product Listings. https://launchstudio.bluetooth.com/ListingDetails/66490, 2021.

[30] Espressif Systems. ESP32 SoC Product Details. https://www.espressif.com/en/products/socs/esp32, 2021.

[31] Espressif Systems. Official ESP32 Development Kit ESP32-WROVER-KIT. https://www.espressif.com/en/products/hardware/esp-wrover-kit/overview, 2021.

[32] Actions Technology. ATS2815 Datasheet. http://www.lanzhi-tech.com/filedownload/99240, 2021.

[33] Jieli Technology. AC63 Series SDK repository. https://github.com/Jieli-Tech/fw-AC63_BT_SDK, 2021.

[34] Jianliang Wu, Yuhong Nan, Vireshwar Kumar, Dave Jing Tian, Antonio Bianchi, Mathias Payer, and Dongyan Xu. *BLESA*: Spoofing Attacks against Reconnections in Bluetooth Low Energy. In *14th USENIX Workshop on Offensive Technologies (WOOT 20)*, 2020.

[35] Xiaomi. Mi Portable Bluetooth Speaker MDZ-36-DB Product Details. https://www.mi.com/global/mi-portable-bluetooth-speaker-16w/specs/, 2021.