From Bluetooth vulnerabilities to a remote code execution on an Android phone using BlueBorne

Guillaume Bour
guillanb@stud.ntnu.no
TTM4137 Wireless Security Technical Essay

November 5, 2018

1 Introduction

Bluetooth is a widespread technology today, known and used by everyone that has ever connected a wireless keyboard or earphones to her phone or computer. Over the years, Bluetooth has suffered from multiple vulnerabilities. No later than last summer, a new vulnerability affecting Android and Apple products has been disclosed.[1] In this technical essay, we try, using BlueBorne vulnerabilities, to understand why so many vulnerabilities exist in the Bluetooth stack and how they can lead to the compromission of a device.

2 Background

2.1 Bluetooth technology

Bluetooth is a Wireless technology standard developed by Ericson in 1994 which allows data exchange on a short distance (around 10m in average). Although Bluetooth was known as IEEE 802.15.1, it is no longer maintained by the IEEE and new Bluetooth devices must now meet the requirements of the Bluetooth Special Interest Group (SIG) [11].

Bluetooth started becoming popular in 2004 with its 2.0 version, mostly because of its improved data rate theoretically reaching 3 Mbit/s. At the time, encryption was optional and could even be disabled [11]. This issue was addressed with Bluetooth 2.1, released in 2007, which introduced the Secure Simple Pairing (SSP). This made the pairing process easier and more secure. It was also believed to protect from Man in the Middle (MitM) attacks. In this process, keys are derived using the PIN code [5, 11]. Speed increased again with Bluetooth 3.0 in 2009 as data exchange was then using the Wi-Fi radio link. The rate could reach 24 Mbit/s. Bluetooth 4.0 started to focus on sensors and other connected devices. That is why new low energy protocols were developed in Bluetooth. The Security Manager (SM) also started using AES as encryption algorithm. This manager takes place right above L2CAP in the Bluetooth stack (see SMP on Figure 1) and is responsible for pairing, encrypting and signing [5]. The 4.1 and 4.2 versions, in 2013 and 2014 respectively, followed that path, focusing on IoT and improving for instance LTE compatibility along with enabling sensors to directly connect to the internet using IPv6 [2]. The current Bluetooth standard, Bluetooth 5.0 released in 2016, is focusing on IoT (Bluetooth Low Energy) and has improved almost everything compared to its predecessors (speed, range, IoT connectivity, etc.).

[1] For more details, see CVE-2018-5383 (`https://nvd.nist.gov/vuln/detail/CVE-2018-5383`).

Bluetooth has its own stack from the physical layer to the application layer and does not rely on the TCP/IP stack. The lower layers are implemented on the Bluetooth chipset, and the upper one are implemented on the host device. As highlighted by Armis, it means that the Bluetooth stack is the same in a given OS and is not dependant of the hardware [9]. That is why a vulnerability found in a specific stack applies in reality to a wide range of devices.

This paragraph will present some of the protocols of the Bluetooth stack which are involved in the BlueBorne vulnerabilities disclosure. First, the Logical Link Control and Adaptation Layer Protocol (L2CAP) is part of the lower protocols in the stack and can be seen as an equivalent of TCP for the Bluetooth stack. One of the BlueZ (the Linux Bluetooth stack) vulnerability lies in the implementation of that protocol. The Service Discovery Protocol (SDP) enables the discovery of available services provided by other devices in a given operation range using a request-response scheme. A fragmentation mechanism is implemented in SDP to handle SDP responses (called *SDP continuation*). However, one implementation detail, the structure holding the continuation state, is left to vendors [5] and vulnerabilities have been discovered in both BlueZ and BlueDroid (the Android Bluetooth stack) implementations. These vulnerabilities lead to an information leak that can be used to get encryption keys or parts of memory, which can then be used to bypass the Address Space Layout Randomization (ASLR)[2]. Finally, the Bluetooth Network Encapsulation Protocol (BNEP) is the protocol that allows IP encapsulation over Bluetooth and for instance internet connection sharing. BlueDroid vulnerabilities lie in that protocol and will be detailed in Section 2.2.

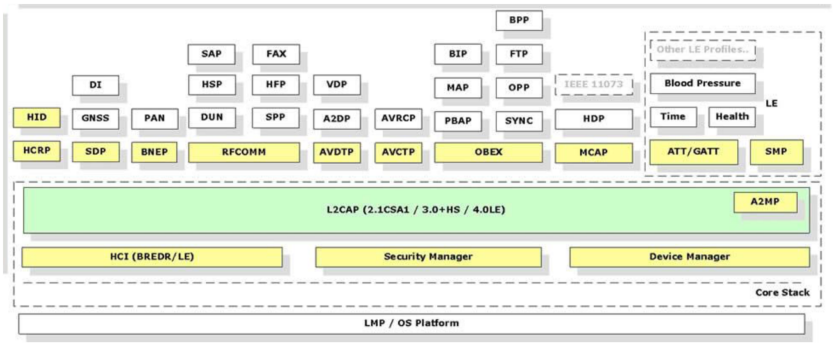

Figure 1: The Bluetooth Stack Architecture. Source: [9]

Multiple vulnerabilities have affected the Bluetooth technology over time, exploiting weaknesses in either the Bluetooth architecture or implementations. As pointed out by Becker et al. in their review of Bluetooth security in 2007, attackers have multiple attack vectors, leading to Deny of Service (DoS), lack of confidentiality and even remote code execution [1].

2.2 BlueBorne vulnerabilities

BlueBorne is a set of 8 vulnerabilities disclosed in 2017 and identified by Armis Lab. BlueBorne affected Linux, Android and Apple devices. However, some devices have not been upgraded by the vendors and might still be vulnerable today.This section presents a full compromission scenario on an Android device running Android 7.1, based on the proof of concept released by Armis Lab [10]. The exploitation of BlueBorne vulnerabilities on Android relies on two CVE. First, the CVE-2017-0785 allows an attacker to bypass the ASLR and is affecting Android 7.*, up to 7.1.2 [7]. Secondly, the CVE-2017-0781 is a Remote Code Execution (RCE) in the

[2]Thanks to ASLR, the address space areas of a process are randomly arranged which prevent an attacker to reliably jump where he wants in the memory.

implementation of BNEP in the BlueDroid stack [6]. As discussed in Section 2.1, the BNEP protocol can be seen as a simpler version of Ethernet and is the one that allows the encapsulation of IP packets over Bluetooth. In this protocol, multiple connection attempts can happened at the same time. New incoming requests are stored to be parsed later when the current connection attempt is established. However, a buffer overflow exists in the code doing the memory copy[3] and allows an attacker to overflow on the heap. This can be followed by a buffer of a size controlled by the attacker which makes the exploitation easier. This vulnerability, used along with the information disclosure due to SDP (CVE-2017-0785), gives the attacker the possibility to execute code in the context of the Bluetooth service on Android. This service has high privileges such as accessing the file system or reading the SMS but also accessing the network.[4] That means the new device can be used to compromise other devices.

3 Problem Discussion

BlueBorne vulnerabilities are dangerous as they have a lot of targets, from Linux to iOS, as well as Android. Moreover, there is no way for the user to prevent himself from being attacked, except by disabling Bluetooth. Indeed, the only requirements for this attack to work is to have the Bluetooth Device Address (BDADDR). As pointed out by Armis, the error leading to the RCE "causes a heap corruption every time the code is triggered" [10] and should have been detected. They suggest that such a mistake remained undetected because the code might not be called in a real-world usage. Vulnerabilities in the Bluetooth stacks might be explained by different reasons. One is that the Bluetooth specification is really big: 2822 pages for the core v5.0 [5]. Also, some part of the implementation is left to the vendors, which makes easier implementation error (such as the one that lead to the information disclosure vulnerability in the CVE-2017-0785). This specification is also complicated and introduces redundancy in the functionalities, which sometimes require a lot of effort to be implemented without bug. For instance, a fragmentation mechanism is available in SDP without a real justification. All that redundancy and difficulty along with the huge amount of code in the Bluetooth stacks makes it really hard for security researchers to audit it in an exhaustive way. Also, it is generally admitted that attacks against Bluetooth are hard to achieve because they often require a BDADDR which is hidden and not available directly if the device is in non-discoverable mode. However, as pointed out by Seri et al., it is quite easy to obtain it using open tools like Ubertooth[5]. Even easier, sniffing the Wi-Fi traffic (where the MAC address is available in cleartext) can leak the BDADDR as it is often the same or differ from one [9]. When using Bluetooth it is now recommended to follow some guidelines [8] provided by organisations such as NIST.

4 Conclusion

Bluetooth is used by a lot of people today and will probably be used even more in the years to come with the grow of the IoT. The development of Bluetooth Low Energy put a new light on Bluetooth and encouraged researchers to dig into Bluetooth implementations. As explained in the Armis white paper, the Bluetooth standard represents a huge amount of code to be audited and their work might only be the beginning of finding new vulnerabilities in the Bluetooth stacks. The new vulnerabilities[6] they disclosed on the 1st of November this year, *BleedingBit* is an example that confirms their thought.

[3] The source code is available in [3]

[4] The full list of permissions can be found at [4]

[5] `https://github.com/greatscottgadgets/ubertooth`

[6] See CVE-2018-7080 and CVE-2018-16986 for more details

References

[1] Andreas Becker and Ing Christof Paar. Bluetooth security & hacks. *Ruhr-Universität Bochum*, 2007.

[2] Nordic Semi Conductor. A short history of Bluetooth https://www.nordicsemi.com/eng/News/ULP-Wireless-Update/A-short-history-of-Bluetooth. Accessed: 2018-11-22.

[3] Google. Coding error in the BNEP implementation in Android 7.1 https://android.googlesource.com/platform/system/bt/+/android-7.1.1_r44/stack/bnep/bnep_main.c#579. Accessed: 2018-11-5.

[4] Google. Permissions of the Bluetooth service on android 7.1 https://android.googlesource.com/platform/packages/apps/Bluetooth/+/android-cts-7.1_r22/AndroidManifest.xml. Accessed: 2018-11-5.

[5] Bluetooth SIG Working Groups. Bluetooth Core v5.0 specification https://www.bluetooth.com/specifications/bluetooth-core-specification. Accessed: 2018-11-5.

[6] NIST. CVE-2017-0781. https://nvd.nist.gov/vuln/detail/CVE-2017-0781. Accessed: 2018-11-5.

[7] NIST. CVE-2017-0785. https://nvd.nist.gov/vuln/detail/CVE-2017-0785. Accessed: 2018-11-5.

[8] Padgette Smithbey Bahr Chen Batra Scarfone and Holtmann. *Guide to Bluetooth Security: Recommendations of the National Institute of Standards and Technology (Special Publication 800-121 Revision 2)*. NIST, USA, 2017.

[9] Ben Seri and Gregory Vishnepolsky. BlueBorne - The dangers of Bluetooth implementations: Unveiling zero day vulnerabilities and security flaws in modern Bluetooth stacks. Technical report, Armis, 2017.

[10] Ben Seri and Gregory Vishnepolsky. BlueBorne on Android - Exploiting an RCE Over the Air. Technical report, Armis, 2017.

[11] Wikipedia. Bluetooth. https://en.wikipedia.org/wiki/Bluetooth. Accessed: 2018-11-5.