The IoT BattLE

Authors
Jennifer Janesko, MSc (Royal Holloway, 2017)
Jorge Blasco Alis, ISG, Royal Holloway

Abstract

Bluetooth Low Energy (BLE) is a wireless protocol designed to consume very little power. BLE networking is somewhat different to other known protocols built on top of TCP/IP. Today, BLE is implemented in many devices that require networking capabilities but are very constrained in energy consumption. The most well-known example of BLE-enabled devices are fitness trackers. These use their BLE interface to send fitness activity related data to a BLE-enabled smartphone. However, BLE is also being implemented in more sensitive devices like baby monitors, smart-locks, biometric authentication systems and health management devices. It is therefore necessary to understand the security implications and risks of using BLE as a means to communicate with other devices. This article provides a set of security guidelines, tools and considerations for anyone within an organization who is considering acquiring or implementing BLE-enabled devices. Our guidelines cover a wide range of responsibilities, from the product manager to the most security-related, the penetration tester. [a]

[a]This article is published online by Computer Weekly as part of the 2018 Royal Holloway information security thesis series `https://www.computerweekly.com/ehandbook/Bluetooth-Low-Energy-The-IoT-battle`. It is based on an MSc dissertation written as part of the MSc in Information Security at the ISG, Royal Holloway, University of London. The full thesis is published on the ISG's website at `https://www.royalholloway.ac.uk/research-and-teaching/departments-and-schools/information-security/research/explore-our-research/isg-technical-reports/`.

Every day a new family of devices joins the (in)famous Internet of Things ecosystem. Refrigerators, baby monitors, vacuum cleaners and even sex toys can now be controlled through a smartphone or the cloud. As the adoption of these devices increase, concerns about the consequences of the widespread usage of interconnected devices with poor security are also growing. Trying to avoid the mistakes made during previous technological developments, both governments and industry are stepping up efforts to provide guidelines for manufacturers to provide safe and secure IoT products.

From the name alone, it is clear that IoT devices are connected to the Internet. This, however, is usually not the only network connection an IoT device has. Devices often communicate with each other inside of a local network using technologies such as WLAN, NFC, Bluetooth Classic or Bluetooth Low Energy.

Bluetooth Low Energy (BLE or Bluetooth Smart) is a special form of Bluetooth introduced in the full Bluetooth 4.0 specification in 2010. Bluetooth Classic and Bluetooth Low Energy are similar in that they enable the establishment of wireless, ad-hoc networks. But, they differ in several conceptual and physical ways. For example, the network stacks for each protocol differ, and they use different physical radio channels.

	Bluetooth Classic	Bluetooth Low Energy
Frequency	2.4 GHz	2.4 GHz
Channels	79	40
Topmost Protocol	Multiple profiles for audio, network, and other kinds data transmission	GATT
Energy Consumption	1 W	Up to 0.5 W

BLE was designed to be a reliable form of wireless communication that consumes very little power, and since its introduction, BLE use has rapidly expanded because it has been integrated into most mobile phones released in the past two years.

Examples of popular BLE products include location beacons, fitness trackers and smart light bulbs. Rather swiftly this technology has moved into more security-critical applications such as smart locks, biometric authentication applications and health management devices including insulin pumps. With the latest revision, the Bluetooth SIG has introduced a BLE mesh networking profile which could be attractive in environmental control applications.

Initially, BLE security concerns focused mainly on privacy, which was fitting for the types of applications available. As the use of BLE expands into new product domains, so must its security scope. Issues of confidentiality, integrity and authenticity need to be evaluated. Further, it should be assessed if misuse could lead to the destruction of property and/or harm to health and safety.

One common way to provide assurance of application security is to perform a penetration test. In a penetration test, a security analyst simulates being an attacker. By doing this, he or she attempts to find the different ways that a device or an application can be exploited. So, what does a security analyst need to know to get started on BLE pen testing? Further, what does a product manager of a BLE application need to know to get the most significant return from a pentest?

Product manager: know thy attack surface

Bluetooth Low Energy is meant to be used for exchanges of relatively small pieces of information like commands or measurement values. An example of this kind of application can be seen in smartphones when they receive heart rate data from a fitness tracker, send a command to open a smart lock or interact with an insulin pump. BLE may also be used in places you don't expect. For instance, some smart meters use BLE to send the readings to the user display installed in their home. In the same way, some industrial control systems rely on BLE to enable wireless control and monitoring.

Developing or purchasing BLE-enabled devices can have a positive impact on an organisation's business operations. The use of BLE, however, will likely introduce new risks to the operating environment. Understanding the attack surface of BLE-enabled devices can help define these risks so that appropriate measures can be selected for risk mitigation.

Attack surface

The different points of entry by which an adversary can compromise target devices.

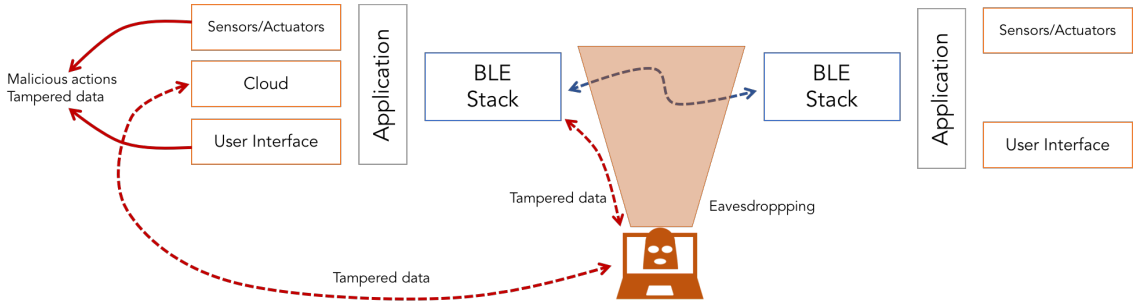
An attack surface describes the different points of entry by which an adversary can compromise target devices. While the over-the-air communication between two BLE devices is obviously an attack vector, there are other options to consider as well. By mapping the entire attack surface, it is possible to visualise what is needed for a penetration test. The attack surface has implications for test environment setup, necessary test equipment and a penetration tester's skillset.

BLE facilitates communication between, at the minimum, two devices. On both sides of this communication, the devices are independent systems that take in application data from various sources, process that data and store or forward that data. Each device will have its own operating environment, physical configuration, application(s) and user interface. If an attacker can control any of these factors, it can influence confidentiality, integrity or availability of a system and its information (Figure 1). This, in turn, can result in a loss of privacy, loss of intellectual property, threats to health and safety and/or destruction of production or property.

The most obvious BLE attack vector is via the over-the-air communication itself. If the appropriate security levels are not configured, an attacker could eavesdrop, relay and/or replay all the communications exchanged through BLE. Further, an attacker could manually connect to the device, read data or issue malicious commands. These actions could compromise the privacy of the owner and could also directly affect the physical world. Going back to our previous examples, an attacker could send a malicious message to an insulin pump which could lead to tragic consequences for the owner.

The next possible attack vectors can be found in the different interfaces by which the BLE applications collect data for processing and communication on the over-the-air interface. Input possibilities include user interfaces, sensors and data flows from upstream applications. Manipulation of the data collected

Figure 1: BLE communications



by a BLE application can lead to exploitation of the receiving devices and downstream services. For example, forged data with malicious code sent via BLE over-the-air could be forwarded to a cloud service and result in the theft of private data stored in the downstream system.

Another attack vector is the device's physical operating environment. The device itself could be modified, its power source could be compromised or the underlying OS could be configured to modify BLE settings. If an attacker is able to influence any of these factors, it could lead to attacks such as denial of service or even complete system takeover.

It is important from an attack surface perspective to keep in mind where evidence of an attack can be detected. Evidence can be present in the devices themselves, and it can also surface in downstream applications. Downstream applications are often vulnerable because they tend to trust that data is transferred from a "trustworthy" source, i.e., the BLE device. It is important to know the flow of data from one system to another to be able to identify which downstream systems are needed for a full-scale penetration test.

Product manager: recruit your army

Identifying a penetration testing partner for a BLE product is not a simple task. There are two main factors standing in the way of finding a testing team with suitable skills. The first is that changes to the specification are being rolled out rapidly, and appropriate security testing tools lag behind. The second is that there is a dearth of penetration testers who understand the full intricacies of the complicated BLE specification. A product manager must develop a strategy for finding a penetration testing partner that can handle the product's BLE version and competently cover the defined attack surface.

First, a partner that can demonstrate experience in testing BLE products should be sought. There are a variety of ways to achieve this. Has a potential service provider published work in the area of BLE? Can the service provider provide references? In response to an RFP for penetration testing of a BLE product, does the service provider ask pertinent questions about the product's BLE version, which Bluetooth profiles have been implemented and which types of pairing, if any, are being used?

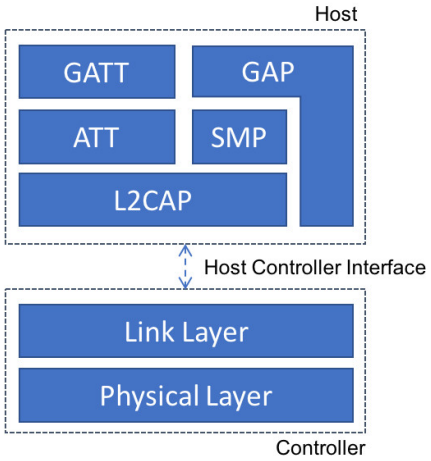
Second, the product's BLE version will play a role in selecting a partner. For versions 4.0 and 4.1, it will be easier to find competent testers. Basic testing tools have been available for over a year for these versions. For versions 4.2 and 5.x and mesh profile implementations, the service provider will need to have developed in-house tools for testing and will require more specialized knowledge than an average pentester. This situation, of course, will change as time goes by and new testing software and hardware become available.

Security analyst: learn the rules of engagement

To perform a competent pentest, the security analyst needs a fundamental understanding of the BLE specification.

Bluetooth firmware is often distributed on two different components. The first component, called the host, stores and manages the data for the Bluetooth application. The second component, called the controller, manages the transmission and reception of over the air data packets and payload encryption. These two components usually communicate between each other over a host controller interface (HCI) pathway. The BLE stack can be broken down into seven layers and are distributed over the host and controller (Figure 2).

Figure 2: BLE stack



The smallest piece of data managed is called an “attribute”. The Attribute Protocol (ATT) and Generic Attribute Protocol (GATT) define how attributes can be combined to provide services and data characteristics that can be accessed, read, manipulated and subscribed to. The Security Manager Protocol (SMP) layer defines security operations that can be performed to protect the BLE application. The Generic Access Profile (GAP) layer provides the fundamental modes and procedures by which communication can occur over the air between two BLE devices. And the Logical Link Control and Adaptation Layer Protocol (L2CAP) prepares data packets for communications to and from the host over the HCI.

BLE communication can almost be broken down into two protocols: an advertising protocol and a data protocol. In the first versions of the specification, advertising takes place on three predefined radio channels. Data communication takes place over 37 additional radio channels.

BLE stack terminology

HCI: Host Controller Interface.
ATT: Attribute Protocol.
GATT: Generic Attribute Protocol.
SMP: Security Manager Protocol.
GAP: Generic Access Profile.
L2CAP: Logical Link Control and Adaptation Layer Protocol.

On the advertising channel there are two types of devices: advertisers (or broadcasters, peripherals) and scanners/initiators (or observers, centrals). Advertising devices broadcast data to scanners and indicate whether or not they are accepting connections. Scanners listen for broadcasted messages and either ignore the broadcasts, use the received data, send scan requests for additional information, or attempt to setup a connection with the advertising device. On the advertising channels, communication is one-to-many.

A scanning device can move to an initiating state by sending a connection request to an advertiser.

If an advertiser receives a connection request and accepts that request, it will send a response to the initiator to establish the connection. From the moment that the advertiser sends this response, it moves to a connected state and is then referred to as a slave. When the initiator receives confirmation from the slave, it moves to a connected state and is referred to as the master.

From this point on, communication will occur over the connection between only those two devices on the 37 data channels. Frequency hopping is used to prevent interference, and the order of the channel hopping is negotiated at the time that the connection is established.

One of the mechanisms that was introduced in the 4.0 specification to help preserve confidentiality and authenticity of the over-the-air communication was "pairing" and "bonding". Pairing is the act of exchange in cryptographic information to establish confidentiality and authenticity. There are three types of pairing: just works, PIN entry, and out of band. Bonding takes this one step further and stores the information long-term so that two devices do not have to pair every single time they establish a connection.

It was reported in 2013 by Mike Ryan that the cryptographic key exchange protocol in the 4.0 and 4.1 specifications was weak. If an attacker is able to eavesdrop on the initial connection establishment procedure and initial pairing, it would be trivial to brute force the key used to derive the other keys used for various types of security goals.

In the 4.2 specification, a new type of pairing was introduced called LE secure. Instead of using a proprietary key exchange protocol, it uses a different key exchange based on elliptic curve cryptography. In addition to this, the specification also included the provision for allowing cryptographic operations to occur in the host as opposed to occurring in the controller.

The 5.0 specification was released in 2016 and expanded the advertising part of the protocol. Instead of only using the three channels for advertising, it also allows advertising to occur on the data channels where larger amounts of data can be exchanged.

Mesh networking is the latest major addition for BLE. It is technically not part of the Bluetooth specification, but instead it has been introduced as a "profile" that is reliant on the 5.0 specification. Bluetooth profiles describe services and requirements for a common functionality that BLE devices can perform. The mesh profile provides the capacity for many-to-many communication within a BLE network and additional cryptographic measures for security.

An analyst performing a security review of a BLE application must be familiar with the evaluation target's BLE specification version and whether or not mesh networking is being used. These variables will have an impact on the availability of tools and the types of evaluations that need to be performed. There are inherent weaknesses in some parts of the specifications themselves, and some parts of the specifications could be poorly implemented. An analyst should be aware of the common weak points be prepared to test them.

Security analyst: choose your weaponry

Of course, knowing the specification inside and out is not enough to run a pentest. An analyst must have the right tools for the attack. Most available tools today concentrate on the initial versions of the protocol (4.0 and 4.1), and may not work against newer versions of the protocol. This has inherent implications for the security analyst.

Most available devices today use BLE 4.0 and 4.1 for which regular off-the-shelf dongles are effective. These can be combined with a Linux operating system, the BlueZ[1] stack, and Linux programs such as `gatttool` and `hcitool`. In addition there are two node.js packages, `noble`[2] and `bleno`[3] that allow an analyst to emulate a client and a server. `Noble` and `bleno` also run on OSX and Windows.

Most mobile phones support BLE. Both Android and iOS allow Bluetooth logs to be activated. These

[1] `http://www.bluez.org`
[2] `https://github.com/sandeepmistry/noble`
[3] `https://github.com/sandeepmistry/bleno`

logs can be useful for mapping communication exchanges supported by BLE applications. In addition to this, apps such as RaMBLE[4] and NRf Connect[5] can also be useful in an analysis.

A handful of security tools have been built for BLE generations 4.0 and 4.1. A sniffer called ubertooth can be used to capture traffic and break LE Legacy pairing. Two man-in-the-middle frameworks have been released in the community called gattacker[6] and Btlejuice[7]. To make it easier to inspect and interrogate a slave device, the tool BLEAH[8] was recently released. A security analyst looking to verify the security properties of the attributes of BLE devices can also use att-profiler[9].

Up to this point, most of the tools that have been introduced test the BLE application layer. Analysing the protocol layer will require more resources, as there are no available off-the-shelf tools. Scapy[10] includes a Bluetooth Low Energy library that allows the manipulation of HCI, and hence the BLE protocol layers below the application. This functionality, however, is undocumented, and more time would be required to develop testing techniques using this tool. There are proprietary fuzzers, such as Defensics, but unless BLE is the main type of testing that an analyst performs, it can be cost-prohibitive.

The situation becomes worse when getting into the most new protocol versions (4.2 and 5.0), as some of their features are not implemented in any security analysis tool. Development kits from either Nordic or Texas instruments could be acquired for this purpose. Testing for these BLE versions will take more time because test tools would still need to be developed on top of the development kits.

BLE: A call to arms

Bluetooth Low Energy is now a common communication interface available in many smart devices. Although the protocol specification is rapidly changing, most of the currently available devices run versions 4.0 and 4.1 which were released before 2012. Fortunately, most of the currently available tools are focused on these versions of the protocol, enabling pentesters to scrutinise the security of BLE devices at the application level. However, the future may hold a worrying prospect. Devices implementing Bluetooth 4.2 are starting to be more predominant and the recent release of the 5.0 version will introduce a new wave of devices for which there are no publicly available tools for security analysis.

At the moment, those tools may be privately developed via specific providers, but the community is missing a set of open and properly documented tools to perform security analysis, as opposed to other areas like web or mobile application analysis. Despite the amount of information transmitted through BLE and the limited availability of tools for its analysis, security researchers have been able to uncover vulnerabilities that, for example, allowed to control, without authorization, a hoverboard[11], even while in use.

More effort is needed to cover the missing parts of the attack surface and also the new specifications, so the penetration testers can efficiently analyse the BLE-enabled devices of the future. Of course, new tools won't be enough if the penetration testers are not properly trained. With the range and number of devices that will be around, it is paramount that we also prepare our penetration testers to analyse less common communication interfaces. As the good penetration tester know, all input is evil, even the one coming from a Low Energy interface.

Biographies

Jennifer Janesko is a security consultant for a mid-sized, German firm. Her professional areas of focus are security testing and secure design of critical infrastructure networks. Prior to her current

[4] https://www.contextis.com/resources/tools/ramble-ble-app

[5] https://www.nordicsemi.com/eng/Products/Nordic-mobile-Apps/nRF-Connect-for-mobile-previously-called-nRF-Master-Control-Panel

[6] https://github.com/securing/gattacker

[7] https://github.com/DigitalSecurity/btlejuice

[8] https://github.com/evilsocket/bleah

[9] https://github.com/projectbtle/att-profiler

[10] https://github.com/secdev/scapy

[11] https://www.ioactive.com/news-events/ioactive-finds-critical-security-vulnerabilities-in-segway-inebot-minipro-hoverboard.html

position, she worked in the IT sector for over 15 years in the areas of education, telecommunications and semiconductors. Jennifer earned an MSc from Royal Holloway, University of London. Her master's project focused on the development of a security testing framework for Bluetooth low energy devices. In addition to her MSc, Jennifer has a masters in information management from Washington University, a master's certificate in instructional technology from the University of Massachusetts, Boston, an M.A. in philosophy from Kent State University, and is a certified GICSP. In her spare time, Jennifer enjoys running, climbing and tinkering with technologies.

Jorge Blasco obtained his PhD from University Carlos III of Madrid in 2012. His dissertation was focused in the field of information security. After obtaining his PhD, Jorge worked as an assistant lecturer in University Carlos III of Madrid. In 2014, he moved to City, University of London, where he worked until 2016 as a Research Fellow in a project about application collusion. His main research interests include mobile malware, steganography and wearable devices. Since September 2016, Jorge Blasco is a Lecturer and MSc in Information Security Course Director in the Information Security Group at Royal Holloway, University of London.

Series editor: S.- L. Ng