Security in Bluetooth

By Nicki Thomson

1. Abstract

Everyday people go about their lives, often carrying one or more Bluetooth enabled devices. With the expansion of the Internet of Things, more and more gadgets are broadcasting their existence to their surroundings. This paper will explore how bluetooth connections can be sniffed, what implications that has for the many growing number of Bluetooth Low Energy connected devices, and what steps can be taken to increase security.

2. Introduction

One of the biggest topics in tech news right now is the Internet of Things (IoT). While the frequently heard prediction of of 50 billion connected devices by 2020 has been lowered to somewhere between 28 billion and 30.7 billion [3], it is evident that there are a lot of devices generating and exchanging information in the world today. Much of this information is making its way between devices through Bluetooth Low Energy (BLE).

BLE is the communication protocol of choice for most IoT devices. Recently Apple even released the iPhone 7 which does not include a headphone jack, instead forcing users to instead use bluetooth connected speakers and headphones.

This paper explores how BLE works is and why security in BLE is relevant, how sniffing bluetooth connections is possible, and what actions can be taken to defend against this sniffing. The rest of the paper proceeds as follows:

- Security Relevance
- How BLE Works
- Sniffing a Bluetooth connection
- Action Items
- Conclusion

3. Security Relevance

As the Internet of Things expands, a lot of information becomes available to hackers. Much of the copious amounts of data generated by IoT gadgets may seem insignificant and therefore not in need of protection. Consider, however, that researchers have shown it is possible to determine a victim's ATM pin using the accelerometer and gyroscope data from the victim's activity tracker [8]. Another paper [6] noted the privacy implications of intercepting a bluetooth connection between a phone and a computer to sync address book information. With the ability

to sniff any BLE connection, all information in flight between the connected devices can be determined.

Bluetooth is also expanding into medical devices such as biosensors [4]. With the ability to both sniff the connection, and inject packets, attackers now have the opportunity to inject false bio sensor information, possibly jeopardizing the victim's health. Because of the prevalence of bluetooth in today's devices, it is important to understand some of BLEs known vulnerabilities so we can work to fix them in the future.

4. How BLE Works

4.1 Master Slave Relationship

Bluetooth Low Energy connections operate with a master slave relationship (also referred to as a central peripheral relationship). A central device can accept multiple peripherals, and it is even possible for a peripheral to be connected to multiple centrals. The other mode of operation is where one device broadcasts, and observer devices can all get the broadcast data, this paper will be focusing on bluetooth connections.

In bluetooth there are two packet types, data and advertising. When a peripheral device wants to pair with a central, the peripheral device sends out an advertising packet stating this intention. A central can then scan for peripheral devices and send a request to the peripheral to make an exclusive connection. If the peripheral responds then a connection is established [7].

4.2 Packet Layout

Like the network stack, BLE also has a protocol stack. At it's highest level the BLE protocol stack can be divided into the application layer, the host layer and the controller layer. The controller layer can be broken down into the Host Controller Interface, the Link Layer, and the Physical Layer. The Link Layer functionality usually includes, cyclic redundancy check (CRC) generation and verification, data whitening, random number generation and AES encryption [7].

5. Sniffing a BLE Connection

5.1 Bluetooth Security Measures

This section lays out some of the security measures and difficulties that come with attempting to sniff a bluetooth connection, and what researchers have found to bypass these obstacles. The topics covered in this section are channel hopping, data whitening, and encryption. Channel hopping is an implicit function of bluetooth, where the central and peripheral devices change the channel they are communicating on for every packet sent. Even if there is no data to be exchanged an empty packet is still sent on each channel [5]. Data whitening is the scrambling of

bluetooth packets based on a sequence determined by the last six digits of the connected devices clocks. Even plain text packets go through data whitening in bluetooth [6]. Finally BLE uses 128 bit AES encryption, however the key exchange for this encryption has very little security.

5.2 Channel Hopping

The master slave connection in bluetooth is maintained over an alternating series of channels, which the central and peripheral devices synchronously hop between. Bluetooth has 37 possible channels, and most connections use all 37. The channels accessed follow a predefined pattern, as explained in [5]. The pattern is:

$$nextChannel \equiv channel + hopIncrement \pmod{37} \quad (eq\ 1)$$

However the time spent on each channel varies between connections. By observing a single channel, the time it takes for a cycle of 37 hops to complete can be found. By dividing this time by 37, a hacker can find the time spent on a single channel:

$$t\_perChannel = t\_fullCycle/37 \quad (eq\ 2)$$

Once the time spent on a single channel is known, the hopIncrement can be determined by observing two separate channels. Using the time between two channels and the time per channel, the number of hops between those two channels can be determined:

$$numHopsBetween = delta\_t / t\_perChannel \quad (eq\ 3)$$

And thus the hopIncrement can be derived by finding the pattern that would result in *numHopsBetween* for the two chosen channels. From here anyone who wishes to sniff the bluetooth connection simply needs to follow this same channel switching pattern as the communicating devices [5].

5.3 Removing Data Whitening

All signals sent over bluetooth, even those sent in plain text go through data whitening. This data whitening essentially scrambles the bluetooth data. This scrambling is based off the last six digits of the clock, which is only known by the devices in communication. The six clock bits are used as the input to a linear feedback shift register to make a pseudo random sequence. The generated pseudo random sequence is XORed with the data, and the data is sent. Because only six digits are used as input, there are only 64 possible inputs which is easy to brute force. The 64 brute forced pseudo random sequences can be XORed with the sniffed data, and the correct sequence can be determined by comparing the linkID or the cyclic redundancy check (CRC). Once the correct clock digits have been found, anyone sniffing the network can stay synchronized with the communicating devices [6].

5.5 Breaking BLE encryption

BLE boasts 128-bit AES data encryption. However the process of the long term key (LTK) exchange between devices is much less secure. In addition, on connections where a LTK has already been established, an attacker can inject a link layer message (LL_REJECT_IND) which exists in BLE protocol for cases where a new LTK needs to be established (for example when a

peripheral loses it's memory). When a new LTK is established, the devices start by establishing a temporary key (TK). The TK is used to as the devices select a short term key (STK), which in turn is used in communications for determining the LTK. When the TK is generated, the key is based on the protocol in use, and a confirm of the TK is sent in plain text. In one protocol the TK is a constant value, and in another the TK is a six digit pin, so it is trivial to brute force all the possible confirms and find the matching one. Once the TK is known this can be used to decrypt the STK which can be used to decrypt the LTK. Once an attacker knows the LTK, encryption becomes useless [5].

6. Action Items

With the knowledge of some of Bluetooth Low Energy's vulnerabilities, the question turns to what can be done to protect users' privacy. The possibilities covered here are using an out of band (OOB) key exchange, editing the central bluetooth controllers to vary the channel hopping pattern, and choosing what devices make sense to use.

6.1 OOB Key Exchange

In [5] where the BLE key exchange is shown to be insecure, it is mentioned that some bluetooth devices have an OOB key exchange to create the temporary key. This is a 128 bit value, and so a well chosen OOB key can not be broken quickly. In future, all devices which carry highly sensitive bluetooth data (such as medical devices) should have some way to exchange keys out of band.

6.2 Varying Channel Hopping

In [1] a counter measure to blindly following the channel hopping is suggested. The counter measure works off the knowledge that at least 20 good channels must used for Bluetooth channel hopping, however not all 37 channels must be used. [1] suggests editing the master so that out of the channels which are actually good, the master occasionally flips which channels are labeled as good for use. By doing this the channels in use are consistently changing, and hence cannot be as easily followed with the initial channel hopping pattern. For devices which might benefit from some additional security, but for which security is not enough of a priority to require encryption with an OOB key exchange, this variable channel hopping may be a good solution.

6.3 Choosing Devices Wisely

Finally, as more and more bluetooth enabled gadgets reach the market the decision falls upon users to decide what is worth sacrificing some information to hackers and what isn't. While the work done to steal ATM pins from victim's activity trackers in [8] is a good shock story, there are so many easier ways to steal a victim's ATM pin that perhaps the benefits that come with wearing an activity tracker outweigh the risks.

7. Conclusion

It’s been found that many bluetooth connections can be broken through a combination of brute force and careful observation. With that knowledge in hand, it is up to developers to keep the tenuous nature of BLE security in mind when creating more and more gadgets for the Internet of Things.

8. Works Cited

1. Albazrqaoe, Wahhab, Jun Huang, and Guoliang Xing. "Practical Bluetooth Traffic Sniffing." Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys '16 (2016): n. pag. Web. 13 Dec. 2016.

2. Lindell, Andrew Y. "Attacks on the Pairing Protocol of Bluetooth V2.1." Blackhat Briefings (2008): n. pag. Web. 13 Dec. 2016.

3. Nordrum, Amy. "Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated." IEEE Spectrum: Technology, Engineering, and Science News. N.p., 18 Aug. 2016. Web. 13 Dec. 2016.

4. Perry, Tekla S. "A Temporary Tattoo That Senses Through Your Skin." IEEE Spectrum: Technology, Engineering, and Science News. N.p., 29 May 2015. Web. 13 Dec. 2016.

5. Ryan, Mike. "Bluetooth: With Low Energy Comes Low Security." USENIX WOOT (2013): n. pag. Web. 12 Dec. 2016.

6. Spill, Dominic, and Andrea Bittau. "BlueSniff: Eve Meets Alice and Bluetooth." USENIX WOOT (2007): n. pag. Web. 12 Dec. 2016.

7. Townsend, Kevin, Robert Davidson, Akiba, and Carles CufiÌ. Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-power Networking. Sebastopol, CA: O'Reilly, 2014. Print.

8. Wang, Chen, Xiaonan Guo, Yan Wang, Yingying Chen, and Bo Liu. "Friend or Foe?" Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security - ASIA CCS '16 (2016): n. pag. Web. 12 Dec. 2016.