Security Vulnerabilities of Bluetooth Low Energy Technology (BLE)

Harry O'Sullivan
Mentor: Ming Chow

Tufts University

Abstract

The steep rise in wearable technology use has been confirmed by the recent launch of the Apple Watch. Forward-thinking technology companies are increasingly looking towards wearables as an opportunity to improve mobile efficiency, enhance communication, and gain invaluable data about users. However, as these devices grow in popularity, so should concerns over security. Today, millions of mobile and wearable devices communicate via battery-optimized communication technologies like Bluetooth Low Energy. Security protocols on these devices have not yet been designed to prevent hacking if the device is somehow misplaced. Vulnerabilities also exist in how these communicative devices send each other information. This paper will explore and infer the implications of these vulnerabilities, as well as pose simple security measures for the common technology user.

Introduction

Apple's recent decision to research, develop, and globally distribute a "smart" watch has given light to the current surge in wearable technology. Take a fitness bracelet. An object that was once just a simple ornament people wore on their wrists has been transformed into a sophisticated microcomputer equipped with a gyroscopic sensor, an accelerometer, a long-lasting battery, and Bluetooth Low Energy capabilities. All of these tools enable the live-tracking of a person's activity levels, and there is strong belief that sectors like healthcare will benefit greatly from the data these devices are now capable of collecting and distributing. Bluetooth, which was initially intended as a "replacement for cabling"[1], enables these devices to communicate this data through bands to other synchronous devices. These bands are supported by a network of other bluetooth dongles, mobile applications, and cloud services, allowing users to view the data on many possible mediums and locations. What is perhaps interesting to touch upon is the steep rise in peoples' interest in self-quantification. The market for applications that provide this service is indeed expanding, and remote servers around the world that stockpile data pertaining to personal health are reaching numbers far from few.[2] The question that remains, however, is whether all of this information is being communicated safely. The sections to follow will explore potential threats to Bluetooth-capable devices and the relevant implications of these threats, as well as offer guidance as to how the average person can best avoid such attacks. Proposals of how to actually mitigate these vulnerabilities within the Bluetooth source itself are out of scope and will not be discussed.

To the Community

Today, the matter of software security has been overshadowed by an ever-increasing tendency to concentrate solely on creating unforgettable user experiences. As a result, security has lagged behind. Indeed, the main security vulnerabilities of Android and Apple applications result from bad code quality and inadequate error handling.[3] And the data being stored on some of these self-quantifying applications falls nothing short of revealing: total steps taken per day, calorie counts, distances and locations travelled, hours and quality of sleep, blood glucose levels, heart rates, sync times, battery levels, IP Addresses, and the list goes on. Products like *Fitbit*[4] even encourage users to synchronize their accounts with social media sites, creating ever larger data sets on entire

---

[1] Neim, *Bluetooth and Its Inherent Security Issues.*, 1

[2] Tractica, *Enterprise Applications for Wearable Technology are Expanding at a Rapid Pace*

[3] Chow, *Understanding the Threat Profiles of Mobile Apps,* 31 & 32

[4] The company is known for its products of the same name, which are activity trackers, wireless-enabled wearable technology devices.

populations and macro-habits.[5] These previously unseen sets of information can be traversed in ways that reveal much more about an average person than that average person would probably ever assume. Health conditions, socioeconomic status, political tendencies, and even personal habits are but few of the myriad inferences that can be made by algorithms that connect the dots. And while people currently may not fully recognize the value of the data being collected, it's no question that countless data-hungry third-parties do. A recent Federal Trade Commission study found that many popular health and fitness applications actively share "… unique device identifiers, usernames, email addresses, medical symptom searches, gender, dietary habits, exercise activity, zip codes, and geo-locations" with third-party companies.[6] Senator Chuck Schumer has even gone so far as to comment on the issue, saying, "The fact that private health data — rich enough to identify the user's gait — is being sold to third-parties without the user's consent is a true privacy nightmare."[7] The unfortunate truth is that as more valuable data is collected — that is, through bands connected by Bluetooth-communicative devices — the more prone it is to criminal attacks. Being oblivious to these existing vulnerabilities may result in an situation where peoples' ignorance is exploited by those who are determined to compromise privacy.

Existing Security Measures

Advances in wireless sensor networking technologies have been fully extended to wearable computing systems. Bluetooth Low Energy is now the de-facto standard for wireless personal network connectivity. It's protocol supports both audio and data communication and provides built in security measures. One such measure is *Frequency Hopping*, which was put in place to solve the problem of interference from other signals using pseudo-random frequency switching.[8] Here, a channel is used only for a very short period, and after a random number is calculated and agreed upon between two devices, they hop between 79 frequencies at about 1600 times per second.[9] This security measure is an attempt to provide protection against potential eavesdropping. But what cannot be left unmentioned are the number of security modes that a bluetooth device can operate within:

---

[5] This is not to say that data collection on this scale has not been occurring already, but rather that wearable mediums create whole new data categories.

[6] FierceMobileHealthcare.com

[7] venturebeat.com

[8] Radak, *The Security of Bluetooth Systems and Devices*

[9] Ibid.

Mode 1: an insecure mode of operation that provides no available security. A device in this state accepts any connections that are offered to it.

Mode 2: a mode of operation in which the device uses service-level enforced security. When connection is made, a device in this mode lets the application decide how to proceed.

Mode 3: a mode of operation in which the device uses link-level security. A device in this mode will not be able to connect unless permission is validated through a PIN code and a link key.

Out of the three modes, the last one is indubitably the most interesting and seemingly secure. Figure 1[10] shows the pairing process of two devices in Mode 3. First, a user enters a PIN code on both bluetooth devices. The device trying to connect sends a unique 48-bit address, while the device trying to authenticate sends a 32-bit challenge number. Both devices then use the identical link key generated from the PIN to compute an authenticated response. If the responses don't match, connection is refused and blocked for a certain about of time in order to deter attackers from attempting to guess the PIN. Of course, devices can't be paired unless they have successfully authenticated each other at least once — but after merely this one test, the devices are configured to automatically trust each other whenever they attempt to pair in the future. Mode 3 also encrypts all data: the encryption algorithm makes use of a key generated by the Authenticated Cipher Offset[11], which is generated along with the authentication response. It is important to note that the ACO is computed using the same link key, which in turn generates a key to encrypt the data in each packet being transmitted. All of this is computed as a function of the Bluetooth unique device address, which, by the way, is transmitted in cleartext in the primary stage of pairing.

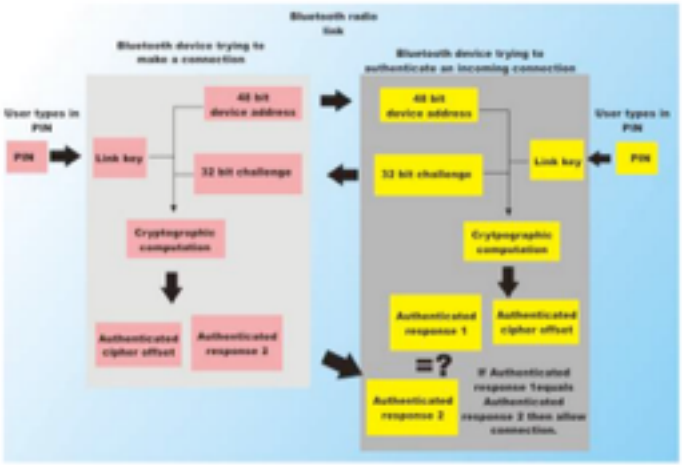

Figure 1: Bluetooth Authentication

[10] www.i-programmer.info, *How Bluetooth Works*

[11] ACO is a deterministic algorithm operating on fixed-length groups of bits, called *blocks*, with an unvarying transformation that is specified by a symmetric key.

Potential Threats

1. Eavesdropping

At first glance, Mode 3 looks adequately secure. Yet with enough time and determination, an attacker can bring to light a number of existing vulnerabilities. Let us first explore the very common scenario of a misplaced device. Recall that none of the default security measures describe user authentication on the device itself, meaning information on a misplaced device has the potential to be freely accessed and used to eavesdrop on further communication between that device and its trusted pairs. Consider a Bluetooth headset, for instance. It is almost certain that a trust relationship already exists between the headset and a mobile phone. An attacker could use the headset unit number for impersonation and pave the way to gain connection to more powerful Bluetooth-enabled devices. Even if the victim were to delete the encryption/decryption keys on the mobile phone, an attacker could still communicate with the phone by using the ID of the device, which could also be used to brute force the PIN number *offline* using a Bluetooth sniffer[12]. Once an attacker arrives at the correct PIN, it is not difficult to derive the the link key and eventually go on to hijack the device.

But what about frequency hopping? Well, just as one could use a Bluetooth sniffer to brute force a PIN, one could just as easily use the same sniffing software to determine the seed of a frequency hopping sequence by using the BD_ADDR[13] and clock of the master device in the piconet[14]. For under $30, *Bluefruit* will scan inquiry frequencies to determine the master device and its BD_ADDR (recall that bluetooth devices send identifying information about themselves in packet[15] headers, in cleartext, as shown in Figure 2). Knowing the BD_ADDR of the victim also allows an attacker to inject carefully crafted packets directly into the victim device.

While on the topic of BD_ADDRs, it is also relevant to mention a study done in 2005 by researchers at MIT who found that BD_ADDRs are *not* globally unique.[16] This presents a huge security vulnerability, for an attacker could compile a list of commonly used BD_ADDRs and use each one (applications like *Spooftooph* allow one to change the BD_ADDR of a computer's Bluetooth module) until packets start to flow in. Knowledge of

---

[12] A *sniffer* is a computer program that detects and records a variety of restricted information.

[13] A *BD_ADDR* is a Bluetooth address, specific to the device in question.

[14] A *piconet* is a computer network which links a wireless user group of devices using Bluetooth technology protocols. A piconet is made up of two or more devices occupying the same physical channel (synchronized to a common clock and hopping sequence).

[15] A *packet* is a block of data transmitted across a network.

[16] Chai, *Hacking Bluetooth,* 13

the BD_ADDR of a nearby device would enable an attacker to implement further packet sniffing or packet injection.

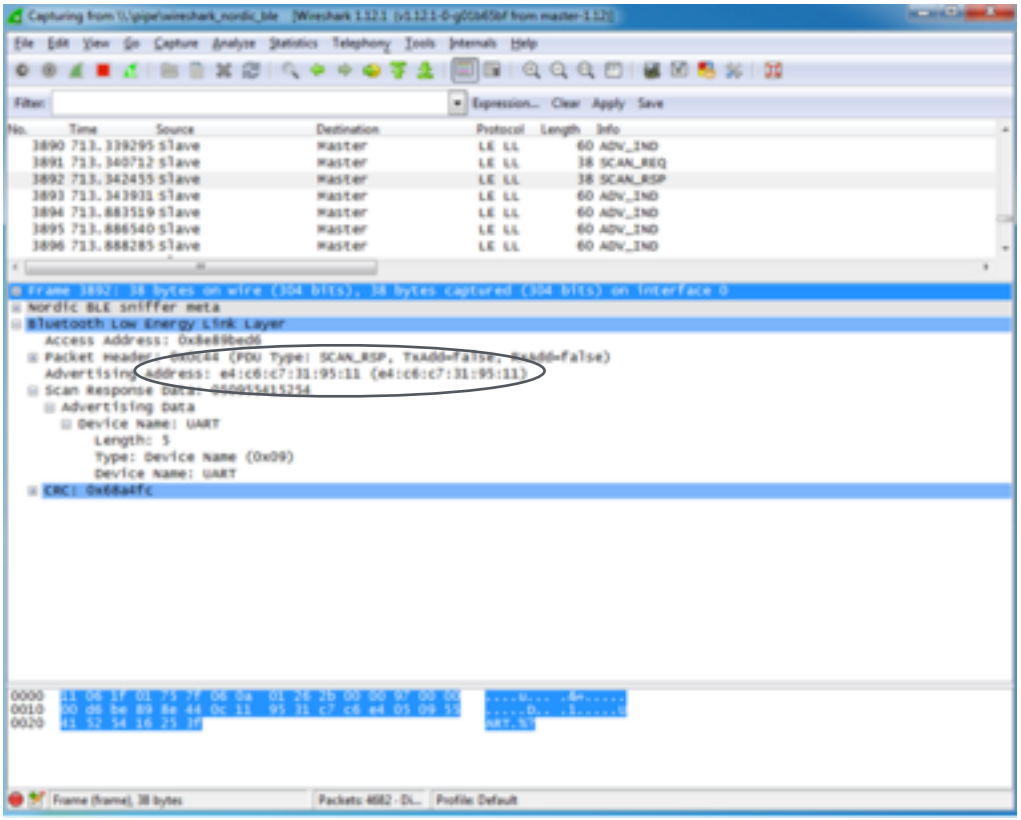

Figure 2: An example of how BD_ADDRs are transmitted in plaintext using a packet-analysis software like *Wireshark.*

2. Man-in-the-Middle Attacks

Bluetooth devices are especially prone to man-in-the-middle attacks. These attacks occur when an attacker secretly relays and possibly alters the communication between two devices that believe are communicating with each other. A 2010 paper by lecturers at the Marthandam College of Engineering and Technology showed that if one could somehow trick the devices to assume that they have been disconnected from each other, then one could use two Bluetooth modules to act as the master and slave devices, which would in turn make possible packet injection[17] and authentication attacks. In fact, they were able to do so by jamming all frequencies.[18] This experiment demonstrated that vulnerabilities need not always be in the the Bluetooth encryption mechanism itself. In this

[17] The process of interfering with an established network connection by constructing packets that appear as if they are part of the normal communication stream.

[18] Saravanan*, A Novel Bluetooth Man In The Middle Attack Based on SSP Using OOB Association Model*., 4

case, a vulnerability was discovered in how devices handle disconnections and reconnections, and by recognizing and exploiting this weakness, the lecturers were able to connect their devices to the victim devices instead of allowing the victim devices to connect to each other.

3. Denial of Service & Fuzzing Attacks

Most wireless devices today operate on built-in battery packs. This dependency makes it hard to ignore the threat of Denial of Service attacks. DoS encapsulates a class of attacks whereby a system providing some service is overwhelmed by malicious requests from an attacker, resulting in the crash of the system and the eventual draining of battery life. A recent study by researchers at the University of Utah showed that "repeated requests make … phone[s] unable to receive any phone calls and the battery life is reduced by as much as 97%." They coined this as a *Battery-Draining-Denial-of-Service Attack,*[19] where, "on average, about 5454 SDP requests were made in a duration of 423 minutes to bring down the phone completely." To mitigate this attack (and those aforementioned), Bluetooth devices now offer 'non-discoverable' mode, where the device will not attempt to respond to any inquiry request messages. However, this does not render the device completely fool-proof. It turns out that so long as the Bluetooth device is on, it will continue to accept packets with appropriately crafted headers that include the correct LAP[20] of the receiving device. In Fuzzing[21] attacks, attackers need only send malformed or otherwise non-standard data to a device's Bluetooth radio and observe how the device reacts. Some inputs can also slow the target device's operation and potentially even crash the system.

Implications & Concluding Thoughts

The potential consequences of eavesdropping are clear: by bypassing default security measures, a catalog of personal information can be exposed, whereby privacy is unwaveringly compromised and there is no knowing where the stolen information will end up or how it will be used. For instance, a simple eavesdropping attack can turn the Bluetooth appliance in a car into a microphone, allowing an attacker to remotely record audio coming from within the vehicle.[22] Thus, it is not too far-fetched to picture having the ability to listen in on, for instance, an executive board meeting without them even knowing. And it turns out

---

[19] Premnath, *Battery-Draining-Denial-of-Service Attack on Bluetooth Devices.,* 1

[20] Stands for *Lower Address Part*, which is a 3-byte word transmitted in every packet as part of the header.

[21] A *Fuzzer* is a program that discovers coding errors and security loopholes in software by inputting random data. Fuzzers work best for problems that can cause a program to crash, such as DoS attacks or buffer overflows. **For this paper, I have written a simple Bluetooth fuzzer in python to act as supporting material.**

[22] This attack is more commonly referred to as "Carwhispering*".*

that Bluetooth-capabilities are proliferating in the office space: companies today are stocked with hands-free headsets, wireless mice, keyboards, PDAs, office mobile phones, and data projectors. In an age where devices with Bluetooth capabilities continue to end up under the scrutiny of millions of smart minds, it is definitely not too far-fetched to imagine corporate environments — where terabytes worth of sensitive data are exchanged daily — falling victim to such attacks.

The consequences of Denial of Service and Fuzzing Attacks, however, may not seem as apparent. Indeed these attacks concentrate more on interrupting a device's hardware rather than stealing the data stored within them. Yet today, a simple DoS or Bluetooth Fuzzer attack could potentially be life threatening. Medical devices like insulin pumps, real-time glucose monitors, and pacemakers have advanced in ways that have allowed them to become small and wearable. Just like fitness bracelets, many of these devices are connected to hand-held controllers using Bluetooth Low Energy technology. Recent studies show that even with simple hardware and a device's PIN number, an attacker can take control of a medical device. At the University of Southern Alabama, students were able to compromise the network protocol of a working pacemaker by performing a Denial of Service attack using HPING3[23].[24] Luckily for them, the pacemaker was placed inside a medical dummy instead of a human, but this does not take away from the main point, that, "If medical training environments are breached, the long term ripple effect on the medical profession, potentially, impacts thousands of lives due to incorrect analysis of life threatening critical data by medical personnel."[25] The FDA's recent stance — that it "[expects] medical device manufacturers to take appropriate steps to protect devices"[26] — is thus hardly surprising.

The threat of Bluetooth-related attacks are evidently real and particularly relevant in today's increasingly connected society. While threats in this area have waned since 2008,[27] the dependency on Bluetooth has not. But after exploring the many possible weaknesses to Bluetooth devices in Mode 3, there is one that has gone amiss: most users, after enabling Bluetooth discoverability, often leave their devices in this configuration for a much longer period than needed. Consequently, there are always devices in public areas that are fully discoverable and exploitable. On some systems like the Apple iPhone, discoverability is rather covertly enabled just by tapping the Bluetooth menu within Settings.

---

[23] an open source TCP packet forging tool

[24] Storm, *Researchers hack a pacemaker, kill a man(nequin)*

[25] Storm, *Researchers hack a pacemaker, kill a man(nequin)*

[26] news.sciencemag.org

[27] In *The Security of Bluetooth Systems and Devices,* Radak attributes this to the release of version 2.1.

Thus, a cautious user would do well to make sure Bluetooth is deactivated when not in use and also not attempt to pair with other devices in a public area. In this increasingly wearable-crazed world, it is indubitably important for the modern-day user of Bluetooth-capable devices to, at the very least, be aware of the potential threats and take even the smallest precautionary measures to stay clear of a potential invasion of sensitive and personal information.

References

Chai, Elaina, Ben Deardorff, and Cathy Wu. "6.858: Hacking Bluetooth." Www.mit.edu. MIT, 2012. Web. 11 Dec. 2015. <https://css.csail.mit.edu/6.858/2012/projects/echai-bendorff-cathywu.pdf>.

Chow, Ming. "Understanding the Treat Profile of Mobile Apps." Tufts University, 30 Nov. 2013. Web. 11 Dec. 2015.

"Could a Wireless Pacemaker Let Hackers Take Control of Your Heart?" Could a Wireless Pacemaker Let Hackers Take Control of Your Heart? Web. 11 Dec. 2015. <http://news.sciencemag.org/health/2015/02/could-wireless-pacemaker-let-hackers-take-control-your-heart>.

Gordon, Nelli. "Bluetooth Security." 10 May 2011. Web. 11 Dec. 2015. <http://www.ece.umd.edu/class/ents650/BluetoothSecurity.pdf>.

"How Bluetooth Works." How Bluetooth Works. Web. 11 Dec. 2015. <http://www.i-programmer.info/programming/hardware/2602-how-bluetooth-works.html?start=1>.

Niem, Tu C. "Bluetooth and Its Inherent Security Issues." Www.sans.org. SANDS Institute InfoSec Reading Room, 11 Apr. 2002. Web. 11 Dec. 2015.

Premnath, Sriram Nandha. "Battery-Draining-Denial-of-Service Attack on Bluetooth Devices." Www.utah.edu. University of Utah. Web. 11 Dec. 2015. <http://www.cs.utah.edu/~nandha/Abstract_2008.pdf>.

Radack, Shirley. "SECURITY OF BLUETOOTH SYSTEMS AND DEVICES: UPDATED GUIDE ISSUED BY THE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST)." Www.nist.gov. NIST, 2002. Web. 11 Dec. 2015. <http://csrc.nist.gov/publications/nistbul/august-2012_itl-bulletin.pdf>.

"Researchers Hack a Pacemaker, Kill a Man(nequin)." Computerworld. Web. 11 Dec. 2015. <http://www.computerworld.com/article/2981527/cybercrime-hacking/researchers-hack-a-pacemaker-kill-a-man-nequin.html>.

Saravanan, K., and L. Vijayanand. "A Novel Bluetooth Man In The Middle Attack Based on SSP Using OOB Association Model." Erode Sengunthar Engineering College and Marthandam College of Engineering and Technology. Web. 11 Dec. 2015.

"Search." FierceMobileHealthcare. Web. 11 Dec. 2015. <http://www.fiercemobilehealthcare.com/story/ftc-vendors-sharing-mhealth-fitness-app-data/2014-05-12; http://venturebeat.com/2014/08/11/senator-chuck-schumer-wants-the-ftc-to-regulate-data-from-fitness-wearables/>.

"Senator Chuck Schumer Wants the FTC to Regulate Data from Fitness wearables." VentureBeat. Web. 11 Dec. 2015. <http://venturebeat.com/2014/08/11/senator-chuck-schumer-wants-the-ftc-to-regulate-data-from-fitness-wearables/>.

"Tractica." Tractica. Web. 11 Dec. 2015. <https://www.tractica.com/newsroom/press-releases/enterprise-applications-for-wearable-technology-are-expanding-at-a-rapid-pace/>.