Tufts University

Into the Blue Depths

A User's Guide to Bluetooth Security Vulnerabilities

Dixon Minnick
12/12/2014

Abstract

Bluetooth is a ubiquitous communication technology present in the majority of today's mobile phones as well as many other personal devices. It is designed to provide both confidentiality and authentication in connections between devices, and does so using custom algorithms for key generation. Unfortunately with the goal of this technology being primarily ease of use and standardization, it is potentially vulnerable to a number exploitations. In spite of this Bluetooth is considered to be acceptably secure in general circumstances and rarely do we hear about significant security breaches as a result of Bluetooth exploitation.

In this paper we discuss a number of potential exploitation techniques that can lead to a breach in personal information or cause inconvenience or harm to the target, as well as their significance to the average consumer. This also includes a risk assessment of these vulnerabilities as well as recommendations regarding best practices to minimize these risks for the everyday Bluetooth user.

To The Community

Everywhere we go we are tasked with keeping our personal information safe whether we realize it or not. Technology has enabled us to experience the world in ways we never have before and given us unprecedented levels of convenience through interconnectivity. All too often this

convenience with the risk that the personal information that we share with trusted individuals, entities and services that we choose, becomes exposed to others that we do not. Bluetooth is one of these extremely convenient technologies, one which the majority of us trust by default without considering what could be at stake when we indulge in its convenience. We as users make assumptions every day about the security of the technology we use. The focus of this document is not intended to be a low-level examination of Bluetooth security protocols, or a classification of vulnerabilities inherent to it. Much research exists already in these areas and an overview should suffice. What individuals should take away when reading this paper is an understanding of the implications of exploiting these vulnerabilities by using specific and plausible examples of Bluetooth attacks. Such an understanding should guide individuals in taking proper precautions to avoid unintentional dissemination of their personal information.

Introduction

Bluetooth was intended as a universal replacement for peer-to-peer wired connections between devices. Since it was rolled out at the turn of the century, it has become a go-to standard for connecting everything from speakers to headsets to car entertainment systems and perhaps most notably mobile phones. The overwhelming majority of mobile phones in use today are Bluetooth capable. Cell phones also contain vast treasure troves of personal information, which makes them the most common target for Bluetooth attacks.

Security Modes

When enabled, most Bluetooth devices operate in one of three basic security modes:[1]

Silent: A device in silent mode will not accept incoming connections; it merely monitors Bluetooth traffic. While almost no consumer products operate in silent mode, many Bluetooth security tools (sniffers and scanners), used by penetration testers and attackers alike, have the ability to operate in silent mode to avoid detection.

Private: A device in private mode will not accept incoming Bluetooth pairing connections and does not broadcast any information. It will however respond to connections if the specific Bluetooth hardware address (BD_ADDR) of the private device is known by the connecting device. Today many consumer devices with Bluetooth enabled operate in private mode until switched into pairing mode to connect to new devices. The Apple iPhone is an example of this.

Public: Also known as Pairing or Discoverable mode. The device is publicly discoverable and broadcasts its Bluetooth Address (BD_ADDR) as well a list of its available connection services. The device will accept incoming pairing connections.

NIST designates "security modes" slightly differently, numbering them 1, 2, 3 and 4[2]. Security modes 2-4 all have varying degrees and implementations of authorization, authentication and encryption steps, each with varying degrees of difficulty for an attacker to exploit. What is more interesting to the average user however, is the existence of Security Mode 1. If this were to be classified as one of the three basic security modes described above, it would be its own designation: "None". Security Mode 1 implements no security measures and devices using this

[1] (Tarique, 2012)
[2] (Padgette, 2012)

mode will accept and authenticate all pairing requests. It is strongly recommended that no personal information be transmitted between devices using Security Mode 1[3].

This however does not mean that an attacker can instantly steal a user's information if they were to connect to a plug and play Bluetooth speaker using Security Mode 1. Likely the user's personal device in question (such as an iPod or phone), will have a stronger security mode implemented. The idea behind Modes 2-4, is that during the authorization step of the pairing process (described below), only certain services will be authorized to be sent over the Bluetooth connection, and such a connection with a device using Security Mode 1 would undoubtedly not permit authorization for services such as address book or any other sensitive user information. Very few devices still fall into this category.

Pairing

The pairing process itself is a multi-step process consisting of authentication and authorization steps followed by encryption steps if the device specifications support encryption. In simple terms, during the pairing process, once a connection is initiated, authentication must take place based on the matching of a cryptographically generated key based on a PIN number entered on both devices. At this point, once the authentication is established, the exchange of encryption keys takes place if available, and the "physical" link between devices is established. The main principle behind securing the pairing process is that a certain chain of events[4] must occur in order and each exchange on its own is not particularly significant to eavesdropping parties. However, some attackers with the right information can piece together the information about a

[3] (Padgette, 2012)
[4] (Tarique, 2012)

pairing exchange into something usable. This is very difficult to do without prior knowledge about the devices taking part in the pairing exchange.

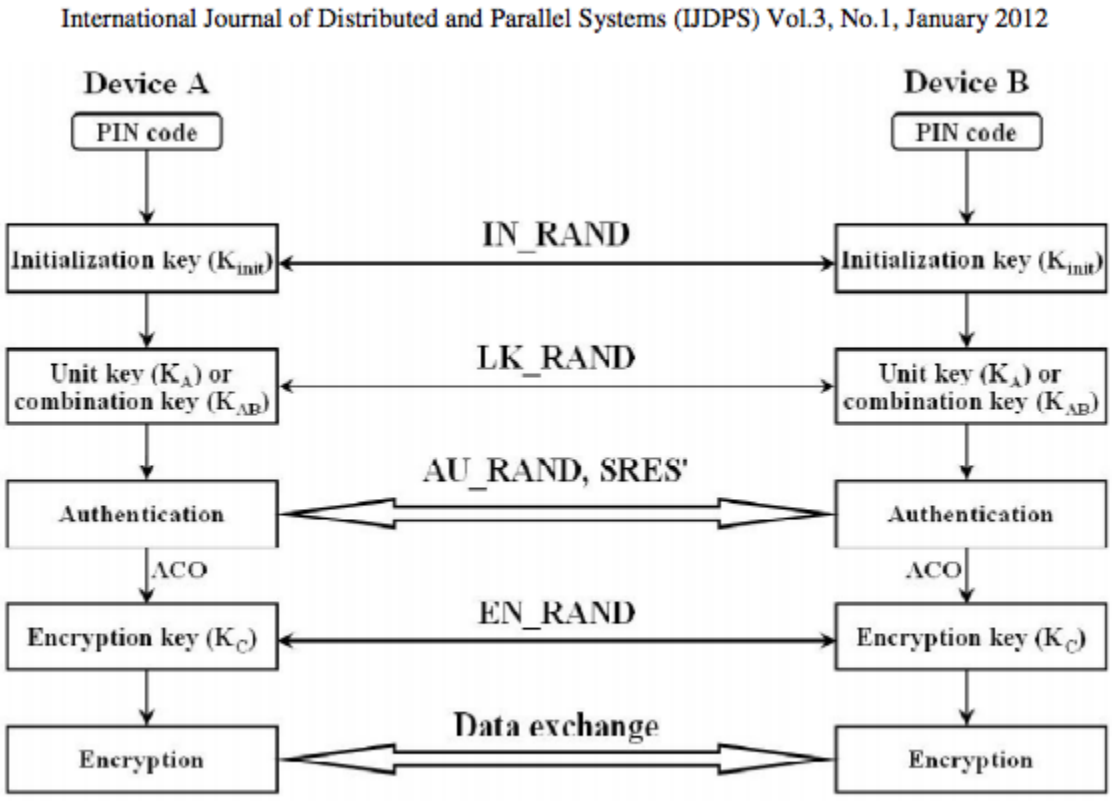


Fig. 2 Illustration of Bluetooth security operations

[5]

As many users may be aware, rarely is it necessary to enter a PIN number on both devices as one of the devices often lacks a method of entry, a screen or both. In these cases, a PIN is produced by one of the devices, outputted to the user either audibly or visually as the case may be to be entered into the other device that has an input capability. The NIST and NSA guidelines recommend that manufacturers generate this PIN randomly, but many manufacturers simply use a pre-set hard-coded PIN number [6] [7]. The problem is that these PIN numbers are easy to guess. This weakness is often exploited by attackers seeking to gain access to a limited I/O device.

[5] (Tarique, 2012)
[6] (Padgette, 2012)

Implemented in 2007 with Bluetooth 2.1[8], modern devices often support pairing using an authentication and authorization process known as Secure Simple Pairing (SSP). This method foregoes the manual entry of a PIN number, but requires a yes or no confirmation from the device accepting the pairing connection. This is more secure than fixed-PIN pairing, though less secure than variable PINs generated at random. For this reason it is recommended that users never pair devices in public settings, as there is no way to guarantee that the identity of the device you are attempting to pair with has not been impersonated by an attacker.

Exploiting Vulnerabilities

Bluetooth being a peer-to-peer technology, in order to exploit a vulnerability, an attacker needs to maintain proximity with their target. Depending on the power rating of the Bluetooth device, this can range anywhere from 10 to 100 feet. In actuality, this range means absolutely nothing. It has been demonstrated [9] that Bluetooth range can be extended up to a mile using a tuned amplifying Bluetooth dongle. Some of these devices are designed specifically to perform various attacks and are referred to as BlueSniper rifles. Attacks can fall into one of three categories: Denial of Service (DoS), Disclosure of personal information, and Misleading the target. DoS is usually considered to be the least severe, but annoying nonetheless. While the majority of these vulnerabilities occur during the pairing process, a device in private or even silent mode can still be detected and in some cases exploited by some attacks. Using tools freely available on the internet, I attempted to exploit these vulnerabilities using my own iPhone 5, running iOS 8.1 to

[7] ([redacted])
[8] (Tarique, 2012)
[9] (Car Whisperer)

demonstrate the applicability of these threats, and in many cases the counter-measures put in place to minimize them.

BlueScanning

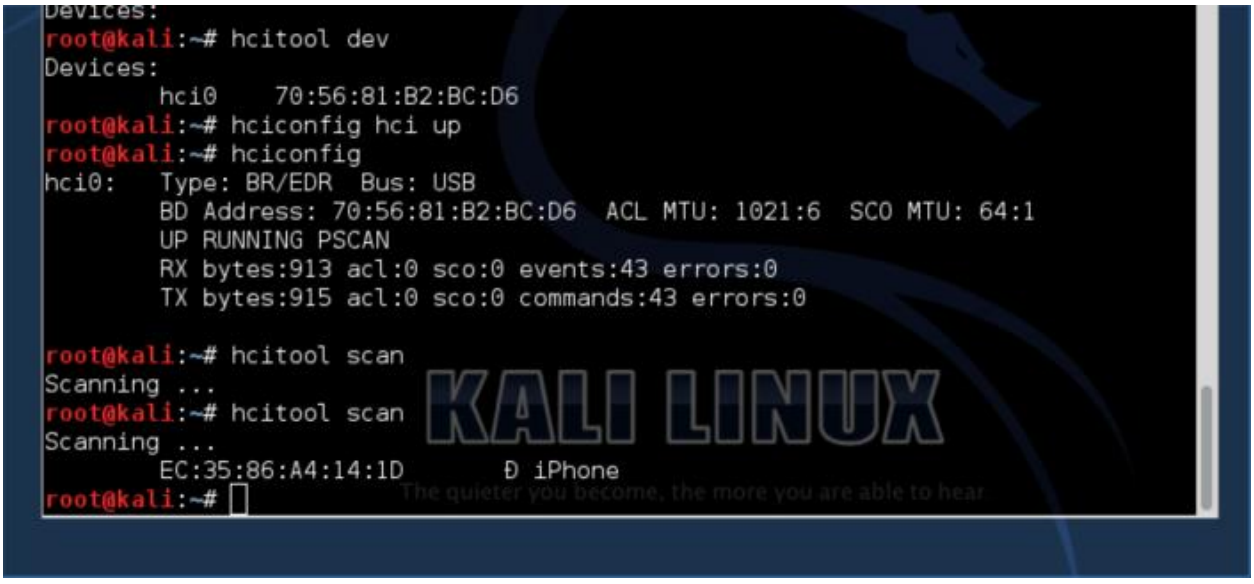
This is simply the process of scanning for Bluetooth devices. The default behavior of most Bluetooth scanners will simply return a list of devices detected in discoverable mode, usually with the device name and its list of available services. Indeed I was able to discover my own iPhone and its associated information using the scanner built into Kali Linux. Using a more sophisticated software scanning tool like SpoofTooph[10], it is possible to pull of information such as full name and device model number. In both cases I was only able to collect BD_ADDR and device name. However often this is enough. Frequently a device name is simply the default factory name which can allow the attacker to determine the specifications of the target device. Other times a device name will be set to something like "John's iPhone," which still gives the attacker a pretty good idea about the target.

While this might at first seem innocuous enough, this type of information collecting can just be the basis with which the malicious collector can use to launch a more sophisticated attack. The aforementioned SpoofTooph software's primary goal is to hide the identity of the attacker. One way it can do this is to collect this collected device profile and use it to impersonate that device. Furthermore this activity is anything but innocuous in the wrong hands, even without using it to breach inside a target. This information once collected can be used to track the physical location of the target, even when in private mode. Back in the early days of Bluetooth, there were confirmed reports of gangs in Britain using Bluetooth scanners around parked cars to see if any

[10] (Dunning, 2010)

Bluetooth enabled electronics inside were in public mode, thus helping them select targets for theft.

Worse still, there are other scanners that are able to brute-force scan for devices in private or even silent mode by iterating over BD_ADDRs. Using SpoofTooph I eventually was able to detect my own iPhone in private mode, but only after several minutes using a 5 address range that I knew contained my BD_ADDR. Even then I was only able to collect the device name, nothing more, though older phones tend to divulge more information. With modern phones, individuals are generally safe from scanners while in private mode. However putting a device in discoverable mode significantly increases the risk of device information being logged, which again, is why it is never a good idea to pair in a public setting. And once a device is logged, the scanner can find the device even when in private mode. This too I was able to demonstrate using my own iPhone.



(Demo image of scanning for my iPhone while in pairing mode)[11]

[11] (Minnick, Bluetooth Hacking Demo)

(Demo image of pinging my device while in private mode)[12]

Bluejacking

Bluejacking in itself does not take control of the target device, but is usually a form of spam. To demonstrate the concept, no additional software is required. Many cell phones support Vcards, or virtual business cards, which is a method of receiving a contact via Bluetooth. Back in the early days of Bluetooth, many cellphones were always in public mode, meaning they always received Bluetooth connections. This allowed people to freely exchange Vcards over Bluetooth, sometimes unsolicited. What Bluejacking actually is, is simply sending these unsolicited Vcards, usually containing spam of some kind.

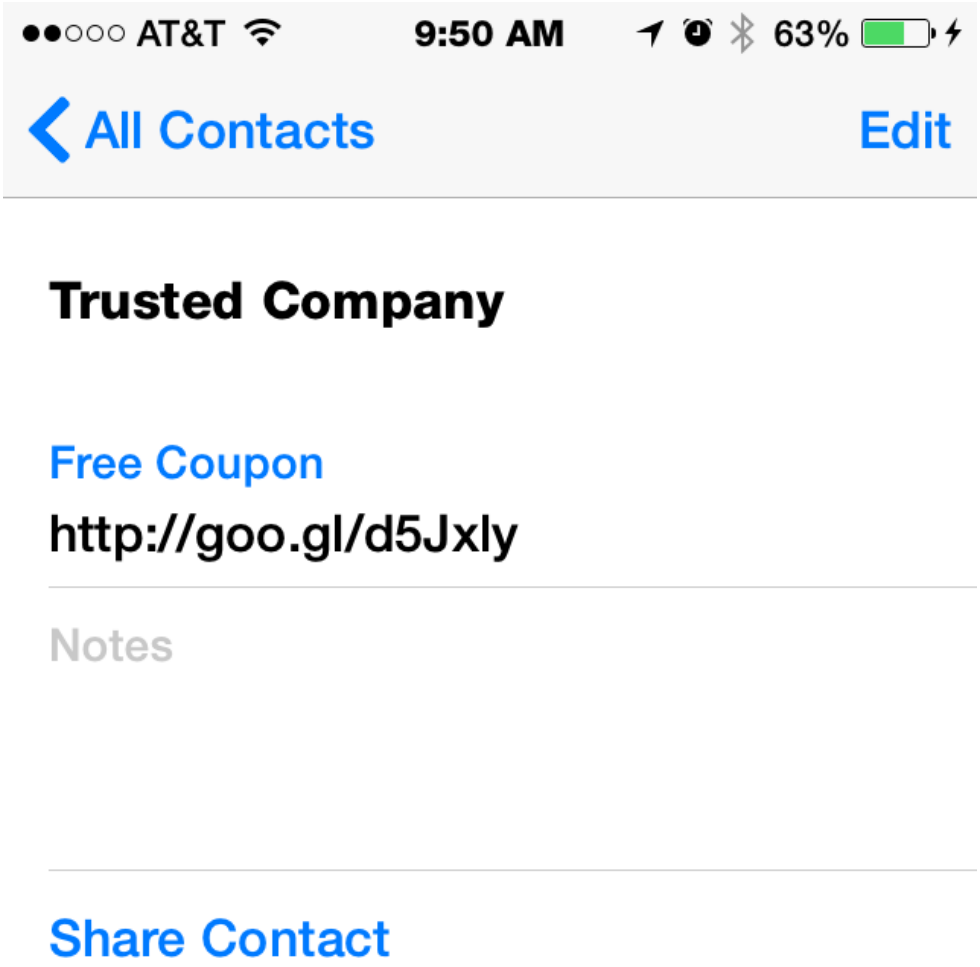
In many devices, it was possible to create a DoS attack by simply sending these Vcards over and over. Causing this type of DoS attack is sometimes referred to as “Blueballing”. There exists a software tool called VcardBlaster specifically designed to carry out these attacks. The vast

[12] (Minnick, Bluetooth Hacking Demo)

majority of smart phones these days are by default in private mode when Bluetooth is enabled, meaning that unless the sender knows what device it is sending to, the spam will simply be ignored. iOS in particular has been virtually immune to these attacks since an update in 2007, as a result of not accepting Vcards over Bluetooth, however it was recently discovered that prior to iOS 8, devices would be susceptible to a DoS style attack during the software update process[13]. In order to enable file sharing of any kind over Bluetooth on an Apple device requires the user to jailbreak it. The reality is that this attack is not particularly harmful to modern phones. Almost all phones require you to confirm before allowing a Vcard to your contact. That being said, another DoS software tool called Blueper exploits a characteristic of some phone operating systems to cache incoming Vcards until the user takes action on it, by sending more cards than the user can possibly act on, thus overflowing the cache and crashing the phone.

A good rule of thumb is to never accept such a Vcard in public from someone you do not know. Many phones that still permit the use of Vcards have software measures in place to prevent succumbing to a DoS attack. Older phones though are more susceptible. Even so, the tact of some BlueJackers is to deceive the user into willingly accepting the contact as being from a trusted source. Usually this will then contain a link to a site containing spam or a type of mobile malware.

[13] (Bluetooth CVEs)



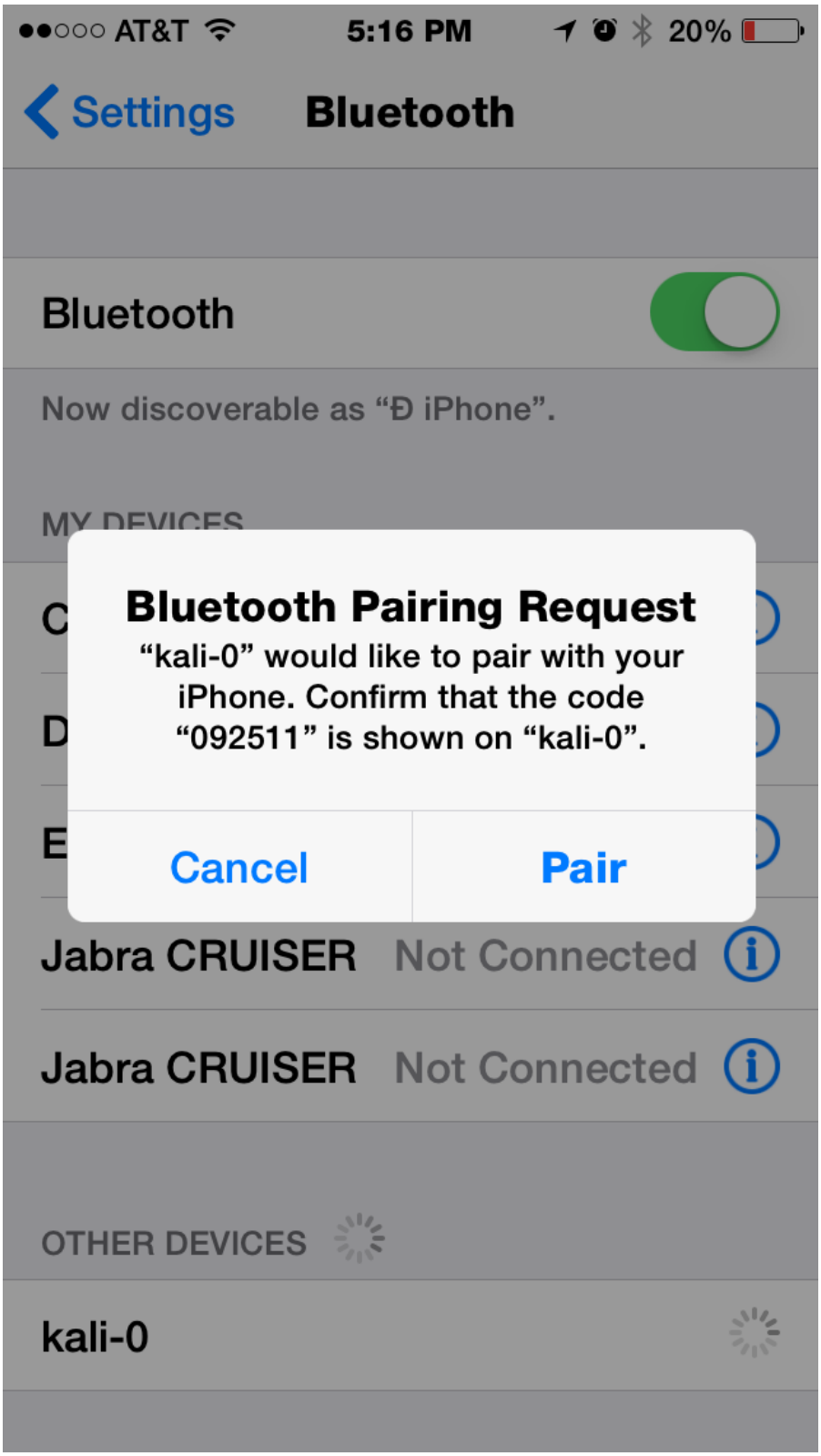
(Demo concept image of Vcard impersonation, following link encouraged)[14]

BlueSnarfing

BlueSnarfing is the process of gaining unauthorized access to a target device by pairing and then collecting the personal information from the device. This can be accomplished in a number of ways. Having successfully eavesdropped during the pairing process between two devices, the

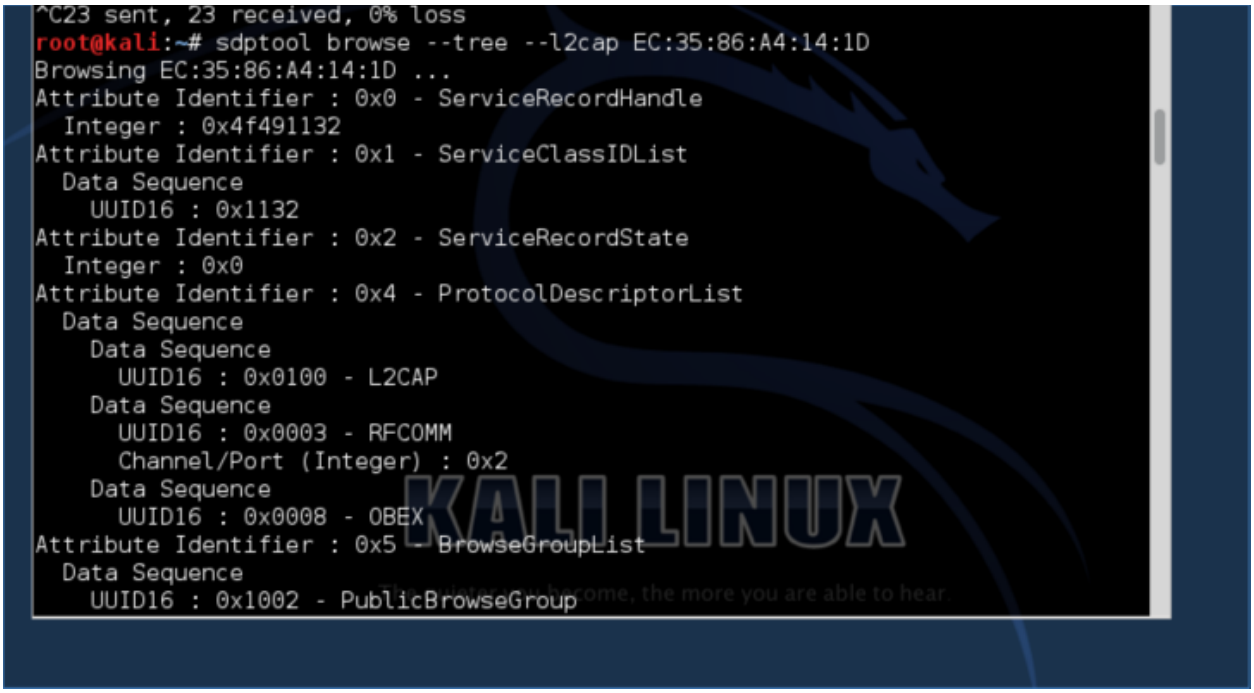
[14] (Minnick, Bluetooth Hacking Demo)

attacking device can act as a Man-In-The-Middle intercepting all of the traffic traveling over the connection. More simply, an attacker, having scanned for devices can use SpoofTooph to impersonate said device and trick the target into intentionally pairing with the malicious device. One available tool to implement this attack is called BlueSnarfer, which comes with built in support for accessing a target's address book. Using BlueSnarfer on my iPhone, I received an SSP pairing confirmation message asking if I wanted to pair with "kali-0". Attackers are rarely going to be that obvious. A target in a public place would undoubtedly be asked to pair with "SAMSUNG_HANDSFREE" when near a cell phone store at the mall, or "car media" when at a rental car center. Even a message asking if the target would like to pair with "Free Public WiFi" could potentially fool the unsuspecting.


●●○○○ AT&T
5:16 PM
20%
Settings
Bluetooth
Bluetooth
Now discoverable as “Đ iPhone”.
MY DEVICES
Bluetooth Pairing Request
“kali-0” would like to pair with your iPhone. Confirm that the code “092511” is shown on “kali-0”.
Cancel
Pair
Jabra CRUISER Not Connected
Jabra CRUISER Not Connected
OTHER DEVICES
kali-0

(Demo image of BlueSnarfing)[15]

Even so during my own testing, though I was able to successfully pair, I was never able to access the address book. When attempting to access that specific service, nothing happened. That being said, I have never successfully gotten the address book to sync via Bluetooth to either of our car systems either.



BlueBugging

Related to BlueSnarfing, is BlueBugging, using the paired connection to take control of the target paired device, often without the user's knowledge or consent. Common actions performed by attackers include sending text messages and making calls, either to expensive toll services, or spamming the user's contact list. BlueSnarfer includes a BlueBugging agent to execute these attacks. Similar to my experience with gaining access to the address book, I was unable to

[15] (Minnick, Bluetooth Hacking Demo)

execute calls or texts from the command line, something that is also true of our car Bluetooth systems as well. It is unclear whether this is Apple's doing or something flawed with my phone. This however, if successful, can be extremely damaging to the target user. Take for example this hypothetical situation: Across Africa, Kenya in particular, thrives an economy based on mobile payments. Using a couple of phone numbers and a PIN number, any Kenyan with a cell phone, smart or not, can withdraw money or make payments to others. There are security measures in place to prevent fraud, of course[16], however if BlueSnarfing/BlueBugging is successfully implemented, an attacker could intercept the PIN in the process of being transmitted and then use a BlueBugging attack to make fraudulent payments from the target's device and then erase the confirmation response to cover their tracks. There haven't been any cases of this reported so far, however many of the cell phones in Africa are older models, many supporting older Bluetooth technology more susceptible to these types of attacks.

Tools

Tools to implement these kinds of attacks are freely available on the internet. In particular, the BlueDiving[17] suite of tools implements scanning, snarfing and bugging. Another tool which implements all of the above attacks plus several others is BlueMaho, which packages them in a graphical user interface. Unfortunately documentation for BlueMaho is virtually non-existent. BlueDiving also implements another kind of Bluetooth based attack called CarWhisperer.

CarWhisperer

[16] (Hibbard, 2014)
[17] (Ballman, 2012)

CarWhisperer takes advantage of the fact that the majority of limited I/O hands free devices use easy to guess default PIN settings or make use of SSP[18]. It also operates off of the assumption that many of these devices are in public mode either all the time or too often (such as when powered on). This might seem like another example of "junk hacking", or trivial exploitation of a random interconnected device, but in fact it represents a very real vulnerability of Bluetooth technologies with serious implications.

When the CarWhisperer tool is run, it scans for such devices and attempts to pair with them using SSP or guessing the PIN based on the information retrieved by the scan. Once paired CarWhisperer can then transmit audio to the hands free device, such as a fake traffic report. Worse still CarWhisperer also records whatever audio is picked up by the hands free device, meaning that any conversation you may be having will be heard by the attacker.

While it is certainly true that this type of attack means the attacker must remain in a relatively close proximity to the target, which presumably is frequently a car that moves quickly, it is a relatively easy exploitation to succumb to. Users of older devices that are constantly or frequently in public mode have virtually no protection against this kind of attack. Those users that by default remain in private mode need to pay particular attention to not pair their devices in a public space, something that can be difficult in a car, especially when attackers may be observing your car for that very purpose. These devices also often lack the option of disabling Bluetooth altogether.

This is why users of rental cars should pay particular attention to the Bluetooth connection system in those vehicles, and taking care to clear out any previously saved entries from the device list of the car. One of those just might be an eavesdropper. And though this particular tip

[18] (Car Whisperer)

has been previously mentioned, do not pair these devices while still in the rental car complex. Doing so increases exposure to CarWhisperer attacks among others.

I attempted, and successfully executed a CarWhisperer attack on the Bluetooth system in my own 2011 Subaru. Thankfully, the results of the scan found that my car was not by default in discoverable mode, which significantly decreases my own personal risk of this exploitation. However, when putting the car in pairing mode, I had absolutely no problems completing the pairing request and was able to capture audio[19]. However, while audio capture is occurring, the LCD display on my stereo displays a "Talking…" message, which probably unintentionally serves as a security feature to detect this kind of attack.

[19] (Minnick, Hacking My Subaru: Car Whisperer Recording, 2014)

(Demo image of Car Whisperer interacting with car stereo)[20]

Overall Assessment

2007 and 2008 proved to be a turning point for Bluetooth security. This is for a number of reasons. First we can credit the release of Bluetooth v2.1, which addressed some of the flaws in the legacy pairing specification and also introduced Secure Simple Pairing (SSP) which both streamlined connections and added a layer of security. Subsequent updates to Bluetooth have continued to strengthen the security at authentication, encryption and link levels during the

[20] (Minnick, Bluetooth Hacking Demo)

pairing process[21]. The second factor is very likely the original release of NIST's Guide to Bluetooth Security in 2008[22], containing Bluetooth security specifications as well as best practices to be used by Bluetooth manufacturers. It also contains a number of security tips for the users.

These tips can be simplified to two main principles. First, it is strongly recommended to simply turn Bluetooth off when it is not in use. There is no better protection than that. The second main idea, which has been previously mentioned, is to never pair devices in a public space. Beyond that it's just a matter of specifics. For example what most iPhone users do not know about their Bluetooth settings, is that while even when Bluetooth is already on, an iPhone is only in private mode until the moment that the user clicks on the Bluetooth menu under settings. Clicking on that menu makes the device discoverable, meaning it becomes vulnerable. That being said, iOS does a relatively good job of defending from Bluetooth attacks.

Performing these types of attacks has become increasingly difficult on modern devices. The numbers speak for themselves. All major viruses, malware and exploitations propagated through Bluetooth showed up during the period between 2003 and 2007[23]. Since then, while some of these vulnerabilities persist, great steps have been taken to minimize the threat they pose. When one takes into consideration the rampant propagation of other forms of data compromization, Bluetooth vulnerabilities play a lesser role. This is not to say they should be ignored, but it is true that Bluetooth can be considered "acceptably secure" when the proper precautions are taken.

References and Supporting Material

[21] (Tarique, 2012)
[22] (Padgette, 2012)
[23] (Tarique, 2012)

[redacted]. (n.d.). *Bluetooth Security.* Retrieved from NSA Systems and Network Analysis Center Information Assurance Directorate: https://www.nsa.gov/ia/_files/factsheets/i732-016r-07.pdf

Ballman, B. (2012, June 7). *bluediving.* Retrieved from GitHub: https://github.com/balle/bluediving

Bluetooth CVEs. (n.d.). Retrieved from National Vulnerability Database: https://web.nvd.nist.gov/view/vuln/search-results?query=bluetooth&search_type=all&cves=on&startIndex=0

Car Whisperer. (n.d.). Retrieved from Trifinite.org: http://trifinite.org/trifinite_stuff_carwhisperer.html

Dunning, J. (2010). *Breaking Bluetooth By Being Bored.* Retrieved from Defcon: https://www.defcon.org/images/defcon-18/dc-18-presentations/Dunning/DEFCON-18-Dunning-Breaking-Bluetooth.pdf

Hibbard, S. D. (2014). *Mobile Payment System Security Breaches in the Developing World: The M-PESA.* Retrieved from Academia.edu: http://www.academia.edu/8040561/Mobile_Payment_System_Security_Breaches_in_the_Developing_World_The_M-_PESA_Pseudo-Hack

Minnick, D. (Composer). (2014). Hacking My Subaru: Car Whisperer Recording. [D. Minnick, Performer, & C. Whisperer, Conductor] Medford, MA, United States of America.

Minnick, D. *Bluetooth Hacking Demo.* Tufts University, Medford.

Padgette, J. a. (2012, June). *Guide to Bluetooth Security*. Retrieved from National Institute of Standards and Technology: http://csrc.nist.gov/publications/nistpubs/800-121-rev1/sp800-121_rev1.pdf

Tarique, M. a.-N. (2012, January). *Bluetooth Security Threats and Solutions: A Survey.* Retrieved from International Journal of Distributed and Parallel Systems (IJDPS): http://www.airccse.org/journal/ijdps/papers/0112ijdps10.pdf