Bluetooth and iOS 11:
How a Change in User Interface Compromises Security

Jacob Jaffe

Dec 2017

Abstract

Apple's line of mobile devices utilizing Bluetooth, such as the iPhone and iPad, received major behavioral changes with the release of iOS 11 in 2017. Introducing an ambiguously disabled state for Bluetooth, dubbed as an "off-ish" state by the EFF[1], these changes reflect a deprioritization of security. Apple has a history of being on the forefront of Bluetooth integration with mobile devices, and it is imperative that the changes they introduced do not become the norm. These changes make affected devices more susceptible to Bluetooth vulnerabilities and to Bluetooth sniffing, and compromise user efforts to keep their devices secure due to the misleading design.

Contents

1 Introduction

Bluetooth is an ever increasingly popular method of exchanging data wirelessly, and has been adopted even more rapidly in recent years as IoT devices have been developed. More than 8.2 Billion devices currently use Bluetooth, and nearly all smartphones use Bluetooth[21]. While Bluetooth has become ubiquitous across mobile devices, the standards for using Bluetooth on mobile devices have been frequently changing.

In 2011, Apple released the iPhone 4S, the first phone to utilize Bluetooth 4.0 / LE[17]. Prior to this, utilizing Bluetooth on a phone generally meant slow speeds at high costs; the iPhone 4S was able to transfer data via Bluetooth at eight times the speed of its predecessors and competitors, while using only a tenth the power[7]. A little over a year later, Google followed suite with the Nexus 4, its latest flagship phone adopting Bluetooth 4.0. Apple had put itself at the forefront of promoting Bluetooth on mobile devices.

Apple has since retained this position as a major influence on Bluetooth usage. In 2016, the iPhone 7 was released, distinctively lacking a headphone jack[12]. Customers choose either to use their headphones via an adapter, or to heed way to Apple's vision, and exclusively use Bluetooth for their listening. A year later, Google released the Pixel 2, also lacking a headphone jack in favor of wireless streaming; again, the precedent Apple had set became adopted.

As the standards of Bluetooth usage, or lack thereof, have been in flux, a general procedure for securing Bluetooth on mobile devices has been inconsistent across devices and their iterations. The release of iOS 11 brings major changes to Bluetooth usage.

2 To the Community

With Apple's history of setting precedents for Bluetooth behavior on mobile devices, it is reasonable to predict that more mobile devices will be developed with similar ambiguous functionality in the near future. This is a dangerous potential, and should be avoided. In order for this to be avoided, it is important that further developments in user interface take security into account.

3 The Changes

Since the original iPhone was released with Bluetooth 2.0 in 2007, all iOS versions have included support for enabling or disabling Bluetooth. Originally, this was done solely through the settings menu.

This remained the only way for users to control Bluetooth being on or off until the addition of the control center to the user interface.

3.1 Old Behavior

In 2013, Apple unveiled the iPhone 5; alongside the new device, Apple also debuted iOS 7, which featured the *control center*. As put by Apple, the control center allowed users to:

> Quick access to commonly used controls and apps with a swipe up from the bottom of the screen
>
> Turn on & off Airplane Mode, Wi-Fi, Bluetooth, Do Not Disturb; adjust screen brightness; access media controls; turn on AirPlay and AirDrop
>
> Quickly access flashlight, timer, calculator, camera and music controls [13]

Significantly, this update enabled users to more easily disable their Bluetooth. When a user toggled Bluetooth of a device off in this new manner, via the control center, they were guaranteed that:

- The device disconnects from all connected Bluetooth devices

- The device no longer connect Bluetooth devices

This behavior was established in 2013, and remained the same through iOS 10.3.3, of 2017.

3.2 New Behavior

In iOS 11, the behavior for toggling Bluetooth from the command center was altered. While similar to the previous behavior, the new operating system includes subtle differences. Under the new behavior, when a user toggles Bluetooth of a device via the control center, they are guaranteed that:

- The device disconnects from all connected Bluetooth devices, *except for Apple Watch, Instant Hotspot, Apple Pencil, and Continuity features, like Handoff* [18]
- The device no longer connects Bluetooth devices, *until it's 5 AM local time*[18]

This new behavior is confusing. The previous method of turning off Bluetooth now only *sort of* turns off Bluetooth: ambigous security settings are dangerous forms of user interface [14] [9].

As we have seen, though, Apple has a history of changing the way Bluetooth is used. This begs the question, why is this change so significant?

3.3 Interface

At a high level, these changes don't seem to be very different than the previous departures from tradition that Apple has led for Bluetooth. Like before, Apple is changing its products in an effort to shift how its customers use Bluetooth; especially like previous changes, the iOS 11 update incentivizes the use of more Bluetooth products. However, the underlying design providing that incentive is drastically different this time.

Previously, Apple incentivized Bluetooth usage by providing better Bluetooth (adopting the more recent versions & accompanying transceivers), promoting more Bluetooth devices and functionality (Apple Watch, Pencil, Airdrop, etc.), and by deterring alternatives (removing the headphone jack). These changes focused on changing what users could use Bluetooth for/with. These changes affected how often Bluetooth was used, and how the phone transmitted Bluetooth signals; the interface for using Bluetooth remained the same. Turning Bluetooth off always meant the same thing.

The changes introduced by iOS 11, however, incentivize Bluetooth usage by altering how users connect with Bluetooth. As above, they change the *guarantees* upon interaction with the device UI. Turning Bluetooth does not mean the same thing within different interfaces of the iOS, nor does it mean the same thing as previous iterations of the operating system. However, the interface for toggling Bluetooth remains the same, and so this is not conveyed to a user.

Apple has chosen to make Bluetooth easier to use by introducing a system that favors Bluetooth always being active: a system that encourages users to keep their devices connected by discouraging disabling Bluetooth. These changes make it more complicated for a user to attempt to secure their device. Worse, these changes convey a false security to users who naively use the same UI that previously did disable Bluetooth. These are dangerous changes, because it is imperative that mobile devices are secured against the very real Bluetooth threats.

4 Vulnerabilities

Ever accompanying Apple's iOS - Bluetooth integration, Bluetooth vulnerabilities have been a serious threat to iOS users. As Bluetooth compatible mobile devices have become more frequently used, notably with phones, tablets, and various IoT devices, a surge of exploits has occurred.

4.1 AirDrop Exploit

In iOS 8, a flaw in the Bluetooth enabled Airdrop system resulted in a vulnerability which, when exploited, allowed an attacker to install malware onto a victims phone [11]. This vulnerability, developed by reasearcher Mark Dowd, required only that the victim had Bluetooth enabled for the attacker to compromise the phone.

Under normal circumstances, AirDrop is used to transfer files via Bluetooth across iOS devices. Part of the the transfer process requires the receiving device to accept the file. Dowd discovered, however, that an attacker could execute a directory traversal attack during the transfer, and modify AirDrop configuration files. The iOS accepted certain certificates without requesting aproval, as long as they were deemed to be authentic Apple enteprise certificates. However, in a similar manner used to jailbreak iPhones, Dowd discovered that modifying these files would allow *any* application to bypass the authorization and automatically be accepted [8].

Exploiting this, an attacker could bypass the lock screen of a physical device they possessed. Worse, an attacker could install malicious apps onto any iOS device with Bluetooth on and within range. Via this installation, the attacker could install malware to the device.

The payload? Trivially, a malicious app installed could access device data such as contacts, pictures, location, and various media; anything a legitimate app could. Dowd explained that it could, however, be even more serious:

> The best thing to do though would be to find a kernel vulnerability that you launch from your app to gain full privileges to the phone in the same way jailbreaks do [8].

While this flaw was mitigated with iOS 9, other Bluetooth vulnerabilities persisted.

4.2 BlueBorne

In 2017, Armis Lab revealed a suite of Bluetooth vulnerabilities that compromise almost all Bluetooth devices, dubbed BlueBorne [3]. Armis describes BlueBorne as an “attack vector”, and that:

> these attacks can be launched without any user interaction on the part of the victim user and without the device being put into “discoverable” mode [2].

The BlueBorne attack vector is composed of eight discovered vulnerabilities across Bluetooth architecture. Some of these vulnerabilities specifically target iOS.

For example, one of the attacks causes exploits an improper validation of packet size that occurs when a device receives a Bluetooth Low-Energy Audio Protocol packet. An attacker can send packets which impersonate this protocol; when memcpy is called to copy the packet data to device memory, the memory is instead copied beyond the intended scope, onto the Bluetooth call stack [19]. An attacker may exploit a Heap-based Buffer Overflow in this manner, injecting commands into the call stack, thus compromising the device [6].

The security researches at Armis disclosed their findings to, among other actors, Apple, before releasing them publically [2]. By doing so, Armis enabled Apple to patch the vulnerability with the release of iOS 10. Had the BlueBorne attack vector not been discovered by Armis, but by a malicious actor, then the vulnerability would not have been patched by Apple before becoming exploited.

Discovering these vulnerabilities preemptively should not reassure us with confidence that our devices are safe from these threats. Rather, they should remind us of the precarious state our security lies in.

4.3 Inevitable Weaknesses

These are just two examples of Bluetooth vulnerabilities prevalent in recent iOS versions. Bluetooth flaws and vulnerabilities have plagued iOS since its beginning with the iPhone 1 [16].

As put by Armis in their BlueBorne white paper:

> Bluetooth is complicated... Too many specific applications are defined in the stack layer, with endless replication of facilities and features... while the WiFi specification (802.11) is only 450 pages long, the Bluetooth specification reaches 2822 page[3].

In general, a more complicated procedure gives rise to more flaws and vulnerabilities. While we are lucky to have had these major vulnerabilities disclosed and thus mitigated appropriately, this is not necessarily the rule.

In order to design secure systems, the prospect of such zero-days being exploited needs to be considered as not only as a possibility, but as an inevitability. As Apple continues to build new features, and especially as it continues to work with new technology with Bluetooth, the risks of Bluetooth exploits emerging is prevalent. The changes in iOS 11, however, do not reflect this mentality; had the behavior of iOS 11 Bluetooth disabling coexisted with an exploited version of BlueBorne, the changes to the UI would have facilitated more exploits occurring, as phones would automatically present their vulnerable connection publicly.

5 Sniffing

Besides making phones more susceptible to potential exploits, the changes in iOS 11 also makes sniffing Bluetooth devices easier. Sniffing Bluetooth is relatively simple through tools like Blue Hydra, which can discover Bluetooth devices and track their use [4]. When a Bluetooth or Bluetooth Low Energy device is active, it constantly polls the world, broadcasting its presence. The data they advertise includes uniquely identifiable information, such as the MAC address and UUID of the device, as well as meta data such as name and device information [10]. For an attacker, this info can serve as a useful way to gain an understanding of the devices and behaviors of a victim. Sniffing the Bluetooth traffic of an area can even be used to determine the distance between attacker and victim via the Received Signal Strength Indication [22].

On iOS 11, a device with Bluetooth set “off-ish” will continue to be connected to Apple products including Apple Watch, Instant Hotspot, and Apple Pencil, among others [18]. To maintain this connection, the device continues to emit the afore mentioned data over Bluetooth. Particularly, the L2CAP channel setup between the two devices remains, and so the devices continue to send and receive packets which advertise their data. For example, an iPhone paired with an Apple Watch will remain connected, and packets will continue to be transmitted.

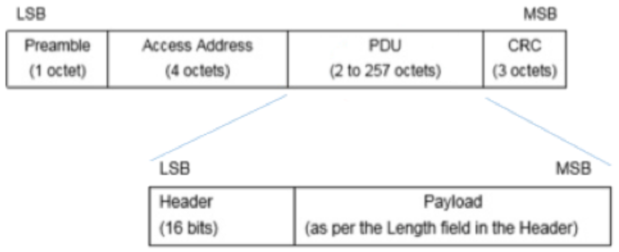

Figure 1: Composition of a Bluetooth packet [5] [20]

When a device is attempting to connect, or transmitting data across a connection, the packets (fig 1) sent guarantee that the certain information is always present. From the Preamble and Access Address, the intention and the UUID of the device is made public. Within the PDU, the Payload generally consists of information which can identify, if not state, a device’s hardware and firmware details.

For devices on iOS 11, this means that even in an “off-ish” state, those devices are still broadcasting their state to the world around them, even if they are not in ‘discoverable’ modes, and the devices they are paired with give further information through their response protocols.

6 Mitigation

In sectors requiring mobile device security, there is one established protocol for securing Bluetooth devices: disable it, or at least disable it when its not being used [15]. With iOS 11, the ambiguity of the interface compromises this practice. The interface leads users to believe that they have disabled Bluetooth, while they are still susceptible to Bluetooth attacks and sniffing.

The misleading user interface needs to reflect the actual behavior of the device in order for people to correctly use it for their own security.To facilitate this correct usage, Apple needs to use a more transparent interface for their Bluetooth devices. Second to that, it is imperative that users are aware of the differences in behavior, so that their intentions of security match their actions.

7 Conclusion

We have presented a history of Apple's Bluetooth development on iOS mobile devices. The changes apple instilled in iOS 11 have been shown to mislead users in what occurs when they disable Bluetooth via the command center. This behavior is dangerous because Bluetooth has been susceptible to numerous vulnerabilities in the past, and almost certainly be susceptible to more in the future. When considering security, a preemptive stance should be taken; In order to provide the best security for users, Apple needs to provide transparent security interface to their users.

References

[1] Andrés Arrieta. *iOS 11's Misleading "Off-ish" Setting for Bluetooth and Wi-Fi is Bad for User Security*. Electronic Frontier Foundation. Oct. 4, 2017. URL: https://www.eff.org/deeplinks/2017/10/ios-11s-misleading-ish-setting-bluetooth-and-wi-fi-bad-user-security (visited on 12/13/2017).

[2] *BlueBorne*. 2017. URL: http://go.armis.com/hubfs/BlueBorne%20Technical%20White%20Paper.pdf (visited on 12/09/2017).

[3] *BlueBorne Information from the Research Team - Armis Labs*. armis. 2017. URL: https://www.armis.com/blueborne/ (visited on 12/13/2017).

[4] *blue_hydra: Blue Hydra*. original-date: 2015-12-30T16:03:47Z. Dec. 12, 2017. URL: https://github.com/pwnieexpress/blue_hydra (visited on 12/13/2017).

[5] *Bluetooth Core Specification*. Dec. 6, 2016. URL: https://www.bluetooth.com/specifications/bluetooth-core-specification (visited on 12/07/2017).

[6] *CWE-122: Heap-based Buffer Overflow*. In: *Common Weakness Enumeration*. 3.0. MITRE, Nov. 14, 2017. URL: http://cwe.mitre.org/data/definitions/122.html (visited on 12/14/2017).

[7] Joe Decuir. *Bluetooth 4.0: Low Energy*. 2014. URL: https://californiaconsultants.org/wp-content/uploads/2014/05/CNSV-1205-Decuir.pdf (visited on 12/11/2017).

[8] Thomas Fox-Brewster. *One Great Reason To Update To iOS 9 - A Nasty Silent AirDrop Attack Is In Town*. Forbes. Sept. 16, 2015. URL: https://www.forbes.com/sites/thomasbrewster/2015/09/16/airdrop-ios-vulnerability/ (visited on 12/13/2017).

[9] Steven Furnell. "Can your users use security?" In: Electronic Workshops in Computing. online, 2006. URL: http://www.bcs.org/content/conWebDoc/6653.

[10] Sean Gallagher. *Hands-on: Blue Hydra can expose the all-too-unhidden world of Bluetooth*. Sept. 12, 2016. URL: `https://arstechnica.com/information-technology/2016/09/hands-on-blue-hydra-can-expose-the-all-too-unhidden-world-of-bluetooth/` (visited on 12/13/2017).

[11] Andy Greenberg. *Hack Brief: Upgrade to iOS 9 to Avoid a Bluetooth Attack — WIRED*. Sept. 16, 2015. URL: `https://www.wired.com/2015/09/hack-brief-upgrade-ios-9-now-avoid-bluetooth-iphone-attack/` (visited on 12/13/2017).

[12] Tom Henderson. *Apple's new Bluetooth security hole*. Network World. Sept. 13, 2016. URL: `https://www.networkworld.com/article/3119351/security/apples-new-bluetooth-security-hole.html` (visited on 12/13/2017).

[13] *iOS 7*. Sept. 18, 2013. URL: `https://support.apple.com/kb/DL1682?locale=en_US` (visited on 12/13/2017).

[14] Faouzi Kamoun and Halaweh Mohanad. "User interface design and e-commerce security perception: an empirical study". In: *International Journal of E-Business Research* 8.2 (2012), pp. 15+. DOI: `http://dx.doi.org/10.4018/jebr.2012040102`. URL: `http://link.galegroup.com/apps/doc/A294896300/AONE?u=mlin_m_tufts&sid=AONE&xid=69d46dac` (visited on 12/13/2017).

[15] Grover Kearns. "Countering Mobile Device Threats: A Mobile Device Security Model". In: *Journal of Forensic and Investigative Accounting* 8.1 (Jan. 2016), pp. 36–48. URL: `http://web.nacva.com/JFIA/Issues/JFIA-2016-4.pdf` (visited on 12/08/2017).

[16] Gregg Keizer. *iPhone's Bluetooth bug under the hacker microscope*. InfoWorld. Sept. 28, 2007. URL: `https://www.infoworld.com/article/2649238/computer-hardware/iphone-s-bluetooth-bug-under-the-hacker-microscope.html` (visited on 12/13/2017).

[17] *Our History — Bluetooth Technology Website*. URL: `https://www.bluetooth.com/about-us/our-history` (visited on 12/13/2017).

[18] *Use Bluetooth and Wi-Fi in Control Center with iOS 11*. Oct. 23, 2017. URL: `https://support.apple.com/en-us/HT208086` (visited on 12/10/2017).

[19] *Vulnerability Note VU#240311 - Multiple Bluetooth implementation vulnerabilities affect many devices.* Vulnerability Notes Database. Nov. 8, 2017. URL: `https://www.kb.cert.org/vuls/id/240311` (visited on 12/14/2017).

[20] Wendy Warne. *Bluetooth Low Energy - It starts with Advertising.* Bluetooth Technology Website. Feb. 15, 2017. URL: `https://blog.bluetooth.com/bluetooth-low-energy-it-starts-with-advertising?_ga=2.66364334.1220805425.1512865107-962131403.1512526995%5C` (visited on 12/13/2017).

[21] *Where To Find It — Bluetooth Technology Website.* URL: `https://www.bluetooth.com/what-is-bluetooth-technology/where-to-find-it` (visited on 12/13/2017).

[22] Sheng Zhou and J. K. Pollard. "Position measurement using Bluetooth". In: *IEEE Transactions on Consumer Electronics* 52.2 (May 2006), pp. 555–558. ISSN: 0098-3063. DOI: `10.1109/TCE.2006.1649679`.