A Low Energy Profile: Analysing Characteristic Security on BLE Peripherals

Pallavi Sivakumaran*
Centre for Doctoral Training in Cyber Security,
Royal Holloway University of London
pallavi.sivakumaran.2012@live.rhul.ac.uk

Jorge Blasco Alis
Information Security Group,
Royal Holloway University of London
jorge.blascoalis@rhul.ac.uk

ABSTRACT

Bluetooth Low Energy is a ubiquitous technology, with applications in the fitness, healthcare and smart home sectors, to name but a few. In this paper, we present an open-source Profiler for classifying the protection level of data residing on a BLE device. Preliminary results obtained by executing the tool against several devices show that some BLE devices allow unauthenticated reads and writes from third party devices. This could expose them to a number of attacks and compromise the privacy, or even the physical safety, of the device owner.

KEYWORDS

Bluetooth Low Energy, attribute security, pairing, Just Works, Passkey

ACM Reference Format:
Pallavi Sivakumaran and Jorge Blasco Alis. 2018. A Low Energy Profile: Analysing Characteristic Security on BLE Peripherals. In *CODASPY '18: Eighth ACM Conference on Data and Application Security and Privacy, March 19–21, 2018, Tempe, AZ, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3176258.3176945

1 INTRODUCTION

Bluetooth Low Energy (BLE) is a wireless data communication technology which is rapidly increasing in popularity. Its focus on low-power and low-cost devices has resulted in the technology being implemented in a variety of use cases, ranging from personal health and fitness devices, such as glucose monitors and fitness trackers, to home automation and security systems, such as smart locks and energy monitors.

Data on these devices may reveal personal information about their users, and may directly or indirectly enable critical functionality. As such, a range of attacks may be possible against data that is not suitably protected. Passive eavesdropping on the wireless interface, or unauthorised data access requests made directly to the device, violate a user's expectation of confidentiality. In addition, active MitM eavesdropping and data modification, as well as inducing unexpected behaviour via fuzzing, may all be viable attacks against unprotected data.

With the increased proliferation of BLE into everyday devices, there is a greater likelihood that attacks on devices will not only affect the devices' functionality, but also cause harm in the "physical world". For example, recent research into BLE-enabled hover-boards has shown that unprotected data on the device can be overwritten, enabling an attacker to gain control of the board and potentially cause injury to the user [3].

Mechanisms have been described in the Bluetooth specification [1] for protecting data on BLE devices by allowing communicating devices to authenticate themselves to each other. However, oftentimes these security measures are not implemented. This may be due to device constraints or increasing competition and the pressure to reduce time-to-market. Difficulty in navigating the somewhat complicated Bluetooth specification may also be a factor.

In this paper, we present a novel Profiler (Section 3) for identifying the minimum level of security applied to data on a low energy device, to assess the safety and reliability of device-resident data. This can then be translated to possible attacks that the device may be vulnerable to. The tool does not identify the default security level supported by the target device, but instead aims to determine the *lowest* possible level at which the data can be accessed. This is important because an attacker would not be confined to normal operating conditions or default security behaviour.

The Profiler may also be able to identify whether a static PIN code is used by the BLE device during its authentication process, by testing against a dictionary of commonly-used PIN codes. Static PINs are typically used when a device does not have the input/output capabilities required for dynamic PINs. However, they reduce the security of the authentication process, as a known value can simply be reused at a later time.

Our tool may be useful for developers to validate their security implementations, for security analysts as a starting point into their testing, or even for end users to determine how secure their data is.

We also present preliminary results obtained from executing our code against a number of test devices (Section 4). Finally, we briefly discuss limitations of our method, potential avenues for future work, and recommendations for improving BLE data security.

2 BLE BACKGROUND

BLE is a technology that enables the wireless transfer of small amounts of data between two devices. The devices operate in an asymmetric Central-Peripheral configuration, where typically the more resource-constrained will assume the role of Peripheral and the more powerful device will act as the Central. As an example, in

*The author was supported by the EPSRC and the UK government as part of the Centre for Doctoral Training in Cyber Security at Royal Holloway, University of London (EP/P009301/1)

CODASPY '18, March 19–21, 2018, Tempe, AZ, USA

ACM ISBN 978-1-4503-5632-9/18/03.
https://doi.org/10.1145/3176258.3176945

the case of a smart lock communicating with a mobile phone, the lock would function as the Peripheral and the phone as the Central.

Attributes: BLE devices store data as attributes, where attributes may be read or written if appropriate permissions are set. *Characteristics* are a type of attribute that contain the actual user or application data. Characteristics have associated *properties* that describe how the characteristic value can be accessed. Examples include "read", "write", "notify" and "indicate". Only "read" and "write" properties are considered for the purpose of this paper. Some characteristics on a device may need to be protected, and to access a protected characteristic, the two devices will typically need to communicate over an encrypted link. The keys required for link encryption are established during a process known as *pairing*.

Pairing: There are two methods or "generations" of pairing: LE Legacy and LE Secure Connections (LESC), where LE Legacy was the original pairing method for BLE and is still the most widely used. The pairing process also uses one of four *association models*: Just Works, Passkey Entry, Out-Of-Band (OOB) and Numeric Comparison. With the exception of Numeric Comparison, which can only be used with LESC, all association models can be used with both generations of pairing.

The level of protection applied to a connection depends on the method as well as the association model used for pairing. LESC uses FIPS-compliant Elliptic-Curve Diffie-Hellman for key generation and is generally considered to be the more secure of the two pairing methods, as the key establishment protocol in LE Legacy has been shown to be vulnerable to passive eavesdropping attacks [5].

Of the four association models, Numeric Comparison provides the most protection against eavesdropping and MitM attacks, but is rarely used. OOB requires a non-Bluetooth channel for transmitting keys and is also not widely seen in practice. The most commonly used association models are Passkey Entry and Just Works. Passkey Entry requires manual user intervention in terms of PIN code entry and is therefore considered to result in an "authenticated" key. Just Works uses an all-zero PIN code with no user input, and the resultant key is considered to be "unauthenticated". When used with LE Legacy, neither Just Works nor Passkey Entry provide protection against passive eavesdropping. However, Passkey Entry offers better protection against MitM attacks.

Attacks: A BLE device that has rudimentary or no authentication requirements may expose its readable characteristics' values to unauthorised third parties, which could be a violation of the device owner's privacy. Writeable characteristics are more critical, as they could be overwritten to cause unexpected behaviour and possibly harm to the owner. For example, overwriting a glucose measurement on a monitoring device may cause the user to think they needed an insulin shot when they did not, which could have serious consequences.

3 THE SECURITY PROFILER

We have developed a Profiler[1] for identifying the lowest level of security required to access the value of a characteristic attribute. The tool is written on top of a modified version of *noble*[2], an open-source Node.js implementation of a BLE Central device.

[1]https://github.com/projectbtle/att-profiler
[2]https://github.com/sandeepmistry/noble

The Profiler scans for and connects to a user-specified BLE Peripheral device. It then attempts to access (i.e., read or write) every applicable characteristic, where a characteristic is applicable if the access type is present in its properties set.

If access is denied because the characteristic is protected, then the tool will attempt to pair with the test device using the lowest level of security. If pairing is successful, then it will re-attempt characteristic access. If pairing fails, then the tool will increment the security level and re-attempt pairing. The Profiler will continue to increase security and attempt characteristic access until either all characteristics have been accessed or the highest level of security (as implemented in code) has been reached.

The Profiler has four such security levels: None - No security; Low - Unauthenticated pairing with 64-bit key; Medium - Unauthenticated pairing with 128-bit key; High - Authenticated pairing with 128-bit key. In our code, unauthenticated pairing corresponds to Just Works and authenticated pairing indicates Passkey Entry. The code attempts to reach these levels of security by using custom pairing requests with different security requirements and purported capabilities in each request. It should be noted that these are levels that are requested by the Profiler code, but, depending on the capabilities of the target device, some levels may not be achievable.

The output of the code is a JSON file containing details of all characteristics on the test device and the results from access attempts against applicable characteristics. Each such characteristic will have a security key which specifies the level of security at which the characteristic value was accessed (or an indication of why it could not be accessed), as well as the final pairing method and association model that were used.

Dictionary "attack" with static PINs: Some BLE devices utilise fixed PIN codes with the Passkey Entry model. This somewhat defeats the purpose of passkeys since a known fixed PIN can be entered programmatically, with no need for the user intervention that is required with dynamically generated PINs. The Profiler provides the option for trying to identify whether a BLE device uses a static PIN. It does this by making repeated pairing attempts using a dictionary of commonly used passkeys, derived in part from an analysis of six-character passwords [4]. If a match is found, then this will be indicated in the output file, but the PIN itself will only be available from the code execution terminal.

4 DEVICE TESTING

Preliminary tests were conducted against four fitness trackers of different prices and capabilities, a posture monitoring device, and a smart lock. All devices were under our ownership and control.

The number of characteristics on each device accessible at each security level has been graphed in Figure 1. The level "Unknown" on the graph refers to when a characteristic was not accessible due to a non-security related reason, while "Custom" indicates potential application-layer security.

The graphs show that 83% of the tested devices allowed at least one unauthenticated write. Further, three out of the six devices allowed all their characteristics to be read, and one also allowed all of its characteristics to be written, without any authentication required. These devices will therefore likely be vulnerable to unauthorised data read/writes and perhaps also MitM attacks.

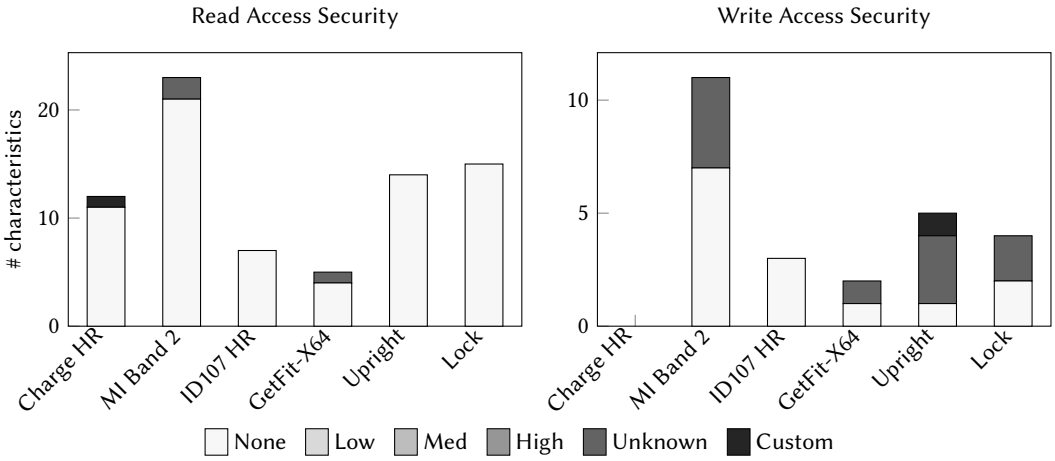

Figure 1: Security Levels for Read/Write Access

The devices that supported standard BLE pairing primarily used Just Works, which is the least secure association model, offering no MitM protection. Those that used Passkey Entry employed static, commonly-used PINs (000000 and 123456 were observed). This again would allow MitM attacks against these devices, if the attacker had previous knowledge of the static PIN.

The results appear to indicate that the Fitbit Charge HR has implemented application-layer security, which is consistent with previous findings [2]. Another fitness tracker, the Mi Band 2, responded to some write requests with an Application Error, which also may be indicative of protection applied at the application layer.

We also found that not all devices implemented the BLE specification exactly. While non-standard behaviour and services may reduce the amount of information that is immediately available to an attacker, they run the risk of being not as well defined or tested as standardised methods, which in turn could potentially result in security vulnerabilities.

5 DISCUSSION

5.1 Recommendations

We propose the following recommendations for BLE developers.

If the data stored in a characteristic is sensitive or enables critical functionality, developers should enforce security, particularly in the case of a writeable characteristic. Passkey Entry should be used at a minimum, preferably with dynamic passkeys. In the case of new developments, LESC with Numeric Comparison is preferred over LE Legacy pairing. This may require the use of new libraries and additional hardware. Further, the amount of data that can be accessed from the device should be reduced to the absolute minimum that is necessary for functionality, and standard, well-tested code or libraries should be used where possible.

5.2 Limitations and Future Work

While the Profiler can aid in understanding the security applied to some characteristics on a device, it has certain limitations which prevent it from testing the security of *all* characteristics. This is due to the characteristic properties mentioned in Section 2. The Profiler is currently able to test characteristics that have properties "read" and "write". However, a characteristic value can also be obtained via "notifications" and "indications" and can be written using a "write without response" method. These are not handled in the code at present, but will be included in a future version.

Another potential avenue for future work is to fuzz test unprotected writeable characteristics identified by the Profiler, to determine if unexpected behaviour can be elicited.

Further, in its present form, the Profiler analyses how much data can be read from or written to a resource-constrained BLE Peripheral by a Central device or Central emulator. We aim to also explore whether a Peripheral can access data that it is not authorised to, from a connected Central device.

6 CONCLUSION

This paper describes a custom profiling tool for determining the minimum level of security applied to characteristics on a Bluetooth Low Energy device. Preliminary tests against some consumer devices show that several of the devices allow data to be read, and in some cases written, by a connected device without the device first authenticating itself. This translates to potential attacks that could be detrimental to user privacy and, in some cases, safety. The paper also presents some recommendations for reducing the attack surface of a device. The code for our tool is freely available, and we welcome suggestions and execution results.

REFERENCES

[1] Bluetooth Special Interest Group. 2016. Bluetooth Core Specification v5.0. (2016).
[2] Britt Cyr, Webb Horn, Daniela Miao, and Michael Specter. 2014. Security Analysis of Wearable Fitness Devices (Fitbit). *Massachusetts Institute of Technology* (2014).
[3] Thomas Kilbride, James Thomas, and Stefan Boesen. 2017. Ninebot by Segway miniPRO Vulnerabilities. (2017). Retrieved September 21, 2017 from https://www.ioactive.com/pdfs/IOActive-Security-Advisory-Ninebot-Segway-miniPRO_Final.pdf.
[4] David Malone and Kevin Mahern. 2011. Investigating the Distribution of Password Choices. (2011). Retrieved from https://arxiv.org/pdf/1104.3722.pdf.
[5] Mike Ryan. 2013. Bluetooth: With Low Energy Comes Low Security. In *WOOT '13: 7th USENIX Workshop on Offensive Technologies, Washington, D.C., USA, August 13, 2013.*