Group 3
Luvneesh Mugrai (lm3300)
Matthew Cowell (mc7243)
Willie Yee (wy667)
Jacky Teoh (jt2908)

Battling BlueBorne: Enhancing the Security of Bluetooth Systems

Abstract **-** An Internet of Things (IoT) personal assistant smart speaker, such as Alexa or Google Home, is a machine that allows the user to connect to the internet to accomplish daily tasks with the help of Bluetooth audio commands. These daily tasks can include, but are not limited to, telling the user their local weather and news or the ability to control automated devices in the user's home remotely. These devices also function as smart speakers by allowing smartphones to connect via a Bluetooth protocol that is always enabled in the IoT personal assistant smart speaker. With Bluetooth constantly enabled, it exposes these devices to the threat BlueBorne, eight discovered vulnerabilities found in the implementation of Bluetooth [1]. In this paper, we will explore a solution in which we implement stricter authentication and authorization protocols to limit these vulnerability in Bluetooth that are specifically targeted by BlueBorne. Stricter authentication will prevent malicious users from easily gaining access and will prevent the malicious user from having full access in case of a breach.

Keywords - Authentication, Authorization, BlueBorne, Bluetooth Speakers, Exploit, Internet of Things, Personal Assistance Smart Speakers, Security Level, Security Mode, Vulnerabilities

I. Introduction (with Motivation)

Bluetooth enables a communication link through radio waves to be established between an IoT personal assistant smart speaker and a user's smartphone. Through this communication, the user is able to transmit data to the connected personal assistants such as smart speakers for playing back audio or music. To connect to the smart device, the phone's bluetooth has to be turned on and then paired with the smart device. However, there is no need to turn on Bluetooth, as smart speakers, including Amazon Alexa and Google Home, have Bluetooth permanently enabled by default. Leaving Bluetooth on indefinitely can become problematic as unwanted malicious users would be able to gain access and take control of the device, specifically via vulnerabilities exposed by BlueBorne [2][3].

We propose a solution that implements stricter **authentication** and **authorization** protocols for the personal assistant smart speaker devices. Our solution will implement an Out of Band security measure for ensuring safety when pairing devices, as well as using Security Level 4 which includes a form of Diffie-Hellman Key Exchange, ensuring that our methods of encrypting and sending data from device to device is also secure. Our solution focuses on mitigating the chances of a successful attack and in the case of an actual breach, we have taken preemptive measures by limiting usability to be only that which is intended when the Bluetooth is connected - to act as a smart speaker and to stream audio sent from the connected device.

II. Background

Since its beginning in 1998, Bluetooth had always been focused on the functionality and usability in order to provide the best experience to its users. This focus on usability, rather than security, was the key characteristic of Bluetooth's creation that led to it being as successful as it was, reaching 1 million devices by 2006, but would also lead to Blueborne attacks [4]. Based on the vulnerabilities discussed by Armis, the always-on functionality of the Bluetooth speakers is the main problem [5]. An effective solution to this problem would be to implement stricter authentication and authorization protocols to limit the unwanted user access and information theft.

While this solution might be easy to come up with, attempting to implement such a solution will prove to be quite difficult due to the amount of restrictions that authentication and authorization protocols introduce. For example, the most successful authentication solution would be to only enable Bluetooth speakers to listen to user conversations when a physical button on the device is pressed, assuming only authorized users have physical access to the device. However, while this would solve the problem at hand, this solution would create a new problem by limiting the Bluetooth speaker's functionality of voice control. Having this feature also defeats the purpose of having the "hands free" functionality of Bluetooth technology.

One new idea that had come out in the last few years that can provide a solution for this authentication and authorization problem of Bluetooth speakers is **Bluetooth Low Energy (BLE)** security modes. One aspect that we will focus on from BLE is allowing Bluetooth connections to be more securely established, by preventing unwanted individuals from accessing private information, such as through **man-in-the-middle attacks** and **eavesdropping** [6].

III. Threat Model

During our previous research we identified two key threats that plague Bluetooth speakers and other Bluetooth IoT devices. The first threat focused around the idea of the Bluetooth device constantly listening to the user and their sometimes private conversations. The problem with this is that unauthorized users can gain access to this device or even the communication by masquerading as the company that receives the information sent by the device. The second threat focused on a more common attack used on these devices. This attack is a stack overflow attack on the devices memory. While this problem is caused by a lack of security measures around the foundational memory and software of the device, this problem can be solved by authenticating users and limiting root access authorization to authenticated users.

The proposed solutions we had for preventing these key threats to Bluetooth devices would require a complete revamp of the Bluetooth systems or serious software patches that would require time and create more complexity for the systems. However, with the research and analysis of BLE security modes we can give a more simplistic and augmentable solution. The design of a new BLE Bluetooth system will be able to focus on enhancing the main vulnerabilities of the authentication and authorization of current Bluetooth speakers and other devices.

IV. Design (Our solution)

With Bluetooth Low Energy, different combinations of **Security Modes (SM)** and Levels can be used. There are 4 **Security Levels (SL)**. SL1 is the most basic of the SLs, and basically is a connection without any security at all. As the SL goes higher, more security gets added, such as encryption and mandatory pairing. SL4 is the highest level, and it supports encryption, pairing, and uses Elliptic Curve Diffie-Hellman (ECDHE or P-256), which is also Federal Information Processing Standards Publications (FIPS) compliant [9]. For Security Modes, SM1 is the level without the signing of data and SM2 is the level with the signing of data. The combination that is the most optimal is **Secure Connection Only Mode (SCOM)**. This mode is the combination of Secure Mode 1 and Security Level 4. This means that all incoming and outgoing data is encrypted, unsigned, and a pair is mandatory. The encryption type during the pairing will be ECDHE, which allows for secure public-private key pairing in insecure channels [7] [8].

SM2 can be used instead of SM1 to sign the data, but as all the data is encrypted this would expend more of the already small battery life. Therefore, SM1 with SL4 is an optimal BLE combination for security with respect to Bluetooth speakers and other such IoT Bluetooth devices.

Using Security Level 4 would be ideal because Diffie-Hellman is an extremely popular method of exchanging cryptographic keys. Encrypting traffic between our paired devices will further increase security as it renders man-in-the-middle attacks impractical. Using SL4 allows Bluetooth and other such applications to encrypt user data to prevent unauthenticated and unauthorized users and companies from reading data in plaintext. This allows SCOM to provide better security in the case of BlueBorne attacks.

In addition to establishing a secure key agreement protocol through ECDHE, there are four commonly used methods for pairing two bluetooth devices when sharing the encryption keys. The most secure out of the four is called **Out of Band (OOB)** [7]. Although it's a communication channel outside of Bluetooth, using near field communications (NFC) technology, the information will be secure as long as the NFC channel is secure. However, with ECDHE, even if the channel becomes compromised the data transmitted during the pairing will still be secure through the encryption. Through a public key exchange, OOB will be able to prevent eavesdropping and man-in-the-middle attacks. An example of the OOB pairing method would be with the Apple Watch. When the watch is looking for something to pair with, swirling dots appear on the screen. The user then points the face of the watch towards the other device, and the connection is established. Information can then be sent and received from this connection.

V. Implementation

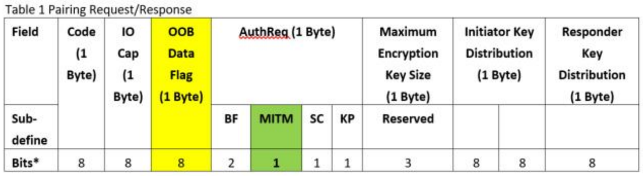
Based on our research we have shown that the best design for enhancing Bluetooth's security against attacks, like BlueBorne, is with ideas like BLE. This can be done by implementing a high security level based on different aspects of encryption, pairing, and other SCOM measures.

We will implement the pairing mechanic with OOB, while using SL 4, in a two-part procedure to authenticate the device at the other end. The purpose of the two-part authentication is to ensure that in case the OOB channel is compromised, there is still another layer of security, which comes from the ECDHE and user authorization.

The first part will be triggered when a device attempts to pair with the IoT smart speaker after it has already been connected to once. The user is notified of a new pairing trying to be established with the IoT smart speaker, the name and the kind of device. The user must then accept the notification on their smartphone synced app. For example, if implemented in Amazon's Alexa, when a device would try to pair with the Alexa via Bluetooth, the synced Alexa app on the owner's smartphone will display a notification asking the user to confirm or deny allowing Alexa to pair with the device. Once accepted, the second part of the procedure will take place. The pairing using the NFC will have the smart speaker exchange and verify a transfer key with the pairing device.

When working with Out of Band pairing, even though we are securing a connection via a channel outside of Bluetooth, we still have to comply with the bluetooth specifications for packet transfer, as shown in Table 1 [10].

Table 1 Pairing Request/Response

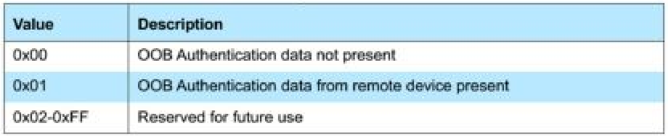


Field	Code (1 Byte)	IO Cap (1 Byte)	OOB Data Flag (1 Byte)	AuthReq (1 Byte)				Maximum Encryption Key Size (1 Byte)	Initiator Key Distribution (1 Byte)		Responder Key Distribution (1 Byte)
Sub-define				BF	MITM	SC	KP	Reserved			
Bits*	8	8	8	2	1	1	1	3	8	8	8

*Bit order is LSB to MSB.

The table requires a OOB Data Flag, which is one byte and contains one of three flags. Byte 0x00 means that there is no OOB Authentication data. Byte 0x01 means that the authentication data is present as shown by Table 2 [10].

Table 2 definition of OOB Data Flag



Value	Description
0x00	OOB Authentication data not present
0x01	OOB Authentication data from remote device present
0x02-0xFF	Reserved for future use

It is important to note that in order to use OOB pairing, both the receiver and the initiator must have their respective OOB Data Flag set. If either of the receiver or the initiator (or both) do not have this flag set, OOB pairing will not work, and Bluetooth will default to using the other 3 methods of pairing, including the unsafe "Just Works" method.

Then, in order to validate authenticity, there are two pseudorandom numbers generated by the initiator and the receiver. These random numbers are passed through a Bluetooth specification cryptographic toolbox, which then returns two new strings of bytes. The initiator can then verify the authenticity of the receiver, and the receiver can verify the authenticity of the initiator [10]. Upon a successful verification, the IoT smart speaker will be paired with the device that initiated the pairing.

VI. Evaluation

Our proposed solution uses two levels of security during the pairing process, the OOB channel combined with ECDHE encryption. The OOB provides support from man-in-the-middle attacks, under the assumption that the channel is secure. On the chance that the channel becomes compromised, ECDHE ensures that the data we transmit over the network is still encrypted.

We understand that there are both pros and cons to the solution we have presented in this paper. On the pro side, our solution does not just focus on smart speakers and their design. This allows for our solution to be extendable to other IoT devices that use Bluetooth. We also do not compromise or limit any of the features that users want to use and enjoy. Some alternate solutions include adding physical buttons on the speakers as a method of two-factor authorization. As a result of our research, we learned that these physical solutions not only do hardly anything in terms of keeping attackers out, they also hinder the user experience that these smart speakers advertise and market towards their customers.

Looking over our approach and solution, there are still areas in which it is weak as it is not a fully exhaustive solution. Our solution does not prevent all possible attacks on smart speakers, but instead makes it so that malicious users must go through a lot more effort in order to successfully breach the new security mechanisms. For example, a malicious user might be able to gain access by abusing a stack overflow attack, in which the stack overwrites malicious code so that the return pointer points to any code the malicious user might want, including code that will gain full control of the smart speaker [11]. However, if a malicious user were to attempt to attack the smart speaker in this manner, due to the encryption of data as a result of ECDHE, the malicious user would have to find a way to decrypt the private data, which is quite the tall order.

Another more effective way to evaluate our solution would be to compare pre-BLE Bluetooth with BLE Bluetooth. This can be done by measuring each type in a variety of way, which can include the number of attacks stopped and the time it takes for a successful attack to be implemented.

VII. Related Solutions

One alternative approach would be to limit the actual availability of the Bluetooth, so that the Bluetooth can be toggled on and off from the synced Alexa app on the owner's smartphone. In addition to the toggle, there would a predefined interval during which the Bluetooth will be available to be paired to on the IoT smart speaker. This would limit the effectiveness of an attack, such as the one used in BlueBorne by limiting the speaker's ability to constantly listen to private use conversation and by preventing the device from being a constantly exposed target.

Another approach would be to have each request, after the first one, get added to a pool of requests which can then be authorized through a web, mobile, or even physical interface. This would allow for two-factor authentication for Bluetooth devices, thus making authentication stricter and more secure. This would also enable more controlled authorization on the part of the user, by giving then control of the system, rather than the system trying to authorize unknown parties in the traditional user to user fashion. By utilizing a central device such as a computer to handle all the requests, we organize and display possible threats in a user-friendly and easy to understand solution.

VIII. Conclusion

For this paper we focused on implementing stricter authentication and authorization protocols based on Bluetooth Low Energy modeled ideas with the intention to mitigate the risk of attacks through BlueBorne vulnerabilities. While we do not directly solve the vulnerabilities BlueBorne exploits once it enters into the IoT personal assistant smart speaker, we do add layers of prevention in order to make it harder for the malicious user to gain access through Bluetooth.

Our proposed solution uses the Secure Connection Only Mode model from BLE to ensure there are only authenticated connections and encrypted communications between the devices. This leads to implementing an OOB channel to establish a secure communication. That way, in the case of a breach we use ECDHE for the key exchange protocol and then encrypt the data we need to be transferred. The OOB channel prevents the unauthorized users from accessing the Bluetooth device by having a secure channel with a public key exchange. Finally, the flexibility of our solution allows for it to be used on Bluetooth devices, as well as other IoT devices, and increase the vulnerabilities that BlueBorne exploits in its attacks.

IX. References

1. Slashdot discusses BlueBorne and the vulnerabilities of Bluetooth devices https://it.slashdot.org/story/17/11/17/00372
2. Slashdot discusses BlueBorne and the vulnerabilities of Bluetooth devices https://it.slashdot.org/story/17/11/17/0037251/bluetooth-hack-affects-20-million-amazon-echo-google-home-devices
3. BlueBorne infecting over 5 million Bluetooth enabled devices https://mobile.slashdot.org/story/17/09/12/2030213/blueborne-vulnerabilities-impact-over-5-billion-bluetooth-enabled-devices
4. Bluetooth’s own history page about their product https://www.bluetooth.com/about-us/our-history
5. Armis’ short, white paper outline of the security vulnerabilities of Bluetooth https://go.armis.com/hubfs/BlueBorne%20Technical%20White%20Paper.pdf
6. An in-depth, but easy to understand article on an introduction to BLE https://www.digikey.com/eewiki/display/Wireless/A+Basic+Introduction+to+BLE+Security
7. An article about understanding Bluetooth Security and its design https://www.duo.com/decipher/understanding-bluetooth-security
8. Used for explaining the purpose of Elliptic Curve Diffie-Hellman cryptography http://www.secg.org/sec1-v2.pdf
9. NIST explanation of the FIPS (use, implementation, etc.) https://www.nist.gov/itl/itl-publications/federal-information-processing-standards-fips
10. Overview explanation of OOB implementation https://blog.bluetooth.com/bluetooth-pairing-part-5-legacy-pairing-out-of-band
11. List of Blueborne vulnerabilities and their descriptions https://armis.com/blueborne/#/technical