Security Analysis of Internet-of-Things: A Case Study of August Smart Lock

Mengmei Ye Nan Jiang Hao Yang Qiben Yan*
Department of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, NE 68588-0115, USA
*Corresponding author, Email: yan@unl.edu

***Abstract*—To realize the vision of Internet-of-Things (IoT), numerous IoT devices have been developed for improving daily lives, in which smart home devices are among the most popular ones. Smart locks rely on smartphones to ease the burden of physical key management and keep tracking the door opening/close status, the security of which have aroused great interests from the security community. As security is of utmost importance for the IoT environment, we try to investigate the security of IoT by examining smart lock security. Specifically, we focus on analyzing the security of August smart lock. The threat models are illustrated for attacking August smart lock. We then demonstrate several practical attacks based on the threat models toward August smart lock including handshake key leakage, owner account leakage, personal information leakage, and denial-of-service (DoS) attacks. We also propose the corresponding defense methods to counteract these attacks.**

I. Introduction

Increasingly, the Internet-of-Things (IoT) devices have been widely used in our lives, and have brought unprecedented convenience. The benefits provided by IoT begin to transform everything from businesses, governments to homes, hospitals around the world [1]. Specifically, the smart home appliances, as an essential part of IoT devices, have been extremely popular in the IoT market, and the functionalities of which have become increasingly specialized and powerful. For instance, we can turn on/off the lights by speaking to the air using voice-controlled speaker, or control the home coffee machine remotely using the mobile device to make beverages without physically touching the machine.

However, despite these benefits IoT provides, the IoT devices also bring a wide range of emerging security issues, including the potential of damaging physical systems, industrial outage, and privacy leakage [2]. Meanwhile, there exist a number of potential vulnerabilities in smart home appliances that greatly threaten personal safety and data privacy [3].

In this paper, we primarily focus on the security analysis of a popular smart lock, namely August smart lock [4]. There are numerous types of IoT devices in the market, we choose to investigate August smart lock due to the following reasons: 1) the smart home appliances play significant roles, which are closely intertwined with user experience and usable security; 2) among all the smart home appliances, the security of smart locks is widely concerned by customers, because nothing in the home will be protected if the lock gets hacked; 3) the August smart lock leverages bluetooth to connect with a mobile app, and users control the smart lock through the mobile app, which is the most popular way of managing smart home appliances; 4) the August smart lock has been widely adopted in smart homes and integrated to work with other popular platforms such as Amazon Alexa, Samsung SmartThings and Airbnb; and 5) the August Smart Lock app has been updated very frequently, which makes it challenging to hack the lock and the corresponding app.

In this paper, we demonstrate the following attacks toward August smart lock:

- *Handshake Key Leakage Attack*: in which the attacker is able to steal the handshake key from the smart lock, and illegally and covertly control the lock using a third-party device;
- *Owner Account Leakage Attack*: in which the attacker is able to disguise himself/herself to be the owner, by logging into the lock owner's account in the third-party device to control the smart lock without being discovered;
- *Personal Information Leakage Attack*: in which the attacker is able to obtain the lock user information, which seriously threatens the user privacy; and
- *Denial-of-service (DoS) Attack*: in which the attacker disrupts the regular usage of smart lock, which dramatically brings down the user experience.

The rest of this paper is organized as follows: we first review the related work for the common attacks toward smart home appliances, especially smart locks, in Section II. In Section III, we provide an overview on the August smart lock system. In Section IV, we present the attacks toward August smart lock, and illustrate the potential defense mechanisms. In addition, we generally discuss about the security analysis of IoT devices, focusing on analyzing their mobile apps, mobile system and the smart home appliance hardware in Section V. Finally, we conclude our paper in Section VI.

II. Related Work

A large number of smart home appliances are in the market today, as shown in Table I. Some of them focus on extending the functionality of a specific home equipment to improve the usability, while others are able to connect, monitor and control the home equipments to provide automation and convenience to our lives.

978-1-5386-2784-6/17/$31.00 ©2017 IEEE

TABLE I
POPULAR SMART HOME APPLIANCES ON MARKET [5]

Appliances	Descriptions
Samsung SmartThings [6]	Dozens of smart apps controlled by SmartThings Hub to monitor the house for the security purposes
Amazon Echo [7]	Wireless and voice-controlled speakers that can control smart home equipments or provide useful information
Philips Hue [8]	Wireless-controlled indoor lighting for convenience
Nest [9]	WiFi-controlled devices to ensure the home security using the smart camera, and monitor the room temperature using the smart thermostat

TABLE II
SUMMARY OF ATTACKS ON SAMSUNG SMARTTHINGS [3]

Attacks	Descriptions
Backdoor Pin Code Injection Attack	Feeding the OAuth token to the SmartApp, and injecting the command by OAuth to compromise the mobile smart app that uses Groovy dynamic method invocation
Door Lock Pin Code Snooping Attack	Eavesdropping or leaking the device identifier from the battery monitor to attack the SmartApp
Disabling Vacation Attack	Interfering the SmartApp and disabling the protection set up on the vacation mode
Fake Alarm Attack	Sending fake events, such as sounding the alarm, to misguide the user

A. Attacks Toward Smart Home Appliances

Table II lists four types of attacks on the smart home appliances discovered by Fernandes et al. [3]. They mainly focus on the attack demonstrations on Samsung SmartThings platform.

In addition, Hernandez et al. [10] demonstrate the threat models on the smart Nest thermostat, and provide a security solution on this hardware platform. They analyze the security vulnerabilities of this smart thermostat, and compromise the Nest system remotely to spy on the house activities through wireless networks. They suggest enhancing the security of the bootloader authentication to defend against the attacker exploiting such vulnerabilities.

B. Attacks Toward Smart Locks

Recently, some researchers have been focusing on the security of smart lock. Rose et al. [11] investigate the security of bluetooth-enabled smart locks, and demonstrate some critical vulnerabilities of various smart locks. They find that the old version of the August smart lock has hard-coded secret key in the application source code; the Kwikset Kevo smart lock [12] leverages the strong security techniques on the bluetooth protocol, but the physical lock contains serious vulnerabilities making it easily compromised, which only takes 10 seconds; the QuickLock smart lock [13] does not encrypt the passwords and sends the password to the user who forgets the password in *plaintext*; and also the iBluLock smart lock [14] only requires 6-character password, which is vulnerable against the brute forcing attack.

TABLE III
ADDITIONAL ATTACKS FOR SMART LOCKS [18]

Attacks	Descriptions
Physically-present Attack	Physically performing the attack for the user who forgets to lock the smart lock
Revoking Attack	Performing the attack from the user who had the legal accessing before, such as the Airbnb tenant, or the household worker
Stealing Attack	Performing the attack as the thief, and stealing the user device to control the smart lock
Relaying Attack	Performing the attack by two attackers to relay the data for interfering with the smart lock control

Besides the research work on the vulnerabilities of the bluetooth-enabled smart locks, the blog from jmaxxz [15] particularly focuses on the August smart lock and further illustrates several serious flaws on it, including: the August smart lock does not perform the 2-factor authentication properly, and the hackers compromising the user email and text message could illegally control the lock [16]; the August smart lock does not perform the password reset process properly, and the attackers can easily figure out the true verification code for resetting any passwords [17].

In addition, Ho et al. [18] claim four additional attacks as shown in Table III. Furthermore, they evaluate the security challenges and primarily focus on designing the countermeasures against the physically-present attack and relaying attack on the smart lock mechanisms.

Different from all the previous work, our work constitutes a comprehensive case study on the security of August smart lock. Based on the August system architecture, we identify and validate new security threats toward the smart lock, and provide the potential protection mechanisms at different levels, such as securing the mobile smart lock app, patching the flaws on mobile operating system, and enhancing the security of the smart lock hardware system.

III. OVERVIEW OF AUGUST SMART LOCK SYSTEM

The August smart lock system consists of three components: August smart lock, August mobile app, and August remote server. The workflow is illustrated in Figure 1. The August smart lock communicates with the August app using Bluetooth low energy (BLE) protocol, and the user is able to operate the mobile app to control the smart lock. In addition, the August server synchronizes with the August mobile app for authenticating and conducting the lock control.

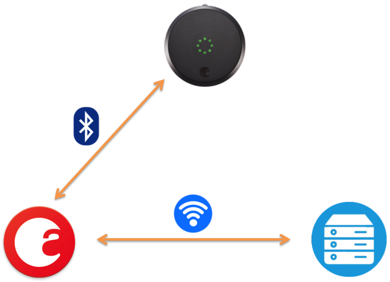


Fig. 1. System Workflow for August Smart Lock

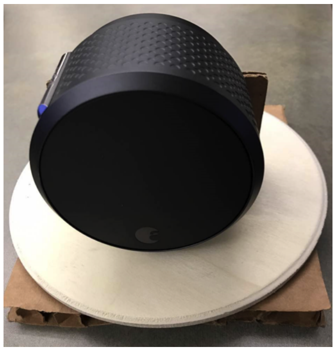
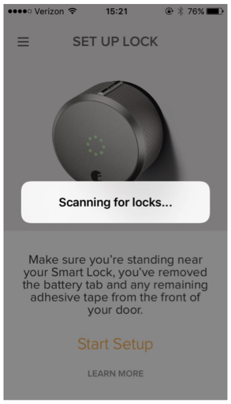


Fig. 2. August Smart Lock Deployment

Figure 2 shows the hardware of the August smart lock. There is a gravity sensor embedded inside the lock. Therefore, the lock must be held vertically to enable the regular usage. An official smart lock mobile app is provided by August company for both Android and iOS platforms. We create a user account and log into the app to setup the lock, as shows in Figure 3.

Basically, there are two types of user levels defined in the app, for users who are able to operate the smart lock on the app, namely the owner and guest, and the operation permission for different users is shown in Table IV. Lock/Unlock door is the most basic operation, and both owner and guest are able to control the door by using the app. Lock activity shows all the activity history including the user who locked/unlocked the door with the specific timestamp, and the updated status for guest list. The guest list shows all user information including user profiles and user levels. The user invitation is the function to invite new users. The user level control is used to update user role (i.e., owner, or guest) by owner. User permission control is the function to set the specific time slot for guests to operate the lock. From Table IV, we note that the owner owns the highest authority, and he/she is able to perform all the operations.



●●●●○ Verizon 15:21 76%

SET UP LOCK

Scanning for locks...

Make sure you're standing near your Smart Lock, you've removed the battery tab and any remaining adhesive tape from the front of your door.

Start Setup

LEARN MORE

Fig. 3. August Smart Lock Setup in Mobile App

TABLE IV
August Smart Lock Operations for Different User Levels

	Owner	Guest
Lock/Unlock Door	✓	✓
Lock Activity	✓	
Guest List	✓	
User Invitation	✓	
User Level Control	✓	
User Permission Control	✓	

Attack Model. In the entire August smart lock system, the attacker can target the functions on any of the aforementioned system components. At the network level, if the attacker installs a bluetooth jammer nearby the smart lock, it will seriously affect the normal communications between the lock and the legitimate mobile device. Also, at the mobile app level, the attacker is able to either fake the official smart lock app or use a malicious app to steal the users' private information. The vulnerabilities of the mobile app can be exploited to escalate the privilege of malicious users, compromise benign users' privacy, and disrupt the normal operations. Mobile apps are usually connected to a remote server for command and control, and data management. To avoid the detection by the remote server, the attacker can use a third-party device to control the lock without any data transmission or synchronization on the remote server. In the next section, we demonstrate four attacks with respect to the proposed attack model, and elaborate the suggested defense strategies for the different components of the smart lock system.

IV. Attack and Defense Strategies for August Smart Locks

In order to launch the attacks, the attacker requires a rooted/jailbroken mobile device, so that he/she is able to illegally access the xml files that stores the secret data, such as handshake key, user account and personal information from it, as Figure 4 illustrates. After the attacker obtains these secret information, he/she is able to further control smart lock, and perform malicious operations toward the smart lock. In the following sections, we propose and validate several attacks toward August smart locks.

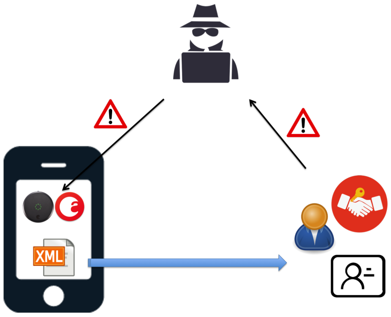
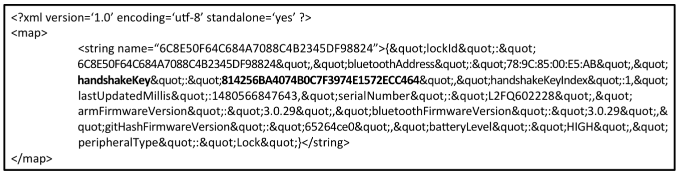


Fig. 4. Attack Workflow for August Smart Lock



```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
        <string name="6C8E50F64C684A7088C4B2345DF98824">{&quot;lockId&quot;:&quot;
        6C8E50F64C684A7088C4B2345DF98824&quot;,&quot;bluetoothAddress&quot;:&quot;78:9C:85:00:E5:AB&quot;,&quot;
        handshakeKey&quot;:&quot;814256BA4074B0C7F3974E1572ECC464&quot;,&quot;handshakeKeyIndex&quot;:1,&quot;
        lastUpdatedMillis&quot;:1480566847643,&quot;serialNumber&quot;:&quot;L2FQ602228&quot;,&quot;
        armFirmwareVersion&quot;:&quot;3.0.29&quot;,&quot;bluetoothFirmwareVersion&quot;:&quot;3.0.29&quot;,&
        quot;gitHashFirmwareVersion&quot;:&quot;65264ce0&quot;,&quot;batteryLevel&quot;:&quot;HIGH&quot;,&quot;
        peripheralType&quot;:&quot;Lock&quot;}</string>
</map>
```

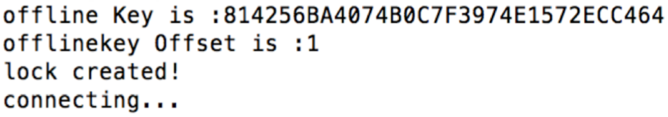
Fig. 5. Example File Exposing Handshake Key

A. Handshake Key Leakage Attack and Defense

The handshake key leakage attack is a fatal attack for the August smart lock, because the handshake key is leveraged by the lock and the lock app to communicate with each other. In other words, the handshake key is the most significant secret, and plays a vital role during transmission. Only the authorized user, who accesses the lock app with the matched handshake key, is able to control the smart lock. The smart lock ignores the requests from the unmatched handshake key. However, because the rooted/jailbroken device exposes the system files of the lock app, the attacker is able to hack the host's mobile device to obtain the system files that contain the handshake key. After obtaining the cleartext handshake key, he/she is able to stealthily unlock the door within the smart lock bluetooth range by a third-party device, which seriously threatens the safety of users.

In the August smart lock app, there are no cryptographic techniques leveraged to protect the handshake key stored in the owner's mobile device. The owner's handshake key is presented in plaintext format in the system file. In particular, for the rooted Android mobile phone, the handshake key can be found under the path: `/data/data/com.august.app/shared_prefs/PeripheralInfoCache.xml`. For the jailbroken iPhone, the handshake key can be found under the path: `/Applications/August/Library/Preferences/com.august.iossapp.plist`. The content of the system file is shown in Figure 5, and an example of the handshake key is depicted in Figure 6.

After obtaining the handshake key, the attacker is able to launch the August Smart lock by executing the lock control program posted in augustctl Github repository [19] without using the official mobile app. Even worse, there is no records showing in the host's app that the attacker locks/unlocks the door using the control program. In other words, nobody, except the attacker, would know the door is locked or unlocked. The entire process for this attack includes extracting the handshake key, creating a connection with the smart lock, and locking/unlocking the door that uses the control program, which takes only around *20 seconds*.



```
offline Key is :814256BA4074B0C7F3974E1572ECC464
offlinekey Offset is :1
lock created!
connecting...
```

Fig. 6. Handshake Key Example

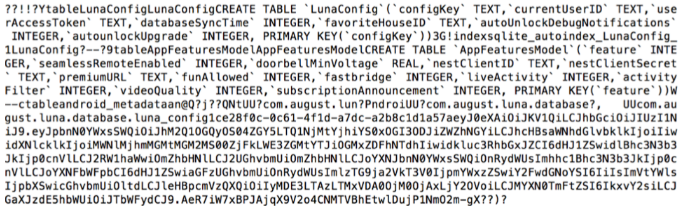
The smart lock is vulnerable to the handshake key leakage attack resulting from the constant and plaintext handshake key stored in the system files. To prevent the handshake key leakage attack, the handshake key is necessary to be protected by the state-of-the-art crypto-system before being stored in the system, so that the attacker is not able to obtain the plaintext handshake key directly from the mobile device. In addition, the mobile device communicates with smart lock by utilizing direct bluetooth pairing, which is only based on the constant handshake key. Therefore, to prevent the attacker from obtaining the handshake key and further controlling the lock, we suggest to leverage the secure communication protocol to ensure the authentication of lock controlling requests. That is to say, even if the attacker steals the handshake key, he/she is still not able to control the lock on his/her device because of the communication authentication. The secret handshake scheme introduced by Balfanz et al. [20] can be employed, which leverages pairing-based crypto-system to realize the authenticated communication between the legitimate mobile device and smart lock.

B. Owner Account Leakage Attack and Defense

The owner account leakage attack is the one revealing the user account in system files. Specifically, the attacker is able to import the system files into the lock app to be a faked owner, and further control the smart lock. The consequences of this type of attack are also extremely severe. Once the attacker is able to pretend to be a faked owner, he/she is able to access the owner's account, and further perform all the operations mentioned in Table IV, such as controlling the owner's door lock and manipulating the guest list.

In particular, the owner's sensitive information is stored in the system files as xml format, namely, databases and shared preferences in owner's mobile phone. In our experiment, we create a new user account first in the August app, say Eva. Also, there is an original owner account, say Alice. Then we assume that Eva illegally obtains the system files in Alice's mobile phone. Figure 7 shows the system file including Alice's sensitive account information in xml script, where we will be able to obtain user access token, database sync time, favorite house ID, primary key, etc.

To launch this attack, we first login using Eva's account. The app then shows that there is no available lock that can be controlled because it is a brand new account. We further



```
??!!?YtableLunaConfigLunaConfigCREATE TABLE `LunaConfig`(`configKey` TEXT,`currentUserID` TEXT,`use
rAccessToken` TEXT,`databaseSyncTime` INTEGER,`favoriteHouseID` TEXT,`autoUnlockDebugNotifications`
 INTEGER,`autounlockUpgrade` INTEGER, PRIMARY KEY(`configKey`))3G!indexsqlite_autoindex_LunaConfig_
1LunaConfig?--?9tableAppFeaturesModelAppFeaturesModelCREATE TABLE `AppFeaturesModel`(`feature` INTE
GER,`seamlessRemoteEnabled` INTEGER,`doorbellMinVoltage` REAL,`nestClientID` TEXT,`nestClientSecret
` TEXT,`premiumURL` TEXT,`funAllowed` INTEGER,`fastbridge` INTEGER,`liveActivity` INTEGER,`activity
Filter` INTEGER,`videoQuality` INTEGER,`subscriptionAnnouncement` INTEGER, PRIMARY KEY(`feature`))W
--ctableandroid_metadataan@Q?j??QNtUU?com.august.lun?PndroiUU?com.august.luna.database?,   UUcom.au
gust.luna.database.luna_config1ce28f0c-0c61-4f1d-a7dc-a2b8c1d1a57aeyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1N
iJ9.eyJpbnN0YWxsSWQiOiJhM2Q1OGQyOS04ZGY5LTQ1NjMtYjhiYS0xOGI3ODJiZWZhNGYiLCJhcHBsaWNhdGlvbklkIjoiIiw
idXNlcklkIjoiMWNlMjhmMGMtMGM2MS00ZjFkLWE3ZGMtYTJiOGMxZDFhNTdhIiwidkluc3RhbGxJZCI6dHJ1ZSwidlBhc3N3b3
JkIjp0cnVlLCJ2RW1haWwiOmZhbHNlLCJ2UGhvbmUiOmZhbHNlLCJoYXNJbnN0YWxsSWQiOnRydWUsImhhc1Bhc3N3b3JkIjp0c
nVlLCJoYXNFbWFpbCI6dHJ1ZSwiaGFzUGhvbmUiOnRydWUsImlzTG9ja2VkT3V0IjpmYWxzZSwiY2FwdGNoYSI6IiIsImVtYWls
IjpbXSwicGhvbmUiOltdLCJleHBpcmVzQXQiOiIyMDE3LTAzLTMxVDA0OjM0OjAxLjY2OVoiLCJMYXN0TmFtZSI6IkxvY2siLCJ
GaXJzdE5hbWUiOiJTbWFydCJ9.AeR7iW7xBPJAjqX9V2o4CNMTVBhEtwlDujP1NmO2m-gX??)?
```

Fig. 7. Leaked Owner Account Example

terminate the app and copy Alice's system files to replace Eva's. When we launch the app again, we are able to log in as Alice and obtain all the information and permission that Alice has.

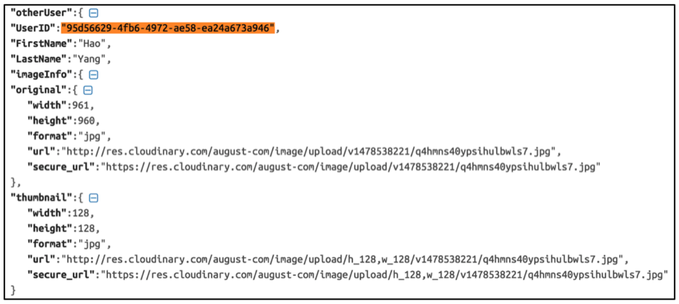
The owner account leakage attack exposes the vulnerabilities of both the mobile systems and the August app. First, once the mobile device is rooted/jailbroken, any secret system files are fully visible for any applications, i.e., nothing is secure and protected in a rooted/jailbroken device anymore. Second, the August app does not apply any security protections for the system files, so that any applications are able to visit and transfer these files easily. Therefore, to prevent this attack, an authentication mechanism to safeguard the system files should be applied, which ensures that only the authorized applications are able to obtain these files. FlaskDroid [22] provides a policy-driven mandatory access control for Android system, which can be applied to protect the access to sensitive data stored in Android devices. We plan to implement FlaskDroid for file access control in our future research.

C. Personal Information Leakage Attack and Defense

The personal information leakage attack is the attack that contributes to privacy invasion for users. The personal information leakage can result in a series of security issues. The attacker is able to use the leaked information to infer the comprehensive personal information. Also, the attacker is able to further track the bank account information and family member information, and so on, which are harmful for both user privacy and security.

Once the Android phone/iPhone is rooted/jailbroken, the current user information under the August app database folder of user's mobile phone is exposed without any protections, as Figure 8 shows. By accessing the system file that stores the current user profile, the attacker is able to obtain the user name and profile photo, etc.

Because the rooted/jailbroken mobile devices result in the insecure system files, to prevent the personal information leakage, the personal information is necessary to be protected by the state-of-the-art crypto-system before being stored in the system, so that the attacker is not able to obtain the sensitive information in plaintext directly from the mobile device. With an encrypted file system, the attackers will have to find the secret key for decryption, which increases their difficulties. Wang et al. [21] design an optimized encryption filesystem to guarantee both security and tolerance performance, which is able to protect system files stored in the rooted devices.



```
"otherUser":{ ⊟
"UserID":"95d56629-4fb6-4972-ae58-ea24a673a946",
"FirstName":"Hao",
"LastName":"Yang",
"imageInfo":{ ⊟
"original":{ ⊟
   "width":961,
   "height":960,
   "format":"jpg",
   "url":"http://res.cloudinary.com/august-com/image/upload/v1478538221/q4hmns40ypsihulbwls7.jpg",
   "secure_url":"https://res.cloudinary.com/august-com/image/upload/v1478538221/q4hmns40ypsihulbwls7.jpg"
},
"thumbnail":{ ⊟
   "width":128,
   "height":128,
   "format":"jpg",
   "url":"http://res.cloudinary.com/august-com/image/upload/h_128,w_128/v1478538221/q4hmns40ypsihulbwls7.jpg",
   "secure_url":"https://res.cloudinary.com/august-com/image/upload/h_128,w_128/v1478538221/q4hmns40ypsihulbwls7.jpg"
}
```

Fig. 8. Leaked Personal Information Example

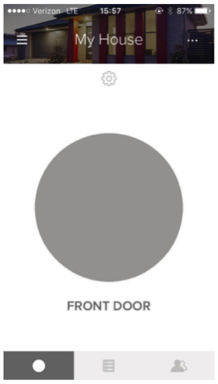


Fig. 9. The Interrupted Status on the August Lock App

D. Denial-of-Service (DoS) Attack and Defense

The DoS attack is able to interfere with the signal receiving, influence the regular usage of users, and even crash the application. For the August smart lock, the communication between the smart lock and the app is using Bluetooth. When there are multiple user applications connecting to the smart lock, it will suspend the app, and no user is able to lock/unlock at the same time. Actually this is supposed to be a solution to avoid the August app crashing, in case there are multiple requests controlling the August lock simultaneously. However, this also brings opportunities for the attacker to launch the DoS attack. Because the lock ID is stored in the system file of rooted/jailbroken Android phone/iPhone in plaintext, the attacker can easily keep sending the signal to the smart lock, and interrupt the connection with the normal user app and the smart lock.

For instance, there are two users: Alice and Eva. We assume that Eva already has some critical information about the lock. Therefore, she is able to communicate with the lock and control the lock. If Eva keeps sending lock/unlock command to the lock, the lock will not give any responses to Alice's operations. Figure 9 illustrates the interrupted status shown on Alice's app interface. Because of the DoS attack, the lock/unlock button turns grey, which means unavailable lock status. In the regular case, the button should be red or green to indicate the lock/unlock status.

Because the DoS attack is always hard to completely defend against, we can develop a potential solution to mitigate this kind of attack. The suggestion for the August smart lock is to provide a simple priority-based request control mechanism. For example, the smart lock should only process the requests from the authorized party. In other words, only the requests sent from the official August app can be safely accepted by the smart lock. Also, for the authorized users, namely the lock owners and guests, the owners are supposed to have the highest priority to control the lock using the August app. According to the defense strategy mentioned in subsection IV-A, the communication authentication also facilitate the priority-based request control.

V. Discussions on Smart Home Device Security

Based on the attack and defense strategies for the August smart lock in Section IV, we believe that these attack and defense strategies are more generic, and not only limited to the particular August smart locks.

A. Mobile Apps of Other Types of Smart Home Appliance Systems

The mobile app is always considered as the most vulnerable component on the security of the smart home systems. First, human developer is impossible to implement an absolutely flawless software interface for controlling the smart home devices. Second, some users who are lacking in the security usable skills are easily trapped into a security crisis deliberately posed by the attacker. For instance, the attacker is able to create a fake mobile app that pretends to be an official app to misguide the user to leak their private information unintentionally. Therefore, the smart home appliance should utilize an effective authentication mechanism to correctly identify and authorize the communication requests from legitimate mobile apps, while dropping the requests from faked apps. We plan to conduct a more in-depth investigation of such authentication mechanisms in our future work.

B. Mobile System Components Related to Smart Home Appliances

Besides the security of the mobile apps for the smart home appliances, the security of mobile operating platforms, namely Android, and iOS, also plays a vital role on the smart home appliance protection. Some of the known flaws on the mobile operating systems are discussed on Section IV, for example, the system files are not protected on the rooted/jailbroken Android/iOS. Here, we advocate the protection of system and apps' critical files on rooted/jailbroken devices for the purpose of protecting user privacy, even when users root their devices.

C. Smart Home Appliance Hardware

Besides the vulnerability of smart home appliances on the software level, we also need to consider the security of hardware in smart home appliances. Even though a smart home appliance leverages the strongest techniques on the software level, a vulnerable hardware on the smart home appliance is also able to contribute to a security disaster for the users. The security of the smart home appliance hardware is also influenced by the communication techniques with the mobile devices. For instance, a bluetooth-enabled hardware is vulnerable to the bluetooth signal interfering attacks. Thus, protections should be provided at the hardware level.

VI. Conclusion

In this paper, we investigated the security vulnerabilities on the smart home appliances from examining August smart lock by leveraging reverse engineering. We analyzed the security of the August smart lock system comprehensively, and demonstrated four types of attacks toward the system by exploiting the vulnerabilities in the smart lock system. We then provided the corresponding defense suggestions for the smart lock. We proposed to provide security mechanisms for the smart home appliances in general at multiple levels to ensure the security of smart devices, including the mobile app, mobile operating system, and smart home appliance hardware. In future work, we plan to investigate other types of IoT devices, and develop a holistic security framework to secure the IoT systems.

References

[1] D. Evans, The Internet of Things - How the Next Evolution of the Internet is Changing Everything, White Paper. Cisco Internet Business Solutions Group (IBSG), April 2011.

[2] J. Wurm, K. Hoang, O. Arias, A. R. Sadeghi and Y. Jin. Security Analysis on Consumer and Industrial IoT Devices. 21st Asia and South Pacific Design Automation Conference (ASP-DAC), January 2016, pp. 519-524.

[3] E. Fernandes, J. Jung and A. Prakash, Security Analysis of Emerging Smart Home Applications, IEEE Symposium on Security and Privacy (SP), San Jose, CA, May 2016, pp. 636-654.

[4] August Smart Lock, http://august.com/.

[5] E. Griffith and A. Colon, The Best Smart Home Devices of 2017, December, 2016, http://www.pcmag.com/article2/0,2817,2410889,00.asp.

[6] Samsung, Samsung SmartThings, https://www.smartthings.com/.

[7] Amazon, Amazon Echo with Alexa Enabled, https://www.amazon.com/Amazon-Echo-Bluetooth-Speaker-with-WiFi-Alexa/dp/B00X4WHP5E.

[8] Philips, Philips Hue, http://www2.meethue.com/en-us/.

[9] Nest, https://nest.com/.

[10] G. Hernandez, O. Arias, D. Buentello, and Y. Jin, Smart Nest Thermostat: a Smart Spy in Your Home, Black Hat USA, August 2014.

[11] A. Rose and B. Bramsey, Picking Bluetooth Low Energy Locks from a Quarter Mile Away, DEF CON 24 Hacking Conference, 2016.

[12] Kwikset, Kwikset Kevo Smart Lock, http://www.kwikset.com/kevo/default.

[13] QuickLock, https://www.thequicklock.com/.

[14] iBluLock Bluetooth Padlock, http://iblue.eu/.

[15] Jmaxxz Blog, https://jmaxxz.com/blog/.

[16] The August Smart Lock's not so 2-Factor Authentication (Part 1), February, 2015, https://jmaxxz.com/blog/?p=476.

[17] The August Smart Lock's not so smart password reset (Part 2), March, 2015, https://jmaxxz.com/blog/?p=498.

[18] G. Ho, D. Leung, P. Mishra, A. Hosseini, D. Song and D. Wagner, Smart Locks: Lessons for Securing Commodity Internet of Things Devices, ACM Asia Conference on Computer and Communications Security, May 2016, pp. 461-472.

[19] D. Walters (sretlawd), GitHub Repository: augustctl, https://github.com/sretlawd/augustctl.

[20] D. Balfanz, G. Durfee, N. Shankar, D. Smetters, J. Staddon and H. Wong, Secret Handshakes from Pairing-Based Key Agreements, IEEE Symposium on Security and Privacy, May 2003, pp. 180-196.

[21] Z. Wang, M. Rahul and S. Angelos, Implementing and Optimizing an Encryption Filesystem on Android, IEEE 13th International Conference, July 2012, pp. 52-62.

[22] S. Bugiel, S. Heuser, and A. Sadeghi, Flexible and fine-grained mandatory access control on Android for diverse security and privacy policies. In Proceedings of the 22nd USENIX conference on Security (SEC'13). USENIX Association, Berkeley, CA, USA, 131-146.