CITIoT: Data Leakage in Home Automation Systems

In recent years, smart home devices have become one of the most popular categories in the Internet of Things (IoT) accounting for $3.5 billion of a $292 billion industry; over 29 million smart home devices are expected to ship in 2017, a 63 percent increase over 2016 [1]. Wi-Fi and Bluetooth Low Energy (BLE) are two of the primary protocols used in these devices and are commonly implemented in security cameras, locks, medical devices, sensors, and a myriad of other devices. With the increasing prevalence of BLE and Wi-Fi devices in the home, consumers must be aware of the information these devices inadvertently broadcast and what kind of privacy data an outside observer can infer.

To stress this threat and the need for secure smart home development, we provide the following scenario: a user is in a smart home connected to a Wi-Fi access point (AP) and interacts with various Wi-Fi and BLE IoT devices. In the morning, the user turns on lights, activates sensors while walking throughout the house, and eventually turns off the lights before leaving the house for work (the door is locked before leaving). While at work, the user checks on the temperature in the house or other devices (e.g., security cameras) remotely. After work, the user returns home, unlocks the front door, turns on lights, and activates sensors throughout the house. Before going to bed, the lights are turned off. During the day, an observer is outside the house sniffing wireless data packets attempting to infer information about the user, devices, and events within the house. The observation process is completely passive and, therefore, undetectable by the user. The eavesdropper has no access to network credentials and is not part of the smart home network.

This work contributes to the field of IoT security, specifically privacy within a smart home, by illustrating how devices leak data and demonstrating how users can prevent leakage. In doing so, we make four principal contributions:

Smart home architecture. To analyze IoT data leakage in the wild, we provide a realistic smart home architecture that integrates Wi-Fi and BLE commercial-off-the-shelf (COTS) devices with Apple's home automation application, HomeKit. Examples of interactions with the smart home environment include turning on lights, opening doors, activating motion sensors, and unlocking locks. These interactions occur while the user is home or away.

Vulnerability analysis. We explain how an eavesdropper can use device vulnerabilities to extract information while outside the smart home environment via raw signals sniffed over the air. These, combined with characteristic data exchanges and packet sizes, can be used to fingerprint components of the smart home environment.

Classification, Identification, Tracking of IoT (CITIoT). We present a tool that demonstrates four capabilities enabled by data leakage: network mapping, device classification, event identification, and user tracking. For Wi-Fi, a fingerprinting technique is applied to classify smart home devices into one of three groups: sensor, electrical outlet, or camera. For BLE devices, we similarly classify devices, but provide more descriptive information. The fingerprint technique is also applied to identify events within the smart home such as turning on a light or movement in the house. Lastly, the observed smart home traffic is used to predict when users will be in the smart home.

Synthesis. We stress the importance of smart home operational security by demonstrating how an observer can use the information gathered from smart home devices to create pattern-of-life models, crack a BLE lock, and gain access to the home when a user is predicted to be away. We observe that these vulnerabilities are not unique to the devices under study, but indicative of IoT design and can be extended to critical infrastructure (CI) technologies or any smart environment. We also provide methods to prevent data leakage and defend against the vulnerabilities presented.

The remainder of this paper introduces the reader to the smart home architecture created to provide real-world data to CITIoT, the methodology in creating the CITIoT tool, and an analysis of results.

Smart Home Architecture

As shown in Figure 1, the smart home architecture includes three controller components and various connected devices. The controller components include (i) a Raspberry Pi running the Homebridge server that emulates the iOS HomeKit API and exposes supported devices to Apple's HomeKit, (ii) an iPhone 6+ running Apple's HomeKit and device specific applications, and (iii) an Apple TV Generation 2 acting as a smart home hub to allow access to HomeKit supported devices while the user is away from the smart home. The communication between controllers and devices can be observed in Figure 1 and is described in the rest of this section.

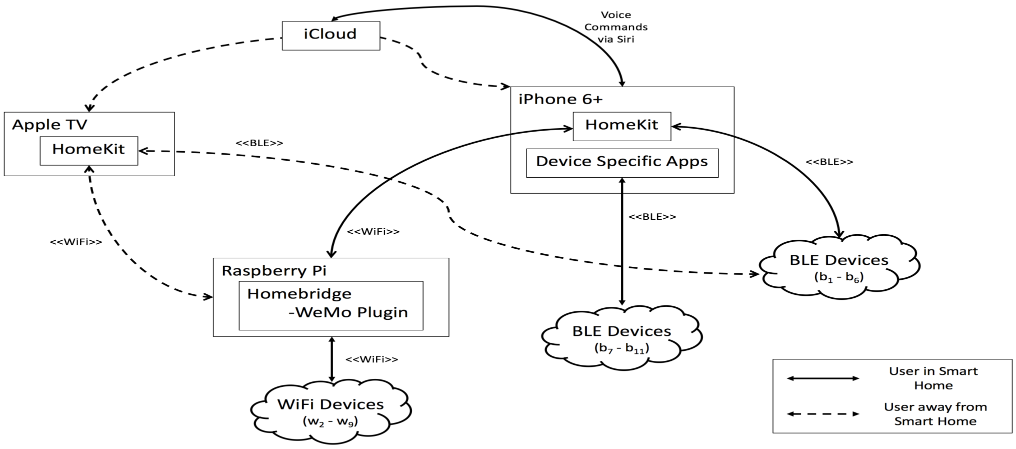

Figure 1. Smart Home Architecture

Raspberry Pi.

The Raspberry Pi 3 Model B with Raspbian Jessie Lite version 4.9 operating system is connected to the smart home network via the on-board 802.11 b/g/n 2.4 GHz wireless chip [4]. The Raspberry Pi runs Homebridge version 0.4.14 as a systemd service, and each interaction between a controller and device is logged in the systemd journal [5]. A Homebridge module is utilized to expose Belkin devices to Apple's Homekit and is loaded into Homebridge [3]. The Apple devices communicate with the Raspberry Pi to interact with the Wi-Fi devices (devices w_2-w_9 in Table 1).

Apple Devices.

The iPhone 6+ and Apple TV act as controllers in the smart home architecture and connect to devices via Wi-Fi and BLE. When the user is home, the iPhone connects to Wi-Fi devices via the Homebridge and directly to the BLE devices. Some of the BLE devices are not supported by Apple's Homekit and can only be accessed through the manufacturer-provided iOS application on the iPhone (devices b_7-b_{12} in Table 2). When the user is away from the smart home, the iPhone can communicate with Homekit-supported devices via the iCloud and Apple TV acting as a hub. For example, if the user is away from home and wants to access the temperature in a room, the iPhone communicates with the Apple TV via the iCloud and the Apple TV will communicate with the device in the home via Wi-Fi or BLE. This will only work with Homekit supported devices, therefore, BLE devices b_7-b_{12} cannot be accessed while the user is away from the home.

Wi-Fi Devices.

To facilitate Wi-Fi communication in the smart home architecture, a 2.4 GHz Wi-Fi AP, "Prancing Pony", was setup with Wi-Fi Protected Access 2 - Pre-Shared

Key (WPA2-PSK) security on channel 1. A list of devices connected to the AP can be found in Table 1. The smart home devices include a camera, six outlets (four smart plugs, one mini plug, and one energy plug), and a motion sensor (w_2-w_9). These devices use the Homebridge to communicate with Apple's HomeKit on the iPhone.

Bluetooth Low Energy (BLE) Devices.

For BLE communication to occur in the smart home architecture a BLE master must be present. In the smart home architecture, the iPhone and Apple TV act as masters while each of the BLE devices are slaves. A list of devices operating in the BLE can be found in Table 2. The Media Access Control (MAC) addresses for each device are not included because they are randomized per the BLE protocol [2].

Classification, Identification, Tracking of IoT (CITIoT) Tool

CITIoT operates in five-steps: (i) reconnaissance and scanning, (ii) passive sniffing, (iii) device classification, (iv) event identification, and (v) user tracking. During reconnaissance and scanning, the observer identifies the target, obtains the target's MAC address, identifies the target's home network, and scans the network creating a network map of the Wi-Fi devices. In passive sniffing, the obsever sniffs BLE and Wi-Fi wireless traffic throughout the day and parses the capture file for values of interest. Next, the captures are sent to CITIoT and the smart home devices are classifed as an outlet, sensor, or camera. After the devices are classified, the captures are parsed again and all device events are identified. Lastly, the tool utilizes the Wi-Fi MAC address to determine when a user is in the home or away.

Table 1. Wi-Fi Devices.

ID	Manuf	Device Type	Device Name	MAC	IP Address
w_1	Calix	Wireless Router	Prancing Pony	EC:4F:82:73:D1:1A	-
w_2	Belkin	Camera	NetCam	EC:1A:59:E4:FD:41	192.168.1.44
w_3	Belkin	Outlet	Switch1	B4:75:0E:0D:33:D5	192.168.1.40
w_4	Belkin	Outlet	Switch2	B4:75:0E:0D:94:65	192.168.1.41
w_5	Belkin	Outlet	Switch3	94:10:3E:2B:7A:55	192.168.1.42
w_6	Belkin	Outlet	Switch4	14:91:82:C8:6A:09	192.168.1.7
w_7	Belkin	Motion Sensor	Motion	EC:1A:59:F1:FB:21	192.168.1.43
w_8	Belkin	Outlet	Insight	14:91:82:24:DD:35	192.168.1.47
w_9	WeMo	Outlet	Mini	60:38:E0:EE:7C:E5	192.168.1.51
w_{10}	Raspberry Pi 3B	Computer	Pi	B8:27:EB:09:1A:81	12.168.1.50
w_{11}	Apple	iPhone 6+	Steves-phone	A0:18:28:33:34:F8	192.168.1.4
w_{12}	Apple	TV 2	Apple-TV	08:66:98:ED:1E:19	192.168.1.54

Table 2. BLE Devices.

ID	Manuf	Device Type	Device Name
b_1	Elgato	Indoor Temperature	Eve Room
b_2	Elgato	Outdoor Temperature	Eve Weather
b_3	Elgato	Motion Sensor	Eve Motion
b_4	Elgato	Outlet	Eve Energy
b_5	Elgato	Switch	Eve Light
b_6	Elgato	Door Sensor	Eve Door
b_7	Instant Pot	Smartcooker	Instant Pot
b_8	MPow	Lightbulb	Playbulb
b_9	ZKTeco	Lock	BioLock
b_{10}	BitLock	Lock	Bike lock
b_{11}	SafeTech	Gunsafe	Gunsafe
b_{12}	Apple	iPhone 6+	Steves-phone
b_{13}	Apple	TV 2	Apple TV

Results

CITIoT was evaluated while a user interacted with the smart home environment ten hours a day for five days. During experimentation, devices are accessed while the user is both home and away to observe differences in communication. The user interacted with devices while home in the morning and evening, and while away in the afternoon. During the interactions, the user logged the time, device, and type of each event. Each device was activated twice in the morning, twice in the evening, and a few times throughout the day for a total of more than twenty activations per device per day. An analysis of the accuracy of CITIoT is presented through the comparison of device classification with actual device types, event identification with the logs of actual event times, and user tracking with actual user locations. Preliminary results show that, on average, the device classification was 95% successful, event identification was 84% successful, and user tracking was 100% successful in determining if the user was in the home or not. On average, the tool failed to identify 16% of events and identified 5 events per day that did not actually occur.

Bibliography

1. Consumer Technology Association. "Record Year Ahead: Consumer Enthusiasm for Connectivity to Propel Tech Industry to Record-Setting Revenues," 2017. [Last accessed August 27, 2017], *https://www.cta.tech/News/Press-Releases/2016/January/Record-Year-Ahead-Consumer-Enthusiasm-for-Connect.aspx.*

2. Bluetooth SIG. *Specification of the Bluetooth System Core Version 4.2*, 2010. [Last accessed on August 30, 2017], *www.bluetooth.com/specifications/bluetooth-core-specification.*

3. devbobo. "homebridge-platform-wemo," 2017. [Last accessed on August 30, 2017], Available at *npmjs.com/package/homebridge-platform-wemo/.*

4. Raspberry Pi Foundation. "Raspberry Pi 3 Model B Speicfications," 2016. [Last accessed on August 30, 2017], Available at *raspberrypi.org/products/raspberry-pi-3-model-b/.*

5. nfarina. "Homebridge," 2015. [Last accessed on August 30, 2017], Available at *github.com/nfarina/homebridge/.*