Through the Spyglass: Towards IoT Companion App Man-in-the-Middle Attacks

TJ OConnor
Florida Institute of Technology
Melbourne, FL, USA
toconnor@fit.edu

Dylan Jessee
Florida Institute of Technology
Melbourne, FL, USA
djessee2020@my.fit.edu

Daniel Campos
Florida Institute of Technology
Melbourne, FL, USA
dcampos2015@my.fit.edu

ABSTRACT
The lack of mature development in smart home companion applications complicates Internet of Things (IoT) security and privacy. Companion applications offer transparency and control for smart home devices that otherwise lack displays or interfaces. We access our smart home devices through a distributed communication architecture that seamlessly integrates smart home devices, cloud-based servers, and our mobile devices. This paper seeks to better understand IoT security and privacy by studying the design flaws of this distributed communications channel for smart home devices. To understand this, we then assess the vulnerability of 20 popular smart home vendors to this attack. Our analysis discovers pervasive failures in the distributed communications channels across 16 different vendors. A successful attack allows adversaries to conceal device users, manipulate the state of locks, spoof camera images, and manipulate history log files. While our work uncovers pervasive failures, vendors can take measures to improve confidentiality and integrity in smart home devices and their applications.

CCS CONCEPTS
• Security and privacy → Security protocols; Mobile and wireless security; • Computer systems organization → Sensors and actuators.

ACM Reference Format:

1 INTRODUCTION
Always-on and always-responsive smart home devices offer security and convenience to our digitally connected homes. Connected locks, motion sensors, and security cameras can provide us ease of mind. Digital speakers can play our favorite music, check the weather, or set an alarm for the following morning. These devices have become so commonplace that they are increasingly used in criminal cases as forensic evidence [3, 12, 14, 17, 26]. In these cases, the courts have used the history of digital voice assistants and fitness trackers to confirm and deny alibis.

The rapid adoption of these devices into the market and the willingness for the courts to use them as forensic evidence presents a concern. We hypothesize that the distributed communications architecture of IoT introduces vulnerabilities that allow an attacker to intercept and manipulate the communications channel, affecting the user-level perception of an IoT device. We apply this problem against a broad array of smart home device vendors to conceal malicious users, suppress motion reporting, modify camera images, unlock doors, and manipulate history log files. Our work identifies systemic design failures that introduce threats to the confidentiality, integrity, or availability of IoT sensors and actuators in smart home IoT devices. This paper makes the following contributions:

1. We propose and implement an attack methodology that manipulates IoT sensors and actuators by modifying IoT devices’ distributed communication channels. Our attack conceals users, manipulates reporting, and modifies the state of IoT devices while intercepting privacy-sensitive information.
2. We evaluate the susceptibility of our attack for 20 popular smart-home vendors. We identify that 16 of the 20 vendors fail to implement security measures, enabling pervasive attacks. For reproducibility purposes, we include our experiment code at https://research.fit.edu/iot. Further, we offer countermeasures to prevent our attack vector.

Findings: In this paper, we examine the critical design and implementation flaws on companion applications that inform broad findings. First, smart home companion applications implement naive and insecure protocols that rely on binary-to-text instead of cryptographic schemes to protect message confidentiality. Next, companion applications lack mechanisms to preserve the integrity of messages, leading to spoofing the state, users, or history of IoT devices. Finally, the distributed architecture and reliance on content distribution networks (CDNs) contribute to design flaws as vendors fail to validate content from CDNs properly.

2 BACKGROUND & MOTIVATION
2.1 Overview of IoT Companion App Protocols
Resource-constrained smart home devices commonly rely on managed cloud environments for storage and processing. Through these managed cloud platforms, users interact with smart home devices through companion applications that leverage a meet-in-the-middle approach. However, Alrawi et al. [1] performed a large-scale evaluation and identified that over 40% of IoT companion applications did not properly enforce encryption and allowed for communication
Attacker Goals:
We consider an attacker whose goal is to modify a device’s transparency and functionality surreptitiously. To illustrate, we consider an attacker who would like to create and conceal back-door accounts on a device, manipulate the state of a device such as a connected lock, or spoof images from a security camera.

Attacker Capabilities and Assumptions:
We consider a technically sophisticated attacker that has the presence and privileges to perform a man-in-the-middle (MiTM) attack on a user’s mobile device. As such, the attacker must have the access and privileges to install a malicious certificate on the victim’s device to proxy encrypted traffic. The attacker may be a domestic partner who has access to the victim’s phone. However, it also may be an attacker that wishes to secretly spy on employees of a particular company by accessing their smart home devices. In this case, the attacker can compromise the company mobile device management (MDM) servers to install a certificate on an employee’s mobile device. While companies can deploy a proxy via MDM to support a company’s deep packet intrusion detection system, an attacker may leverage the same functionality to maliciously proxy, intercept, and modify traffic [2]. In a similar approach, a targeted attack leveraged MDM for side-loading malicious applications onto mobile devices [20].

Figure 1: Attackers can also leverage man-in-the-middle attacks to spoof response messages carrying user lists or device histories.

We consider attacker goals similar to those of mobile spyware engineered to surveil, intimidate and harass victims.

2.2 Motivation
Unfortunately, the always-on and always-connected nature of smart home devices makes them an attractive platform to facilitate intimate partner violence [7, 10, 15, 18, 23]. IoT devices offer the promise of security with connected locks, alarms, and security cameras. However, attackers can leverage the immature but pervasive nature of IoT to intimidate, threaten, monitor, and harass victims [10]. With the rapid proliferation of smart home devices and their breadth of sensors, IoT has the regrettable potential to transform technology-enabled abuse. Cameras and microphones can abusively surveil our most sensitive moments. Video-connected doorbells and smart locks can reveal occupancy information about our homes. Lights, temperature controls, and smart appliances can be used to gaslight, intimidate, and control victims [18]. The lack of transparency in IoT devices further complicates this problem. The limited device interfaces and naive companion applications often do not present a user with an understanding of device access control. Our motivation is supported by a recent ADT employee who pled guilty to accessing the security cameras of 220 women over 9,600 times during a four-year period [7]. We believe it necessary to examine the pervasive failures in smart home companion applications to prevent against our hypothesized vector.

2.3 Threat Model
Attacker Goals: We consider an attacker whose goal is to modify a device’s transparency and functionality surreptitiously. To illustrate, we consider an attacker who would like to create and conceal back-door accounts on a device, manipulate the state of a device such as a connected lock, or spoof images from a security camera.
present a deceptive state of the system. Section 4 examines some of the other messages that we can intercept or manipulate.

4 EVALUATION

4.1 Experiment Setup

We implemented a smart-home lab environment with devices from 20 different vendors to explore the severity and pervasiveness of attacks against IoT devices. We purchased all the low-cost devices in either 2019 or 2020 from well-known US retailers, including Walmart, Lowe’s, Target, Best Buy, and Amazon. Further, we installed the vendor companion applications on an 8th generation iPad and iPhone XR, running version 14.4.2 of iOS (current at the time of the experiments). As described in our threat model, we installed a self-signed mitmproxy [5] certificate to intercept and modify HTTPS headers and content. The companion application versions are listed in Table 1. To indicate the popularity of the vendor, Table 1 also lists the number of application downloads. We used the Android app version to benchmark vendor popularity since the Apple Store does not release app download metrics.

4.2 Attacks Tested

We developed 16 mitmproxy scripts to perform functionality that deceives the state of IoT devices (e.g., hiding users, manipulating logs, intercepting sensitive information, manipulating user files, and controlling user behaviors.) For reproducibility purposes, we include all the code for the following experiments at https://research.fit.edu/iot

- August Lock: hide/manipulate shared users
- UltraLock: hide/manipulate shared users
- Sifely Lock: hide/manipulate admin users
- Simplisafe Alarm: manipulate/clear alarm log files
- Smartthings: manipulate/clear log files
- Lockly: manipulate/clear log log files
- Amazon Echo: intercept messages responses
- Blink Camera: intercept cloud account credentials
- NightOwl Doorbell: intercept local account credentials
- Hue Lights: leak internal IP address of hub
- Google Home Camera: spoof camera images
- Nest Camera: spoof camera images
- Wyze Camera: spoof camera images
- Momentum Camera: spoof camera images
- Roku TV: spoof roku tv show images
- Schlage Lock: force lock to unlock

5 RESULTS

Table 1 summarizes the results of our experiment. Our results demonstrate that a majority of smart home vendors (16 out of 20) fail to enforce any mitigation measures to prevent man-in-the-middle attacks, enabling our unique attack approach. Further, all 16 out of 20 vendors implement naive communication protocols relying solely on TLS for protecting the confidentiality and integrity of the data. However, the reliance on TLS proves insufficient, as the applications fail to prevent MiTM attacks by performing proper certificate validation or certificate pinning. These results confirm our hypothesized attack vector that IoT devices’ unprotected and naive distributed communication channel enables pervasive attacks that can present a deceptive state of devices. Despite these pervasive findings, we argue that vendors can realize secure solutions and identify that the Arlo, Geeni, TP-Link Kasa, and Ring vendors properly validate certificates and enforce certificate pinning.

5.1 Evaluation Findings

Finding 1: IoT Apps Rely on Naive and Insecure Protocols

Our experiment observed that 16 vendors used naive inter-exchange protocols and binary-to-text encodings to transmit sensitive IoT data. In most cases, HTTPS requests and responses consisted of JSON exchange format messages or HTTPS parameters. As opposed to using proprietary protocols or end-to-end encryption, vendors encoded sensitive data using base64 encoding, allowing the sensitive information transmitted and received by the companion applications to be manipulated. These insecure approaches facilitated easily modifying the sensitive information transmitted and received by the companion applications.

Finding 2: IoT Apps Lack Message Integrity: We observe that we can modify the messages of devices in transmit without presenting an error message. These findings reinforce previous works [9, 23] that identify that IoT devices lack message integrity and data authentication. The lack of message integrity and data authentication presents a troublesome concern. Smart home devices are being increasingly used to confirm or deny alibis in legal cases [3, 12, 14, 17, 26]. Spoofed messages could be used to fabricate alibis.

Finding 3: IoT Apps Rely on Unsecured CDNs: IoT Apps rely on content distribution networks (CDNs) to provide high availability and service spatially to users. By leveraging providers like Google Cloud, Tuya Smart, or Amazon AWS, IoT vendors attempt to reduce the latency to streaming IoT sensor content. However, they often rapidly deploy these platforms without concern for security [4, 25]. One explanation for this may be the reliance on turnkey solutions for IoT devices. Turnkey providers offer complete solutions that provide the required infrastructure and hardware components for an IoT ecosystem. For example, the Scienier turnkey platform provides the app SDK, Cloud API, and libraries to develop an IoT ecosystem for a smart lock rapidly. Sifely, relying on the Scienier, inherits the vulnerabilities baked into the turnkey solution.

6 ATTACK COUNTERMEASURES

Certificate Pinning: Applications can verify if the proper certificate authority (CA) signed the certificate and inform the user of a spoofed certificate. However, 16 smart home vendors in Table 1 accepted our spoofed certificate without proper validation. Several solutions exist to validate and pin certificates properly. We observe that both the Ring and Arlo vendors prevent this attack by leveraging the Trustkit application programming interface (API). The TrustKit API [6] implements RFC 7469: HTTP Public Key Pinning Specification [8]. This approach instructs HTTPS user agents to pin the cryptographic identities, mitigating the likelihood of man-in-the-middle attacks. Other popular frameworks for certificate pinning on the iOS and Android platforms include AlamoFire [24] and AFNetworking [19]. In contrast to the lightweight TrustKit solution, AlamoFire and AFNetworking offer a complete network library. After configuration, either library will enable SSL pinning for
Table 1: Our results demonstrate pervasive failures in companion applications that enable our proposed attack methodology.

<table>
<thead>
<tr>
<th>Vendor</th>
<th>App Version</th>
<th>App Downloads</th>
<th>Vulnerable To Attack</th>
<th>Transparent Attack</th>
<th>Vulnerable Domains</th>
</tr>
</thead>
<tbody>
<tr>
<td>August</td>
<td>v11.01</td>
<td>500,000+</td>
<td>●</td>
<td>○</td>
<td>api-production.august.com, logger.august.com</td>
</tr>
<tr>
<td>Amazon Alexa</td>
<td>v1.24.307576.0</td>
<td>50,000,000+</td>
<td>●</td>
<td>○</td>
<td>alexa.amazon.com, kinesis.us-east-1.amazonaws.com, avs-alexa-12-na.amazon.com</td>
</tr>
<tr>
<td>Arlo</td>
<td>v3.2 (2202)</td>
<td>1,000,000+</td>
<td>○</td>
<td>○</td>
<td>(rest-prod</td>
</tr>
<tr>
<td>Blink</td>
<td>v6.2.9</td>
<td>1,000,000+</td>
<td>●</td>
<td>○</td>
<td>(webapi.camera.home)</td>
</tr>
<tr>
<td>Geeni</td>
<td>v2.1.1</td>
<td>1,000,000+</td>
<td>●</td>
<td>○</td>
<td>clients3.google.com, nexusapi-g1.camera.home.nest.com</td>
</tr>
<tr>
<td>Google Home</td>
<td>v2.36.113</td>
<td>100,000,000+</td>
<td>●</td>
<td>○</td>
<td>notifications-pa.googleapis.com, play.googleapis.com</td>
</tr>
<tr>
<td>Hue</td>
<td>v3.48.0</td>
<td>5,000,000+</td>
<td>●</td>
<td>○</td>
<td>discovery.meethue.com, am12.amplidude.com</td>
</tr>
<tr>
<td>TP-Link Kasa</td>
<td>v2.30.0</td>
<td>1,000,000+</td>
<td>○</td>
<td>○</td>
<td>api.yonomi.com, in.appcenter.ms</td>
</tr>
<tr>
<td>Lockly</td>
<td>v1.9.8</td>
<td>10,000+</td>
<td>●</td>
<td>○</td>
<td>api.semify.com, servlet.science.cn</td>
</tr>
<tr>
<td>Nest</td>
<td>v5.60.0</td>
<td>5,000,000+</td>
<td>●</td>
<td>○</td>
<td>(api</td>
</tr>
<tr>
<td>Momentum</td>
<td>v2.0.2</td>
<td>500,000+</td>
<td></td>
<td>○</td>
<td>wzztk.com, apipentive.com</td>
</tr>
<tr>
<td>Night Owl</td>
<td>v5.0.95</td>
<td>100,000+</td>
<td></td>
<td>○</td>
<td>api-rest.nightowlconnect.com, host.nightowldev04.com</td>
</tr>
<tr>
<td>Ring</td>
<td>v5.38.1</td>
<td>10,000,000+</td>
<td>○</td>
<td>○</td>
<td>(prod.mobile</td>
</tr>
<tr>
<td>Roku</td>
<td>v7.71.2</td>
<td>10,000,000+</td>
<td>●</td>
<td>○</td>
<td>api.allegion.yonomi.cloud, in.appcenter.ms</td>
</tr>
<tr>
<td>Schlage</td>
<td>v4.2.0</td>
<td>100,000+</td>
<td>●</td>
<td>○</td>
<td>servlet.science.cn</td>
</tr>
<tr>
<td>Sifely</td>
<td>v1.2.1</td>
<td>5,000+</td>
<td>●</td>
<td>○</td>
<td>api.simplisafe.com</td>
</tr>
<tr>
<td>SimpliSafe</td>
<td>v2074.67.00</td>
<td>500,000+</td>
<td>●</td>
<td>○</td>
<td>api.simplisafe.com</td>
</tr>
<tr>
<td>SmartThings</td>
<td>v1.6.65-502</td>
<td>500,000,000+</td>
<td>●</td>
<td>○</td>
<td>api.smartthings.com, us-auth2.samsungosp.com, accountant.samsungiotcloud.com</td>
</tr>
<tr>
<td>UltraLoq</td>
<td>v1.10.1</td>
<td>50,000+</td>
<td>●</td>
<td>○</td>
<td>dlx1.dl.xatlas.samsung.com</td>
</tr>
<tr>
<td>Wyze</td>
<td>v2.19.24</td>
<td>1,000,000+</td>
<td>●</td>
<td>○</td>
<td>(logtail</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>○</td>
<td>wyze-device-alarm-file.s3.us-west-2.amazonaws.com</td>
</tr>
</tbody>
</table>

- ●: Attack is successful; attack is transparent
- ○: Attack fails to succeed; attack prompts user

future communication. While attacks exist to overcome these mitigations [11], they generally require debugging or app modification that is not consistent with our threat model.

End-to-End Encryption and Digital Signatures: End-to-end encryption (E2EE) with device-specific keys presents an opportunity to preserve confidentiality and ensure the integrity of smart home message traffic. Additionally, digital signatures could be used to protect the integrity of messages. These approaches would prevent the eavesdropping and manipulation attacks presented in our work. However, these approaches require pre-provisioning keys for IoT devices, companion applications, and cloud-based servers that communicate in a distributed architecture. Approaches that leverage key-exchange algorithms to create cryptographic keys would still be vulnerable to man-in-the-middle attacks, eliminating any benefit gained by E2EE [9]. These approaches offer an interesting problem that we reserve for future work. Previous works have proposed leveraging unique device identifiers from the smart home devices (e.g., the device serial number) to seed key generation. However, hard-coding keys to serial numbers could lead to key prediction and guessing. While E2EE offers promise, future works must examine the design and implementation of key distribution algorithms.

7 RELATED WORK

Previous works have examined the feasibility of surveilling or manipulating IoT device traffic. Hariri [13] and O’Connor et al. [23] explored blinding IoT sensors and confusing their state through network-based selective forwarding attacks. Janes et al. [15] examined systemic flaws in cloud-based access control platforms that enable attackers to persist on cloud-based smart home cameras after account revocation. Previous works have also leveraged MiTM attacks against IoT devices [9, 21, 22]. Moughaddam et al. [22] constructed a tool to perform best-effort TLS interception and examined the sensitive information leaked by Smart TVs. Jeske explored sensitive information leaks from the Waze and Google Maps applications through man-in-the-middle attacks [16]. Mitev et al. [21] proposed and implemented a series of MiTM attacks against the Alexa Skills ecosystem by manipulating audio transmissions to digital assistants. Similar to our work, Fereidooni et al. [9] explored vulnerabilities in Fitness Trackers that enabled MiTM attacks to leak sensitive information and inject fake data.

8 CONCLUSION

In this work, we hypothesized that smart home devices’ naive architecture and communication protocols enable network-mode rootkits to conceal device users, manipulate the state of locks, spoof camera images, and manipulate history logs. This paper explored the pervasive design flaws in the companion applications of smart home devices that facilitate these attacks. We have shown that the majority of vendors naively implement companion applications without concern for certificate validation or certificate-pinning. To demonstrate the broad scope of the problem, we evaluated the vulnerability of 20 popular smart home vendors. We uncover that 16 out of 20 vendors suffer from critical design flaws that fail to: (1) properly validate certificates (2) protect the integrity of message traffic. Further, we examined the impact of such attacks and presented countermeasures to prevent future attacks.
ACKNOWLEDGMENTS

This material is based upon work supported in whole or in part with funding from the Office of Naval Research (ONR) contract #N00014-20-1-2798. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ONR and/or any agency or entity of the United States Government.

REFERENCES


