

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:

TJ OConnor, Dylan Jessee, and Daniel Campos. 2021. Through the Spyglass: Towards IoT Companion App Man-in-the-Middle Attacks. In *Cyber Security Experimentation and Test Workshop (CSET '21), August 9, 2021, Virtual, CA, USA*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3474718.3474729>

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,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CSET '21, August 9, 2021, Virtual, CA, USA

© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-9065-1/21/08...\$15.00
<https://doi.org/10.1145/3474718.3474729>

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

over unverified connections. This critical flaw leaves the link between the companion application and cloud-based servers open for man-in-the-middle (MiTM) attacks. We leverage this design flaw to implement attacks against the user perception of IoT devices.

IoT devices often leverage lightweight publish/subscriber protocols such as MQTT(-S) or XMPP. In contrast, we observe that companion applications predominately rely on HTTPS, enabling support and scalability for cloud-based platforms. Companion applications use lightweight data-interexchange formats such as JavaScript Object Notation (JSON) and binary-to-text encodings (e.g., base64) to share and encode data. As the always-on and always-connected nature of IoT devices produces continuous traffic, lightweight and standard protocols can reduce bandwidth requirements. However, using these lightweight and naive protocols reduces the required attack complexity since an attacker does not need to calculate digital signatures, correct error correction codes, or perform cryptographic attacks. To understand this naivety, consider the Schlage Wireless Lock. In our experiments, we identified that the lock indicates its state (i.e., locked or unlocked) by a single integer set to 0 (locked) or 1 (unlocked) in a JSON message. Flipping this integer, as described in Section 3, is all that is required to change the state of the lock. The attacker does not need to perform any additional steps such as sequence number prediction or computing a digital signature. She can change the state of the lock by intercepting and manipulating one integer in HTTPS traffic.

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.

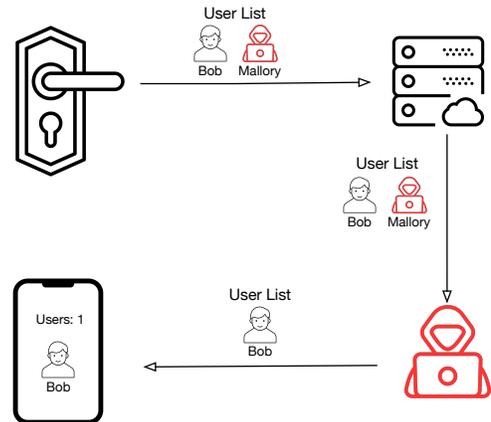


Figure 1: Attackers can also leverage man-in-the-middle attacks to spoof response messages carrying user lists or device histories.

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

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 with physical access to a device that intends to use a rootkit to threaten, intimidate, or monitor their partner [10]. We consider an attacker who has a similar presence and privilege to install mobile phone spyware. This might 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].

3 ATTACK OVERVIEW

Figure 1 depicts our straightforward attack methodology. In this example, the attacker intercepts and manipulates message traffic containing the list of users who have access to the lock. We accomplish this by manipulating the HTTPS response when the companion application polls the cloud servers for the user list. It is important to note that manipulation can occur in traffic either to or from cloud servers. Companion applications for IoT offer a new paradigm for attack vectors since they provide the only transparency and control of a device. By modifying messages in transit, we can

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
- **UltraLoq Lock:** 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 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 transit 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 Sciener turnkey platform provides the app SDK, Cloud API, and libraries to develop an IoT ecosystem for a smart lock rapidly. Sifely, relying on the Sciener, 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.

Vendor	App Version	App Downloads	Vulnerable To Attack	Transparent Attack	Vulnerable Domains
August	v11.01	500,000+	●	○	api-production.august.com, logger.august.com
Amazon Alexa	v1.24.307576.0	50,000,000+	●	●	alexa.amazon.com, kinesis.us-east-1.amazonaws.com, avs-alexa-12-na.amazon.com
Arlo	v3.2 (2202)	1,000,000+	○	○	
Blink	v6.2.9	1,000,000+	●	●	(rest-prod apphelp rest-u011).immedia-semi.com
Geeni	v2.1.1	1,000,000+	○	○	
Google Home	v2.36.113	100,000,000+	●	●	clients3.google.com, nexusapi-gl1.camera.home.nest.com notifications-pa.googleapis.com, play.googleapis.com discovery.meethue.com, api2.amplitude.com
Hue	v3.48.0	5,000,000+	●	○	
TP-Link Kasa	v2.30.0	1,000,000+	○	○	
Lockly	v1.9.8	10,000+	●	●	apiserv03c.pin-genie.com
Nest	v5.60.0	5,000,000+	●	●	(webapi.camera.home logsink.home home).nest.com
Momentum	v2.0.2	500,000+	●	●	(api us-west-2) .pepperos.io, pepper-prod-recordings.s3.us-east-2.amazonaws.com wzrkt.com, api.apptentive.com
Night Owl	v5.0.95	100,000+	●	●	api-rest.nightowlconnect.com, host.nightowldvr04.com
Ring	v5.38.1	10,000,000+	○	○	
Roku	v7.71.2	10,000,000+	●	●	(prod.mobile images.sr.roku ls.cti).roku.com
Schlage	v4.2.0	100,000+	●	●	api.allegion.yonomi.cloud, in.appcenter.ms
Sifely	v1.2.1	5,000+	●	●	servlet.sciener.cn
SimpliSafe	v2074.67.0	500,000+	●	●	api.simplisafe.com
SmartThings	v1.6.65-502	500,000,000+	●	●	api.smartthings.com, us-auth2.samsungsp.com, accountant.samsungiotcloud.com dls.di.atlas.samsung.com
UltraLoq	v1.10.1	50,000+	●	●	(logtail app www).u-tec.com, s3.us-east-2.amazonaws.com
Wyze	v2.19.24	1,000,000+	●	●	(api wyze-platform-service wyze-membership-service).wyzecam.com wyze-device-alarm-file.s3.us-west-2.amazonaws.com

●: 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 OConnor 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]. Moghaddam 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

- [1] Omar Alrawi, Chaz Lever, Manos Antonakakis, and Fabian Monrose. 2019. Sok: Security evaluation of home-based iot deployments. In *Symposium on Security and Privacy*. IEEE, San Francisco, CA, 1362–1380.
- [2] Carlos Esteban Benitez. 2019. Wireless portable personal cyber-protection device. US Patent 10,305,930.
- [3] Nicole Chavez. 2017. Murder charge dropped in Amazon Echo case. <https://edition.cnn.com/2017/11/30/us/amazon-echo-arkansas-murder-case-dismissed/index.html>
- [4] Pietro Colombo and Elena Ferrari. 2018. Access control enforcement within mqtt-based internet of things ecosystems. In *Symposium on Access Control Models and Technologies*. ACM, Indianapolis, IN, 223–234.
- [5] Aldo Cortesi, Maximilian Hils, Thomas Kriechbaumer, and contributors. 2010. mitmproxy: A free and open source interactive HTTPS proxy. <https://mitmproxy.org/> [Version 6.0].
- [6] Alban Diquet, Angela Chow, Eric Castro, Daryl Low, Christopher Harrell, and Plasma Chen. 2020. TrustKit. <https://github.com/datatheorem/TrustKit>
- [7] Erin Dooley. 2021. ADT Technician Pleads Guilty to Hacking Home Security Footage. <https://www.justice.gov/usao-ndtx/pr/adt-technician-pleads-guilty-hacking-home-security-footage>
- [8] C. Evans, C. Palmer, and R. Sleevi. 2015. RFC 7469: Public Key Pinning Extension for HTTP. <https://tools.ietf.org/html/rfc7469>
- [9] Hossein Fereidooni, Tommaso Frassetto, Markus Miettinen, Ahmad-Reza Sadeghi, and Mauro Conti. 2017. Fitness trackers: fit for health but unfit for security and privacy. In *International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. IEEE, Philadelphia, PA, 19–24.
- [10] Diana Freed, Jackeline Palmer, Diana Minchala, Karen Levy, Thomas Ristenpart, and Nicola Dell. 2018. A Stalker's Paradise: How Intimate Partner Abusers Exploit Technology. In *Conference on Human Factors in Computing Systems (CHI)*. ACM, Montreal, Canada, 1–13.
- [11] David Greenwood, Justin Sounthiraraj, Sahs Garret, Zhiqiang Khan, and Latifur Lin. 2014. SMV-HUNTER: Large Scale, Automated Detection of SSL/TLS Man-in-the-Middle Vulnerabilities in Android Apps. In *Network and Distributed System Security Symposium (NDSS)*. Internet Society, San Diego, CA, 1–14.
- [12] Guardian Staff. 2019. Alexa, did he do it? Smart device could be witness in suspicious Florida death. <https://www.theguardian.com/us-news/2019/nov/01/alexa-florida-death-witness-amazon-echo>
- [13] Ali Hariri, Nicolas Giannelos, and Budi Arief. 2019. Selective Forwarding Attack on IoT Home Security Kits. In *European Symposium on Research in Computer Security*. Springer, Luxembourg, September, 360–373.
- [14] Christine Hauser. 2018. Police Use Fitbit Data to Charge 90-Year-Old Man in Stepdaughter's Killing. <https://www.nytimes.com/2018/10/03/us/fitbit-murder-arrest.html>
- [15] Blake Janes, Heather Crawford, and TJ OConnor. 2020. Never Ending Story: Authentication and Access Control Design Flaws in Shared IoT Devices. In *Security and Privacy SmartThings Workshop*. IEEE, San Francisco, CA, 104–109.
- [16] Tobias Jeske. 2013. Floating car data from smartphones: What google and waze know about you and how hackers can control traffic. In *Blackhat Europe*. Blackhat, Amsterdam, Netherlands, 1–12.
- [17] Jamiles Lartey. 2017. Man suspected in wife's murder after her Fitbit data doesn't match his alibi. <https://www.theguardian.com/technology/2017/apr/25/fitbit-data-murder-suspect-richard-dabate>
- [18] Roxanne Leitão. 2019. Anticipating smart home security and privacy threats with survivors of intimate partner abuse. In *Designing Interactive Systems Conference*. ACM, San Diego, CA, 527–539.
- [19] Matt. 2020. AFNetworking. <https://github.com/AFNetworking/AFNetworking>
- [20] Warren Mercer, Paul Rascagneres, and Andrew Williams. 2018. Advanced Mobile Malware Campaign in India uses Malicious MDM. <https://blog.talosintelligence.com/2018/07/Mobile-Malware-Campaign-uses-Malicious-MDM.html>
- [21] Richard Mitev, Markus Miettinen, and Ahmad-Reza Sadeghi. 2019. Alexa lied to me: Skill-based man-in-the-middle attacks on virtual assistants. In *Asia Conference on Computer and Communications Security*. ACM, Auckland, New Zealand, 465–478.
- [22] Hooman Mohajeri Moghaddam, Gunes Acar, Ben Burgess, Arunesh Mathur, Danny Yuxing Huang, Nick Feamster, Edward W Felten, Prateek Mittal, and Arvind Narayanan. 2019. Watching you watch: The tracking ecosystem of over-the-top tv streaming devices. In *SIGSAC Conference on Computer and Communications Security*. ACM, London, UK, 131–147.
- [23] TJ OConnor, William Enck, and Bradley Reaves. 2019. Blinded and Confused: Uncovering Systemic Flaws in Device Telemetry for Smart-Home Internet of Things. In *Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*. ACM, Miami, FL, 140–150.
- [24] Jon Shier. 2021. Alamofire. <https://github.com/Alamofire/Alamofire>
- [25] Qi Wang, Pubali Datta, Wei Yang, Si Liu, Adam Bates, and Carl A Gunter. 2019. Charting the attack surface of trigger-action IoT platforms. In *SIGSAC Conference on Computer and Communications Security*. ACM, London, UK, 1439–1453.
- [26] Zack Whittaker. 2018. Judge orders Amazon to turn over Echo recordings in double murder case. <https://techcrunch.com/2018/11/14/amazon-echo-recordings-judge-murder-case/>