Never Ending Story: Authentication and Access Control Design Flaws in Shared IoT Devices

Blake Janes*, Heather Crawford†, TJ OConnor‡
Florida Institute of Technology
bjanes2013@my.fit.edu*, {hcrawford†, toconnor‡}@fit.edu

***Abstract*—Internet-of-Things (IoT) devices implement weak authentication and access control schemes. The on-demand nature of IoT devices requires a responsive communications channel, which is often at odds with thorough authentication and access control. This paper seeks to better understand IoT device security by examining the design of authentication and access control schemes. In this work, we explore the challenge of propagating credential revocation and access control list modifications in a shared IoT ecosystem. We evaluate the vulnerability of 19 popular security cameras and doorbells against a straightforward user-interface bound adversary attack. Our results demonstrate that 16 of 19 surveyed devices suffer from flaws that enable unauthorized access after credential modification or revocation. We conclude by discussing these findings and propose a means for balancing authentication and access control schemes while still offering responsive communications channels.**

I. Introduction

Internet-of-Things (IoT) devices offer the promise of security and transparency for our connected homes. For example, wireless doorbells can identify and deter package thieves [1]. As these devices have gained more popularity, they have begun offering multiple user accounts per home. The benefit of multiple user accounts permits cohabitants to access shared devices in their homes. Multiple accounts allow a mother to check in on her children while away on a business trip. Simultaneously, a father may remotely close the connected garage door while attending an evening lecture. Unfortunately, the features that allow this same convenience can maliciously and surreptitiously monitor the auditory, visual, and location data between shared users. Technology-facilitated abuse is increasingly being used to stalk intimate partners electronically [2]–[4]. Electronic stalking can persist after revoking physical access to a shared residence. Poor authentication and access control schemes significantly complicate this problem by allowing an attacker access to previously shared IoT devices.

To illustrate this problem, researchers found that Ring doorbells failed to enforce a proper authentication scheme for multiple users that shared a single account [5]. They discovered that when a device owner changed the password, Ring did not immediately force other users to re-authenticate. Instead, users with a current session could remain connected indefinitely without having to enter the new credentials. In response to the vulnerability disclosure, Ring made changes to their authentication model. However, Ring admitted that credential modification still takes up to several hours to propagate [5]. In the balance of usability and security, Ring argued that immediately propagating credential modification would adversely penalize user experience with a burdensome performance impact. Ring recognized this approach was flawed and pushed a companion app update in January 2020 that revoked access with a password change. Attacker persistence after a password change is further complicated when attackers compromise credential databases. Such IoT credential database attacks are on the rise and have affected widely popular brands, including Ring, Wyze, and Nest [6]–[8].

This paper hypothesizes that the Ring failure is not an isolated incident but rather indicative of a systemic design failure in how users authenticate in shared IoT ecosystems. Specifically, we identify how credential revocation and modification often fails to propagate to connected users. Further, we examine the lack of transparency and control that complicates this problem. Authentication and access control should offer device owners the ability to control and limit device capabilities to subjects. Some devices offer a naïve model where successful authentication grants full access to the device. More mature implementations use access control lists that define the device capabilities and the subjects that may control them. Authentication and access control list changes should propagate immediately and be verifiable in order to preserve user privacy and security. However, our work identifies systemic design failures that prevent changes from propagating immediately and lack the transparency needed for verification.

This paper makes the following contributions:

1) We propose an attack methodology to persist after access revocation. Our attack allows a no longer permissioned user to view the video streams of connected cameras after a device owner has revoked or modified access.
2) We evaluate the susceptibility of our attack for 19 common security cameras and doorbells. We identify that 16 of the 19 devices are vulnerable to our attack. Further, we identify that all devices can improve transparency controls to identify privacy violations.

Findings: We uncover systemic design flaws that inform broad findings. First, IoT API servers distribute access control lists between APIs and low-latency content servers, effectively creating different, incomparable versions of the same list. Second, IoT devices fail to consider UI-bound adversaries in their threat models. Third, IoT devices fail to provide transparency of user access and actions. Finally, we argue that

DOI 10.1109/SPW50608.2020.00033

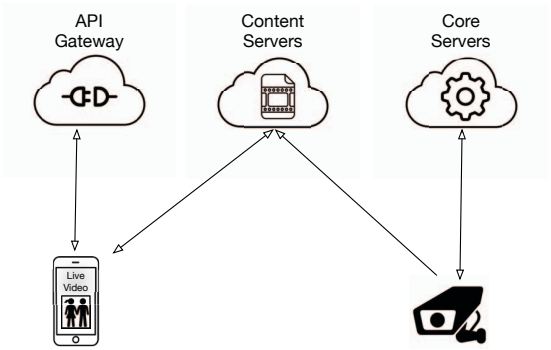

Fig. 1. Managed cloud environments offer centralized frameworks for accessing IoT content. However, vendors often introduce authentication and access control flaws into designs and implementations.

vendors falsely motivate challenges by suggesting a trade-off exists between responsiveness and security.

Organization: Section II provides background, motivation and the adversary threat model. Section III presents our straightforward attack methodology. In Section IV, we document an evaluation of 19 connected cameras and doorbells and summarize our findings. Section V discusses countermeasures to prevent such attacks. Section VI discusses related work. Section VII presents our conclusion.

II. Background & Motivation

A. Overview of IoT Authentication

Managed cloud environments, as depicted in Figure 1, offer the ability for clients to interact with IoT devices through a meet-in-the-middle approach. Managed services (e.g., Amazon IoT Core, Microsoft Azure, TuyaSmart, or Google Home) offer centralized frameworks for accessing IoT content.While these frameworks offer fine-grained access control, device vendors often introduce design and implementation flaws. To better understand how these managed cloud environments deliver IoT content, we describe a few of the key components of these centralized frameworks.

API Gateways provide an interface to the capabilities of IoT devices. API gateways grant authentication tokens to access the low-latency content servers that stream dynamic content (e.g., camera feeds.) To ensure content is available when requested, core servers maintain perpetual connections to IoT devices. Through this perpetual connection, core servers instruct IoT devices to publish their respective content to buckets on the low-latency content servers. Through this meet-in-the-middle approach, clients never directly connect to IoT devices, but are able to access their content through centralized content servers that are generally available via cloud architectures. However, vendors can introduce several flaws to the design and implementation. We describe a subset of these flaws in the next paragraphs.

```
"wakeupServerKey": "<redacted>",
"wakeupServerList": [
    "47.92.3.201:12306",
    "47.254.35.114:12306",
    "47.91.92.46:12306"
    ]
```

Fig. 2. API Servers provide users with the credentials to authenticate to content servers. In this example, the Geenie API provides the authentication and location for streaming video content servers.

```
"access_token": "<redacted>",
"access_token_expires_in": 86400,
"expires_in": 86400,
"refresh_token": "<redacted>",
"refresh_token_expires_in": 63072000,
```

Fig. 3. Content-server tokens with lengthy timeouts enable attackers to connect after a password change. In this example, a user is allowed twenty-four hours of access to SmartThings content servers before the token expires.

B. Authentication and Access Control Flaws

Lengthy Token Expiration: A substantial flaw that enables persistent access occurs when vendors fail to enforce proper content feed timeouts. As illustrated in Figure 2, API Gateways provide the credentials for users to access feeds on content servers. These responses often include the password or token to access feeds on content servers. In most cases, these credentials and content feeds should expire after a brief timeout. A user may need to check a connected doorbell video-feed for thirty seconds. However, a user should not be allowed to access the doorbell hours later with the same token. Unfortunately, vendors fail to include appropriate timeouts, as depicted in Figure 3, as vendors may favor usability over security.

Relaxed Access Control: The always-responsive nature of IoT demands vendors provide low-latency content. To accomplish this, vendors may relax access controls and encryption. Junior et. al [9] demonstrated that as many as 31% of IoT devices fail to enforce encryption. Recently, Xioami and Nest failed to ensure the security and privacy of Xioami camera feeds by storing decrypted content in a cache. This cache failure, coupled with poor integration into the Nest platform allowed Xioami users to access the content of strangers [8], [10]. While platforms such as Amazon's Simple Storage System (S3) offers fine-grained access control, vendors often fail to implement these controls. Recently, researchers discovered the audio recordings from 583,000 CloudPets teddies stored in unsecured S3 buckets without any access control [11].

Login Auditing: Password reuse is a common authentication flaw that affects any system or service that relies on the *something you know* authentication paradigm. This flaw allows attackers to engage in credential stuffing attacks in which they use credentials such as passwords and PINs gained from previous attacks or from leaked credential lists to gain access to other accounts. This approach was recently used to attack the accounts of 3,000 Ring doorbell accounts by spraying

credentials harvested from other accounts [12]. The prevalence of password reuse has been previously studied. Wash et al. [13] found that users typically reused frequently used, complex passwords over multiple websites, likely because of the perceived difficulty of creating, memorizing, and remembering complex passwords. Similarly, Ur et al. [14] found that users did not find password reuse problematic because they trusted that their reused passwords were strong enough to mitigate this vulnerability. Bailey et al. [15] show that users tend to reuse passwords on high-value accounts such as financial since they typically use stronger passwords on such accounts.

C. Motivation

We motivate our work by examining the negative impact poor authentication and access control schemes have on intimate partner violence (IPV). Unauthorized IoT access poses a threat to IPV victims since attackers can leverage IoT devices to intimidate, threaten, monitor, and harass victims [3]. Although IoT devices offer the promise of security with on-demand access to incident maps and taglines such as *Whole Home Protection*, poorly designed access controls can magnify the effects of physical intimate partner violence [16]. Electronic stalking and surveillance present a difficult problem for victims to counteract because IoT devices lack transparency and control to protect victims. Victims are unaware of who is connected and lack the fine-grained access control to restrict unauthorized access. The straightforward attack described in our work illustrates this problem. In our approach, the attacker does not require an arsenal of hacking tools or deep knowledge of networking and device protocols but instead executes within the context of the companion app user interface. In the following paragraphs, we detail the goals, capabilities, and assumptions of such an attacker.

D. Threat Model

Attacker Goals: We consider an attacker whose goal is to retain access to a device's core functionality after an authentication or access control modification or revocation. As an illustration, we imagine a divorced spouse who can surreptitiously monitor a home surveillance camera after being removed from the shared camera by their former partner.

Attacker Capabilities: We consider a technically naïve attacker. In terms of technical knowledge, we assume that the attacker or adversary is what Freed *et al.* aptly term a *UI-bound adversary* [3], and will use this term throughout the paper to mean an authenticated user with adversarial intentions who uses the standard UI provided by IoT devices and services to gain access to information used to control or harass a victim. Thus, the attacker may not need to have technical knowledge beyond app usage details. Specifically, we do not expect the attacker to know device specification details such as protocols used, network, cloud setup or security details that would allow access via means other than the vendor-provided UI.

Attacker Assumptions: Our approach relies on the condition that the attacker has been authorized to access a device's functionality. This access may have been legitimate, as in the case of a domestic partnership or shared residence. Or authentication and access may have also come through the illegitimate disclosure of password credentials (e.g., a password dump.) We assume the attacker can initiate the attack before being removed from the device by an access control modification or revocation.

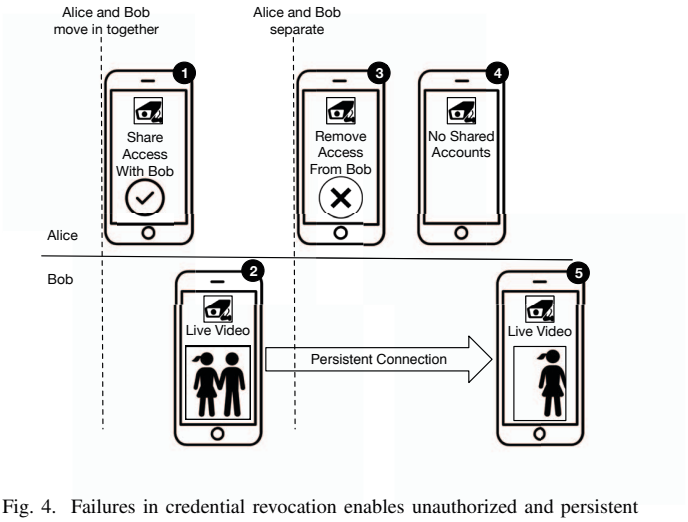

Fig. 4. Failures in credential revocation enables unauthorized and persistent access for shared IoT ecosystems. In this example, Bob retains access despite Alice's attempts to revoke him from a shared camera system.

III. Attack Overview

Figure 4 depicts our straightforward attack method. In this example, subjects Bob and Alice initially share a residence. Bob's goal is to gain indefinite access to Alice's IoT camera system. While living together, Alice grants Bob the ability to view streaming video feeds on the camera system (❶). Next, Bob uses the companion application to request a streaming video feed. The API grants Bob a content-server token, permitting access to a streaming video feed (❷). As the device owner, Alice revokes Bob's access when the two separate (❸). This revocation provides Alice with a message that the API has successfully removed Bob's access (❹). While Bob can no longer connect to the API to get new tokens, his current token is valid and Bob remains surreptitiously connected to the camera live feed (❺). Bob can now use this access to monitor, stalk, and harass Alice. Alice is completely unaware of Bob's access as she believes she successfully revoked his access. In the next section, we reproduce this attack against 19 popular IoT cameras and doorbells.

IV. Evaluation

A. Experiment Setup

We set up a lab environment to examine the vulnerability of IoT devices to our proposed attack vector. We connected all devices to a WiFi network and paired them with companion applications running on two Android phones. When companion applications permitted, we made two accounts, one on each phone, that shared access to the device.

TABLE I

TO DEMONSTRATE THE SYSTEMIC FLAWS IN IOT AUTHENTICATION AND ACCESS CONTROL, WE MEASURED THE EFFECTIVENESS OF ACCESS MODIFICATION AND REVOCATION USING 19 POPULAR CONNECTED CAMERAS AND DOORBELLS. OUR RESULTS DEMONSTRATE SYSTEMIC DESIGN AND IMPLEMENTATION FLAWS EXIST IN THESE SCHEMES.

Device	Firmware Version	App Downloads	App Allows Mitmproxy Cert	Account Types	Persist After Password Change	Persist After Account Revocation
Arlo Camera	1.092.0.24_985	1,000,000+	No	Multiple	*	○
Blink Camera	2.151	1,000,000+	Yes	Single	○	-
Canary Camera	4.0.0	100,000+	No	Multiple	○	●
D-Link Camera	1.05.00	1,000,000+	No	Single	○	-
Geeni Mini Camera	2.7.2	1,000,000+	Yes	Multiple	●	●
Geeni Doorbell	1.8.1	1,000,000+	Yes	Multiple	●	●
Geeni Pan/Tilt Camera	1.3.5	1,000,000+	Yes	Multiple	●	●
Merkury Camera	2.7.2	1,000,000+	Yes	Multiple	●	●
Momentum Axel Camera	51.8	100,000+	Yes	Single	⊙	-
Nest Camera	Current	5,000,000+	Yes	Multiple	⊙	○
Nest Doorbell	Current	5,000,000+	Yes	Multiple	⊙	○
NightOwl Doorbell	WDB-20-V2-20190505	100,000+	Yes	Multiple	⊙	●
Ring Pro Doorbell	Current	5,000,000+	No	Multiple	○	◑
Ring Standard Doorbell	Current	5,000,000+	No	Multiple	○	◑
Samsung Camera	3.6.29.3.3P	100,000,000+	Yes	Multiple	*	●
SimpliSafe Camera	Current	500,000+	Yes	Single	●	-
SimpliSafe Doorbell	Current	500,000+	Yes	Single	●	-
Tend Secure Camera	00.15.009	50,000+	Yes	Multiple	*	●
TP-Link Kasa Camera	2.2.31	1,000,000+	No	Single	◑	-

* : Device does not allow multiple logins of same account
○: Video stream access revoked within 1 minute
●: Video stream access not revoked after 30 minutes
◑: Video stream access revoked within 10 minutes
⊙: Neither video stream access nor API access revoked after 30 minutes

Tested Devices: We evaluated 19 popular connected cameras and doorbells available in 2019. To determine the popularity of each device, we report in Table I the number of application downloads for the companion application on the Google Play Store. When possible, we intercepted the SSL traffic of companion applications using mitmproxy [17] to analyze the communication to Gateway APIs and content servers. We noted applications that did not enforce SSL pinning.

Labeled Results: To evaluate an attacker's ability to retain device access after a password change, we connected to the video stream of the IoT devices using the Android companion applications. We then changed the password using the companion application on a separate device and noted the time until access was revoked from the first device (see Table I). Cases where the companion application revoked access immediately are denoted with ○. We labeled feeds that were revoked within 10 minutes with ◑. Feeds that were available for over 30 minutes are labeled with ●. We used a ⊙ to indicate that the companion application could receive the current feed as well as conduct administrative functions (e.g., viewing previous motion events or controlling access to the device) through the companion application. We then repeated the experiment to evaluate the impact of access revocation using a separate account. To evaluate this, we connected to a video stream using the first account on the companion application on one device. On a separate device, we connected to the IoT device using a second account to revoke the first user's access. We labeled our results using the same scheme as described previously.

B. Evaluation Results

Table I summarizes our evaluation results. Our experiments showed that 16 of 19 devices suffered from either an authentication or access control flaw that permitted an attacker access an IoT device after a password change or account revocation. Further, 4 of the 19 devices permitted access to IoT API servers after a password change. These results confirm our hypothesized attack vector and offer insight into the systemic nature of the problem. In the following paragraphs, we discuss key findings that include isolation, immaturity, and insight problems that accompany IoT. Despite these findings, we argue that problem constraints are falsely motivated and vendors are capable of realizing secure solutions.

C. Evaluation Findings

Finding 1: Isolation between IoT API servers and low-latency content-servers enables unauthorized IoT access: Our experiments showed that API servers often isolate low-latency content servers. This division forces API servers and low-latency content servers to use different versions of access control lists. As an example, 10 devices restricted a user's interaction with the companion app controls through the API but permitted streaming video feeds from low-latency content servers. When a device owner explicitly revokes access, they falsely assume that revocation propagates to both the API and the low-latency content servers. However, in these 10 devices, the device owner only revokes access to the API. Revocation to low-latency content servers does not occur for the hours or days until the content-server authentication token expires.

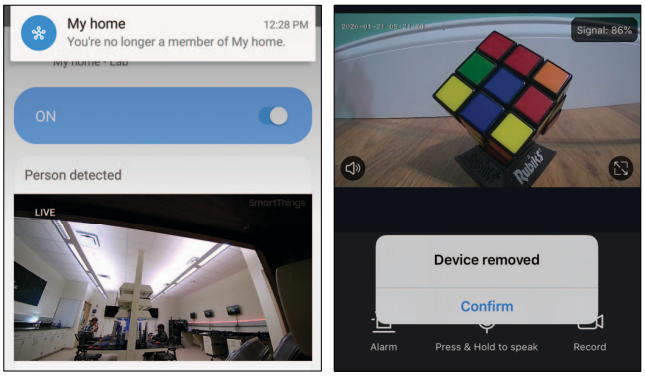

Fig. 5. Access control flaws permit a a user to view a streaming video feed despite access revocation.

Finding 2: IoT designs do not consider UI-bound adversaries: IoT vendors often fail to consider situations where an attacker can leverage the UI as a component of the attack. Recent approaches to IoT security focus on protecting data-in-transit or enacting system controls to protected embedded hardware. However, this often blinds IoT vendors from adversaries that leverage the UI instead of advanced hacking techniques. Figure 5 depicts such an attack where the API removes the attacker via the companion app's UI, but the attacker is still able to view a device's streaming video feed.

Finding 3: IoT provides limited transparency of connections and access control: Traditional hosts and networks offer a broad array of intrusion detection and prevention systems. However, a key finding of this paper is that IoT lacks this same transparency. In our experiments, we found limited notifications that accurately described the correct state of the access control list or could describe connected users. 12 companion applications incorrectly displayed the list of shared accounts, giving users a false sense of security that previous owners had been fully revoked.

Finding 4: Access control is achievable without affecting usability: Device responsiveness is often falsely used to motivate a vendor's ability to perform immediate and verifiable revocation [5]. However, our experiments identified that Arlo, Blink, and D-Link cameras immediately revoked access when requested by the user. While a user study would be needed to further examine this finding, there were no noticeable performance impacts on either device during the course of our experiments.

V. Discussion

A. Potential Countermeasures

In the following section, we introduce transparency and control methods to prevent our proposed attack. In doing so, we examine the trade-offs between security and usability.

Connection Transparency: The limited user interfaces of IoT devices lack the transparency of traditional hosts. Companion apps offer an opportunity to overcome this gap in transparency by providing an interface to view the device behaviors and actions. However, IoT companion apps often focus on simplifying the user experience. This trade-off between usability and security is a key component of IoT vulnerabilities. Providing a list of current connections to the device owner could mitigate the effects of our proposed attack vector. To this end, a device owner could unplug a compromised IoT device. However, identifying a notification scheme presents a challenge. Hiding the connection transparency list inside a companion app view would require the user to access the companion app. In contrast, presenting a device level notification with every new connection could make such warnings ineffective [18]. To combat this possible warning fatigue, we recommend future work should study how appropriately deliver context-aware notifications. One method for gaining context is to leverage device and companion app analytics to develop user insight algorithms designed to deliver notifications when the users need them most.

Credential Insight Algorithms: Credential insight algorithms offer the potential to identify anomalous credential usage. IoT devices and their companion applications produce a wealth of context and analytics as possible input to these algorithms, including IP addresses, geo-location data, companion device operating system details, and user behavior patterns. Vendors such as Synamedia [19] have proposed leveraging credential insight algorithms to detect unauthorized use and credential sharing for online media services such as Hulu, Netflix, and Disney+. These approaches leverage predictive analytics to detect credential sharing and to implement automated responses. Further, credential sharing algorithms can aid in detecting credential compromises. Applied to IoT, these same algorithms can identify anomalous logins to device APIs and low-latency content servers to identify stolen and reused credentials. This approach moves defense into the always-responsive cloud, leveraging the processing and storage not available at the device level. However, this approach requires further study to realize a viable solution.

VI. Related Work

IoT Camera Attacks: Many papers in this area focus on outside attackers gaining access to IoT cameras via vulnerabilities in the camera itself or their connection protocols. Heffner [20] showed that more than 50 IoT cameras were vulnerable to several 0-day vulnerabilities on the cameras themselves that allowed an outside attacker to gain access to the camera feed, as well as replace the feed with a static image. This replacement of the feed with a static image is of most interest to the work described in this paper, as it represents the opposite of what we show here: that an attacker can choose what a legitimate user sees via the camera. Further, O'Connor *et al.* [21] demonstrated the ability to transparently blind camera systems by selective traffic forwarding. Seralathan *et al.* [22] extended Heffner's work to show the vulnerabilities in the network protocols rather than the camera itself. They showed that a Netgear IP-based camera was vulnerable to outside

attackers via unencrypted network data sent between the camera and the cloud, including cleartext camera credentials (e.g., SSID and password). Their suggestions for mitigation were limited to encrypting network data [22].

UI Bound Adversary Attacks: Rather than addressing external attackers, we focus instead on those with legitimate access to a camera retaining the camera feed after their credential revocation. Technology-facilitated abuse, or *tech abuse* has been a topic of research in IoT since home-based IoT devices, including cameras, can be used to observe or control the behavior of victims and survivors of domestic abuse. This area of research is directly related to our work since it is assumed that the abuser and the victim both have legitimate access to the IoT-enabled camera. Freed *et al.* [3] define the term *UI-bound adversary* to describe an authenticated attacker that interacts with target devices via their standard user interface only. This is markedly different from the attackers described in the previous paragraph since it implies that deep knowledge of device inner workings or network protocols is not required to facilitate device control. However, while Freed *et al.* (as well as similar work in the tech abuse research area [4], [23], [24]) identify important means by which UI bound adversaries can launch attacks, they do not focus upon revocation of legitimate access to IoT devices. Thus, we find that there is a need for this type of research, given that it focuses on a vulnerability that may cause an abuse victim to remain in the abuser's control.

VII. Conclusion

Securing the privacy of IoT content is crucial to protecting user privacy. In this study, we analyzed the authentication and access control schemes of 19 popular security cameras and connected doorbells. We hypothesized and implemented an attack to gain persistent access to IoT content. Our results demonstrated that 16 of 19 devices suffered from an authentication or access control flaw that permitted an attacker access to an IoT device after a password change or account revocation. Our analysis identified a systemic failure in device authentication and access control schemes for shared IoT ecosystems. Our study suggests there is a long road ahead for vendors to implement the security and privacy of IoT produced content.

References

[1] S. W. Fu, H. G. Sampson, S. Keenan, and B. Liang, "Package theft prevention device with an internet connected outdoor camera," Google Patents, Tech. Rep., Aug 2019, US Patent 10,389,983.

[2] R. Chatterjee, P. Doerfler, H. Orgad, S. Havron, J. Palmer, D. Freed, K. Levy, N. Dell, D. McCoy, and T. Ristenpart, "The spyware used in intimate partner violence," in *Symposium on Security and Privacy*. IEEE, May 2018, pp. 441–458.

[3] D. Freed, J. Palmer, D. Minchala, K. Levy, T. Ristenpart, and N. Dell, "A stalker's paradise: How intimate partner abusers exploit technology," in *Proceedings of Conference on Human Factors in Computing Systems (CHI)*, 2018, pp. 1 – 13.

[4] T. Matthews, K. O'Leary, A. Turner, M. Sleeper, J. P. Woelfer, M. Shelton, C. Manthorne, E. F. Churchill, and S. Consolvo, "Stories from survivors: Privacy & security practices when coping with intimate partner abuse," in *SIGCHI Conference on Human Factors in Computing Systems*, 2017, pp. 2189 – 2201.

[5] A. Carman, "Ring's smart doorbell doesn't immediately revoke access when an account password changes," May 2018. [Online]. Available: https://www.theverge.com/circuitbreaker/2018/5/11/17345972/ring-smart-doorbell-password-change-revoke-app-permission-access

[6] N. Vigdor, "Somebody's watching: Hackers breach ring home security cameras," Dec 2019. [Online]. Available: https://nyti.ms/36DHILA

[7] N. Karlis, "A huge security camera company just had a huge security breach," Jan 2020. [Online]. Available: https://www.salon.com/2020/01/01/a-huge-security-camera-company-just-had-a-huge-security-breach/

[8] A. Stanley, "Google suspends xiaomi's nest integration after user appears to pick up strangers' camera feeds," Jan 2020. [Online]. Available: https://gizmodo.com/google-suspends-xiaomis-nest-integration-after-it-picks-1840783978

[9] D. M. Junior, L. Melo, H. Lu, M. d'Amorim, and A. Prakash, "A study of vulnerability analysis of popular smart devices through their companion apps," in *Security and Privacy Workshops*. IEEE, 2019, pp. 181–186.

[10] K. Lyons, "Xiaomi says issue that showed strangers images on nest devices is identified but not fully resolved," Jan 2020. [Online]. Available: https://www.theverge.com/2020/1/3/21048061/xiaomi-strange-images-google-nest-devices-identified-fixed

[11] "Data from connected cloudpets teddy bears leaked and ransomed, exposing kids voice messages," Dec 2017. [Online]. Available: https://www.troyhunt.com/data-from-connected-cloudpets-teddy-bears-leaked-and-ransomed-exposing-kids-voice-messages/

[12] B. Molina, "Personal information on more than 3,000 ring owners reportedly compromised," Dec 2019. [Online]. Available: https://www.usatoday.com/story/tech/2019/12/20/ring-camera-data-more-than-3-000-owners-reportedly-compromised/2707943001/

[13] R. Wash, E. Rader, R. Berman, and Z. Wellmer, "Understanding password choices: How frequently entered passwords are re-used across websites," in *Symposium on Usable Privacy and Security*, 2016, pp. 175 – 188.

[14] B. Ur, F. Noma, J. Bees, S. M. Segreti, R. Shay, L. Bauer, N. Christin, and L. F. Cranor, ""i added '!' at the end to make it secure": Observing password creation in the lab," in *Symposium on Usable Privacy and Security*, 2015.

[15] D. V. Bailey, M. D urmuth, and C. Paar, "Statistics on password re-use and adaptive strength for financial accounts," in *International Conference on Security and Cryptography for Networks*, vol. 8642, 2014.

[16] A. E. Bonomi, R. S. Thompson, M. Anderson, R. J. Reid, D. Carrell, J. A. Dimer, and F. P. Rivara, "Intimate partner violence and women's physical, mental, and social functioning," in *American journal of preventive medicine*, vol. 30, no. 6, 2006, pp. 458 – 466.

[17] A. Cortesi, M. Hils, T. Kriechbaumer, and contributors, "mitmproxy: A free and open source interactive HTTPS proxy," 2010–, [Version 5.0]. [Online]. Available: https://mitmproxy.org/

[18] G. S. Bahr and R. Ford, "How and Why Pop-Ups Don't Work: Pop-Up Prompted Eye Movements, User Affect and Decision Making," *Computers in Human Behavior*, vol. 27, no. 2, pp. 776 – 783, 2011.

[19] "Synamedia launches credentials sharing insight – turns casual password sharing into incremental revenues for service providers," Apr 2019. [Online]. Available: https://www.synamedia.com/press/synamedia-launches-credentials-sharing-insight-turns-casual-password-sharing-into-incremental-revenues-for-service-providers/

[20] C. Heffner, "Exploiting surveillance cameras like a hollywood hacker," in *Blackhat*, 2013.

[21] T. OConnor, W. Enck, and B. Reaves, "Blinded and confused: Uncovering systemic flaws in device telemetry for smart-home internet of things," in *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*. Miami,FL: ACM, 2019.

[22] Y. Seralathan, T. T. Oh, S. Jadhav, J. Myers, J. P. Jeong, Y. H. Kim, and J. N. Kim, "IoT Security Vulnerability: A Case Study of a Web Camera," in *International Conference on Advanced Communication Technology*, 2018, pp. 172 – 177.

[23] B. E. Soric, K. K. R. Choo, H. Ashman, and S. Mubarak, "Stalking the stalkers - detecting and deterring stalking behaviours using technology: A review," in *Computers & Security*, vol. 70, 2017, pp. 278 – 289.

[24] J. Vitak, K. Chadha, L. Steiner, and Z. Ashktorab, "Identifying women's experiences with and strategies for mitigating negative effects of online harassment," in *ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW)*, 2017, pp. 1231 – 1245.