Toward Usable Network Traffic Policies for IoT Devices in Consumer Networks

Nicholas DeMarinis
Brown University
Providence, RI, USA
ndemarin@cs.brown.edu

Rodrigo Fonseca
Brown University
Providence, RI, USA
rfonseca@cs.brown.edu

ABSTRACT

The Internet of Things (IoT) revolution has brought millions of small, low-cost, connected devices into our homes, cities, infrastructure, and more. However, these devices are often plagued by security vulnerabilities that pose threats to user privacy or can threaten the Internet architecture as a whole. Home networks can be particularly vulnerable to these threats as they typically have no network administrator and often contain unpatched or otherwise vulnerable devices.

In this paper, we argue that the unique security challenges of home networks require a new network-layer architecture to both protect against external threats and mitigate attacks from compromised devices. We present initial findings based on traffic analysis from a small-scale IoT testbed toward identifying predictable patterns in IoT traffic that may allow construction of a policy-based framework to restrict malicious traffic. Based on our observations, we discuss key features for the design of this architecture to promote future developments in network-layer security in smart home networks.

KEYWORDS

Internet of Things (IoT); Intrusion Detection; Network Security; Home Networks

1 INTRODUCTION

The Internet is going through a new phase in which billions of new types of devices are getting connected in homes, industry, cities, and agriculture, bringing unprecedented possibilities of understanding, automation, and improvement of our physical environment. This revolution, the 'Internet of Things' (IoT), is enabled by a combination of ubiquitous connectivity, devices that are cheap, small, and powerful, and the practically unlimited storage and processing power of the cloud. According to Gartner Inc., for example, forecasts over 20 billion IoT endpoints by 2020, with a sustained annual growth of 33% from 2015 [13].

This new wave of devices creates new requirements for infrastructure, new traffic patterns, and, unfortunately, new vectors and surfaces of attack [28]. For example, in 2016 alone multiple vulnerabilities in IoT devices were exploited for large-scale privacy violations, such as the 'baby-camera search engine' [9], and some of the largest-yet-seen DDoS attacks which brought down prominent blogs and large part of the DNS infrastructure in the US [11]. Toys have leaked recordings of parent-children conversations [10] to the Internet, and, in a spectacular proof-of-concept attack, a drone was able to control the smart lights of an entire building by flying by [20]. The security and privacy implications are vast, and if not addressed will slow down, if not downright prevent, the benefits that the technology can bring to society. As the examples above indicate, vulnerabilities can lead to privacy violations, control and hazardous operation of devices by unauthorized actors, and use of the devices for attacks on other infrastructure.

While there is a vast body of work in network security, there are important differences in the IoT environment that may require revisiting assumptions and techniques used to detect and mitigate various forms of attack. For example, there is a much larger variety of devices; many provide very little feedback of their activities to users, are located in networks with effectively no administrator, and cannot usually run extra software to provide robust security or detect attacks. It is hard for even the manufacturer to patch the devices if they leave the factory with a vulnerability [23].

Some of the differences present potential opportunities: most IoT devices are single, or at least, limited-function devices, which can greatly constrain the way they interact with the network. Given these characteristics, and the increased prevalence of programmable network devices, the gateway between a set of IoT devices and the Internet becomes an attractive point to observe the devices' interaction with the network, as well as to deploy software to detect and, potentially, mitigate the effect of attacks in both directions.

In this paper, we claim that a new network-layer architecture is necessary to to protect against threats from easily-vulnerable IoT devices and present our first steps toward designing a secure architecture targeted for consumer network gateways (*i.e.*, home routers). Our approach is based on the premise that IoT devices perform a small set of functions that follow predictable patterns in network traffic compared to a general-purpose system, thus allowing the construction of policies to restrict their traffic. We specifically focus on devices for home networks, which typically have few (if any) security measures, but can pose a threat to their users' privacy, or, in coordination many similar devices, the Internet architecture as a whole. Our contributions in this paper are as follows:

- Discuss our hypothesis about how (uncompromised) IoT traffic is predictable and thus may be feasible to filter using a policy-based approach

IoT-S&P'17, , November 3, 2017, Dallas, TX, USA.

ACM ISBN 978-1-4503-5396-0/17/11…$15.00
https://doi.org/10.1145/3139937.3139949

- Present initial findings analyzing traffic from a small-scale IoT testbed showing how certain traffic characteristics may support our hypothesis, and demonstrate key areas where it does not
- Outline a preliminary set of requirements for a policy-based approach informed by our testbed results

While our work to date has involved only a small set of devices, our analysis so far has yielded interesting results that help define the scope of a network-layer approach to securing smart homes. We hope to use these results to support further development of a secure architecture and promote discussion on improving IoT infrastructure.

Threat model: Our work focuses on preventing attacks on IoT devices and applications directed to or from networks other than those used by the devices for normal operation. In this way, we aim to mitigate attacks from unknown external sources (e.g., exploiting a device's insecure management interface) or remote attacks launched from device (e.g., leaking user information from outside sources or participating in a DDoS attack) that may be caused by vulnerabilities in device platforms. We do not focus on physical- or link-layer IoT protocol security (ZigBee, Bluetooth, etc.), which is already well-studied, and instead focus on the security of the endpoints between these devices and the Internet, where vulnerabilities may be more widespread. Access control and permission systems provided by devices for controlling usage and sharing of data are also important, but orthogonal, issues outside our scope. Instead, we specifically assume that devices are insecure by default and advocate for a network-layer solution that does not require changes to the devices and can work in concert with other device- or protocol-specific security measures.

2 BACKGROUND AND RELATED WORK

A wide range of IoT devices are now available for home use: including smart appliances, lights, toys, personal assistants, locks, smoke detectors, and more. These smart home devices, or "things", rely on communication with online services in the cloud or with other local devices to provide functionality. Despite this reliance on sharing, IoT devices have typically lacked robust security measures due to limited hardware resources or lack of consideration during design. These vulnerabilities have been widely exploited [8, 9, 18] for purposes such as leaking private information, tampering with device resources, or participating in DDoS attacks. Worse still, many devices provide no support for firmware updates to fix vulnerabilities, or only permit manual upgrades by a user, leading to a persistent ecosystem of vulnerable devices on the Internet [8, 23].

Much work has been conducted to strengthen IoT security protocols and add permissions and access controls to protect user data—we refer the reader to [14] and [5] for more detailed surveys. Of particular interest to modern smart homes are commercial IoT platforms, such as Samsung SmartThings [21] and Apple HomeKit [2], which provide a framework enabling developers to write custom applications for supporting devices, enabling complex functionality while maintaining some standards for transport security and access controls. These systems represent a promising step forward for IoT security, but the devices or the platform itself may still contain vulnerabilities such as overprovisioning of permissions [5], or device-specific issues like protocol misuse or exposed debug interfaces [27]. Addressing the permissions problems is an area of ongoing work [5, 6], but the security of the underlying platform that provides these facilities is also critical to protect against out-of-band vulnerabilities.

Moreover, modern IoT platforms only provide security to devices that support it. Legacy or otherwise unsupported devices may still have inadequate security measures (or none at all), may never receive updates, and could take years to be replaced [23]. Thus, a comprehensive platform for securing a smart home must also include a component at the network-layer to provide an additional line of defense working in coordination with, or in place of, any device-specific security measures.

Network-layer security schemes in IoT contexts are not new. As one example, SVELTE [19] provides network intrusion detection for IoT but focuses on routing attacks in sensor networks. Home networks present a challenging environment for a network-layer solution due to the limited resources available on a home gateway router. IoT-IDM [15] directs IoT traffic to an IDS using OpenFlow, but requires additional hardware to run the IDS and OpenFlow controller. Consumer-grade routers typically use a low-power (< 500MHz) ARM or MIPS processor with very limited memory or storage [17], making these methods infeasible without hardware changes. Coping with these processing difficulties may require a centralized solution where traffic analysis is "outsourced" to an external server [4]. However, this contradicts many important security requirements due to the sensitive nature of IoT data (e.g., camera feeds, health data, voice recordings), as well as latency requirements, which mandate that certain sensitive or real-time data should be handled locally.

In light of these challenges, we argue that a new network-layer architecture is necessary in order to protect vulnerable devices within the available resource constraints of a home network. In the next section, we discuss our current hypothesis on how the characteristics of IoT traffic may make analysis on a home router more tractable and describe an initial design for this architecture.

3 INVESTIGATING TRAFFIC PATTERNS

3.1 Hypothesis

IoT devices are special-purpose: they typically perform a fixed set of tasks based on the device's hardware. For example:

- A security camera may send video data or motion events to a remove server, or receive requests to record events
- A smart plug our outlet may receive events to toggle the plug's state, or send periodic updates to monitor energy usage
- A media device, such as a smart TV, may play various media streams from various content providers

This limited functionality implies that many IoT devices connect to relatively few *entities* on the Internet—most namely limited to the remote servers maintained by the manufacturer. These cloud-based services are critical to modern IoT infrastructure and often provide the necessary control logic and resources to implement complex functionality. While authorized users and devices may connect to an IoT device, this typically occurs indirectly through the cloud.

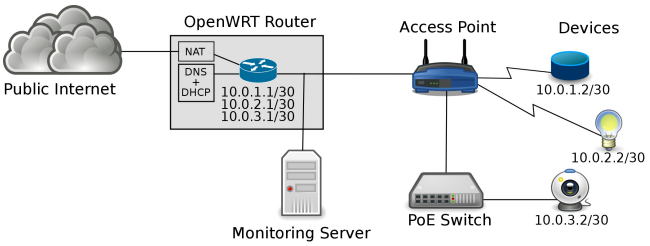

Figure 1: Testbed architecture for IoT device characterization

In contrast, a general-purpose system like a workstation or smartphone, connects to a very high number of entities through various applications or web browsing. Some IoT devices may perform multiple functions while still connecting to few entities: as one example, the Samsung SmartThings hub, which controls an array of smart home devices, but performs all of its control logic via Samsung's cloud API [22]; this appears to be a common trend as devices become more complex.

We hypothesize that the limited number of connections makes traffic analysis on a resource-constrained home gateway more tractable. Since the set of entities is limited, this may allow construction of lightweight access control policies, or whitelists, to restrict communication to these trusted entities and block other traffic. If applied correctly, this scheme could prevent, for example, a malicious host attacking an exposed management interface, or a compromised device from sending packets to untrusted sources, such as leaking data to a botnet or participating in a DDoS attack.

While the core of a whitelist-based approach may be enforced by simple ACL rules, constructing accurate policies that can identify and permit only legitimate traffic introduces significant logistical challenges. To explore the feasibility of this approach, we observed traffic patterns from a small set of devices, discussed in Sections 3.2 and 3.3. We discuss the implications of our findings for designing an architecture that incorporates whitelist-based policies in Section 4.

3.2 Small-scale IoT testbed

To begin a preliminary analysis of IoT device traffic patterns, we developed a basic testbed for measuring traffic from a small number of common devices, shown in Figure 1. Our goal was to create a network similar to a typical home network, while allowing us to capture all traffic from the devices for analysis. We used a TP-Link router [25] with OpenWRT [16] firmware to provide NAT, DNS, and DHCP services, mimicking the services typically provided by a home router. OpenWRT firmware provides us with a minimal Linux distribution on the router, which allows a high degree of flexibility to configure the services provided for the devices.

Test devices are connected to a commodity wireless access point, such that traffic between the access point and the router can be mirrored to a monitoring server. In addition, we include a Power over Ethernet (PoE) switch to accommodate devices that require it, such as IP cameras. In order to capture traffic sent between devices in our testbed, all hosts are assigned IP addresses in individual /30 subnets so that all local traffic must be first directed to the router. This extra forwarding step adds a small amount of latency (≈10 ms) to inter-device traffic, which is negligible in a home network scenario.

Device	Total DNS Queries	Unique DNS Queries
Echo	1231 (24.0)	26 (0.0)
Chromecast	7297 (150.5)	58 (0.0)
Smart Plug	56 (1.0)	4 (0.0)
Laptop	9432 (174.0)	657 (2.5)

Table 1: Summary of DNS traffic observed in IoT testbed. Numbers in parenthesis represent the median number of queries per hour.

Our testbed is currently comprised of four devices: an Amazon Echo Dot [1], a Google Chromecast [7], a TP-Link smart plug [26], and a Dahua PoE IP camera [3]. We selected these devices for initial testing to provide a diverse range of vendors and traffic types. While this is by no means a large or representative sample of the possible types of IoT devices, these four examples provide a starting point to gather initial observations and inform future research questions to help construct a feasible approach.

3.3 Traffic observations

For this analysis, we focused on identifying patterns in the number of hosts with which each device communicates by observing the DNS queries made by each device and the IP addresses it contacts.

We captured traffic from each device during a 48-hour period since its initial configuration. A summary of the DNS traffic observed is shown in Table 1. Overall, we observe that each device makes many repeated DNS queries for the same domains, which correspond to their respective cloud providers. For example, the Chromecast made 7297 total DNS queries to only 58 unique domain names in the 48-hour period; most were to names in Google service domains.

Figure 2 shows the number of unique DNS queries and IP addresses contacted for each device during the measurement period. We show both the total number of unique queries as well as the number of new queries every hour relative to the start of the measurement period. After configuring each device, we performed a number of actions to exercise each device's features (e.g., streaming videos from the Chromecast, asking questions using the Echo, and toggling the smart plug). During our tests, the camera never connected to a host outside the local network or made any non-local DNS queries, so we omit a plot for it.

Our preliminary results show that each device makes relatively few new DNS queries after initialization. Each device performed a median of zero new DNS queries per hour after initialization, which indicates that devices are making repeated queries for the same domain names. Overall, this demonstrates that the devices we observed are communicating with the same few entities over time, supporting hypothesis for building whitelist-based policies from on a small set of entities to account for common-case traffic.

For comparison, Figure 3 shows a similar plot for traffic captured from one of the authors' laptops during normal use to provide one example for traffic from a general-purpose system. The large increases in traffic during hours 20–26 and 40–45 correspond to the times of day when the system was under the heaviest user workload. In total, the laptop queried 657 unique domains during the measurement period at an average of about 30 domains per hour during peak hours, more than ten times more than any of the IoT devices.

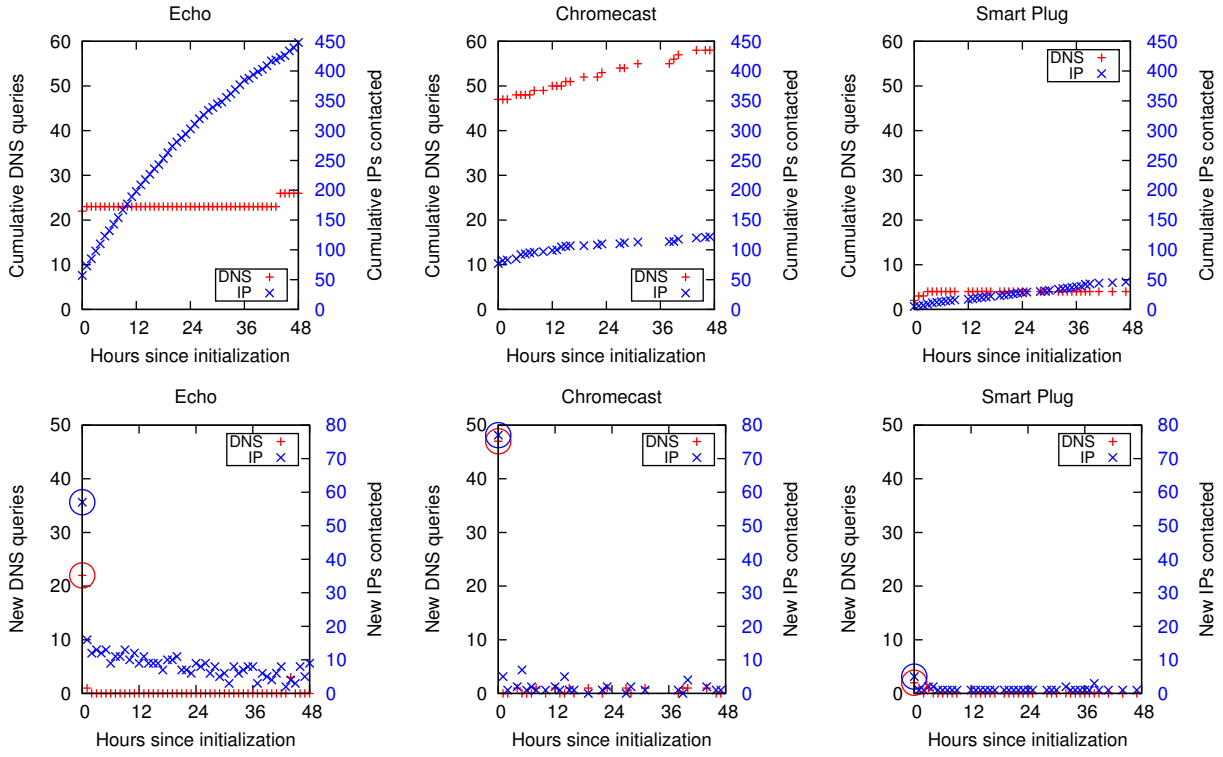

Figure 2: Cumulative (top) and new (bottom) DNS queries and IP addresses contacted by three IoT devices over a 48-hour period since initial configuration. The first hour of IP and DNS traffic in the "new" plots is circled.

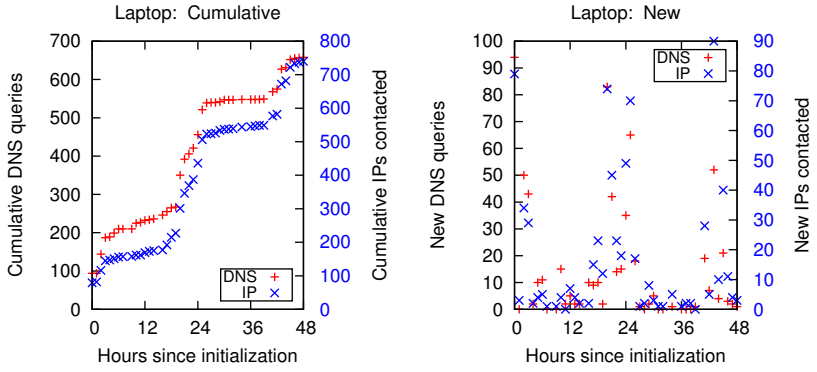

Figure 3: Cumulative (left) and new (right) DNS queries and IP addresses contacted by a general-purpose laptop belonging to one of the authors. Note that both plots have different y-axes.

Critically, we observe two important considerations that challenge our hypothesis. While our devices demonstrate predictable traffic during many operations, certain legitimate device actions initiate connections to entities not used during normal operations. Many functions on the Chromecast and Echo utilize content from third-party sources based on user actions, such as playing third-party audio and video streams. For example, when we asked the Echo to play the stream for a local radio station, the device made several DNS queries and initiated connections to the stream on the station's website. A listing of two example user actions and their respective DNS queries is shown in Table 2. These user actions demonstrate how a device can be directed to load content from an entity other than their main cloud provider, which creates additional challenges for building policies.

We also observe that devices contact a much larger range of IP addresses than DNS names, since the IP address associated with a DNS name may change over time due to load balancing. While it may be desirable to incorporate simple matching of IP prefixes

Action	Domains Contacted
Amazon Echo: Playing a radio stream	opml.radiotime.com wbur-sc.streamguys.com audio.wbur.org
Chromecast: Playing a twitch.tv video stream	secure.twitch.tv cdn.mxpnl.com api.mixpanel.com api.twitch.tv usher.ttvnw.net static-cdn.jtvnw.net video-edge-<id> .jfk03.hls.ttvnw.net

Table 2: Example user actions and associated DNS queries

or AS numbers into our policies, this may not be possible in all cases. For example, the Echo and Chromecast contacted a number of new IP addresses in each measurement period, but most of these are localized to Amazon's and Google's networks, respectively. In contrast, the smart plug's unique IP addresses come from queries to public NTP pools (e.g., `pool.ntp.org`), which results in IP addresses from a very diverse set of networks.

These observations indicate that our device policies must be dynamic enough to account for variations in behavior due to user actions and network-level changes, which provides new research questions to motivate our design. As we continue our work, we plan to perform further studies on other metrics that may further distinguish different types of traffic, such as periodicity, seasonality, and coordination between devices. These features may as highlight further variations that must be taken into account for policy designs. We also plan to perform a systematic study of other traffic features, and also explicitly infect some of the devices with malware to compare traffic patterns in compromised devices. The camera in our testbed, for example, is vulnerable to infection by the Mirai botnet [11].

4 DISCUSSION

Despite our small sample size for traffic measurements, our work has helped us identify important considerations for designing a secure architecture for smart homes using whitelist-based policies. In this section, we list our present set of key features for such an architecture and outline possible approaches for further investigation. We acknowledge that this is very preliminary work: our aim is to present our current findings and promote discussion on smart home security at the network-layer.

F1. Support for user actions and apps: Some devices use data from a variety of sources, such as the streaming media and third-party application features of the Amazon Echo and Google Chromecast from our testbed. While a basic access control list may capture the networks a device may contact during idle time, this is not enough when the device is legitimately directed to make an outgoing connection to another entity, such as a streaming video service. This demonstrates that any whitelist policies must be able to change dynamically to account for changes to traffic based on user actions or configuration changes.

In cases where devices are directed to contact entities by their cloud provider (which is true for the Echo and Chromecast), policies could benefit from a capability-based framework leveraging DNS [24]. Using this approach, a device could obtain a capability from its cloud provider for a certain connection, which is sent to the router and verified against the policy before the connection is allowed. A capability-based system could help ensure that dynamic actions or user-installed applications correspond to legitimate actions permitted by a device policy, rather than malicious traffic from a compromised device.

F2. Secure policy creation and distribution: An area for significant concern is how the device policies would be created, maintained, and distributed. Policies crafted by the application developer or manufacturer may be too permissive, similar to permission systems for smart applications [5]. In addition, policies will require updates to coincide with device firmware updates or discovery of new vulnerabilities. We therefore suggest that policies should be maintained by one or more independent authorities rather than trusting a single party. Similar to vulnerability databases, a policy repository could allow for multiple stakeholders such as manufacturers, organizations, or security-conscious users to transparently submit, audit, and curate policies for many types of devices.

This area also presents an opportunity for automatically generating or verifying policies using automated techniques. For example, machine learning approaches for classifying IoT traffic, such as the work of Meidan et al. [12], could be used to generate or supplement policies to identify anomalies. When combined with one or more central repositories, anomaly detection among home routers could allow for coordination to develop responses for emerging threats.

F3. Resource and privacy limitations: Creation of robust policies, especially when involving coordination between devices and repositories, raises privacy and performance concerns. As discussed in Section 2, home routers typically have very limited computational resources. While it may be advantageous to send raw device traffic to a centralized source for complex processing or to help gather global intelligence, this is not trivial due to the privacy concerns. In our future developments, we will investigate methods for efficiently enforcing policies and gathering intelligence without raw data aggregation and maximizing use of local processing capabilities for latency, efficiency, and privacy.

F4. Device installation and user interaction: Installing a new device on a home network would require an initial configuration process to bind the device to a suitable policy. Conceptually, this requires that the network identify the device is an IoT device, and securely obtain an up-to-date policy for it from the appropriate repository. To facilitate this critical bootstrapping process, a list of trusted repositories could be pre-installed on the router or configured by the user. Ideally, this approach should allow user oversight, while minimizing required interactions as much as possible to ensure policies remain updated.

5 CONCLUSION

Our work has presented the need for a new network-layer architecture targeted specifically for home networks to protect against vulnerable IoT devices. We have provided initial observations of IoT traffic from a small set of devices to suggest that a policy-based approach may be tractable and warrants further investigation to determine its feasibility within resource and privacy constraints. Our analysis, while preliminary, has illuminated key features for

designing this network architecture. In our continued efforts toward this goal, we plan to study additional traffic characteristics on larger and more diverse groups of devices as well as investigate the design for a robust policy framework that supports shared processing capabilities. We hope that the our current findings presented in this work highlight the importance of network-layer security in smart home networks and promote further discussion on how to improve security measures in this area.

REFERENCES

[1] Amazon. 2017. Amazon Echo Dot. (2017). Retrieved August 18, 2017 from https://goo.gl/nMD3Wk

[2] Apple. 2017. HomeKit. (2017). Retrieved August 18, 2017 from https://www.apple.com/ios/home/

[3] Dahua Technology Co. 2017. Dahua IPC-HCB4300C. (2017). Retrieved August 18, 2017 from https://goo.gl/Js1dMo

[4] Nick Feamster. 2010. Outsourcing home network security. In *Proceedings of the 2010 ACM SIGCOMM workshop on Home networks*. ACM, 37–42.

[5] Earlence Fernandes, Jaeyeon Jung, and Atul Prakash. 2016. Security Analysis of Emerging Smart Home Applications. In *Proceedings of the 37th IEEE Symposium on Security and Privacy*.

[6] Earlence Fernandes, Justin Paupore, Amir Rahmati, Daniel Simionato, Mauro Conti, and Atul Prakash. 2016. FlowFence: Practical Data Protection for Emerging IoT Application Frameworks. In *Proceedings of the 25th USENIX Security Symposium*.

[7] Google. 2017. Google Chromecast. (2017). Retrieved August 18, 2017 from https://www.google.com/chromecast/tv/chromecast/

[8] Sarthak Grover and Nick Feamster. 2016. The Internet of Unpatched Things. *Proc. FTC PrivacyCon* (2016).

[9] Alex Hern. 2016. Search engine lets users find live video of sleeping babies. *The Guardian* (January 2016). Retrieved August 18, 2017 from https://goo.gl/BM2FTD

[10] Troy Hunt. 2017. Data from connected CloudPets teddy bears leaked and ransomed, exposing kids' voice messages. Personal Blog, https://goo.gl/TXnlLh. (February 2017).

[11] Brian Krebs. 2016. Hacked Cameras, DVRs Powered Today's Massive Internet Outage. (2016). Retrieved August 18, 2017 from https://goo.gl/eEZOMW

[12] Yair Meidan, Michael Bohadana, Asaf Shabtai, Juan David Guarnizo, Martín Ochoa, Nils Ole Tippenhauer, and Yuval Elovici. 2017. ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis. In *Proceedings of the Symposium on Applied Computing*. ACM, 506–509.

[13] Peter Middleton. 2017. Forecast Analysis: Internet of Things — Endpoints, Worldwide, 2016 Update. Gartner Database (ID: G00302435). (February 2017).

[14] Arsalan Mohsen Nia and Niraj K Jha. 2016. A comprehensive study of security of internet-of-things. *IEEE Transactions on Emerging Topics in Computing* (2016).

[15] Mehdi Nobakht, Vijay Sivaraman, and Roksana Boreli. 2016. A Host-Based Intrusion Detection and Mitigation Framework for Smart Home IoT Using OpenFlow. In *Availability, Reliability and Security (ARES), 2016 11th International Conference on*. IEEE, 147–156.

[16] OpenWRT. 2017. OpenWRT. (2017). https://openwrt.org.

[17] OpenWRT. 2017. OpenWRT: Table of Hardware. (2017). Retrieved August 18, 2017 from https://wiki.openwrt.org/toh/start

[18] Laura Rafferty, Farkhund Iqbal, and Patrick CK Hung. 2017. A Security Threat Analysis of Smart Home Network with Vulnerable Dynamic Agents. In *Computing in Smart Toys*. Springer, 127–147.

[19] Shahid Raza, Linus Wallgren, and Thiemo Voigt. 2013. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad hoc networks* 11, 8 (2013), 2661–2674.

[20] Eyal Ronen, Colin O'Flynn, Adi Shamir, and Achi-Or Weingarten. 2016. IoT Goes Nuclear: Creating a ZigBee Chain Reaction. Cryptology ePrint Archive, Report 2016/1047. (2016). http://eprint.iacr.org/2016/1047.

[21] Samsung. 2017. SmartThings. (2017). Retrieved August 18, 2017 from https://www.smartthings.com

[22] Samsung. 2017. SmartThings Developer Documentation: Architecture. (2017). Retrieved August 18, 2017 from http://docs.smartthings.com/en/latest/architecture/index.html

[23] Bruce Schneier. 2014. The Internet of Things Is wildly insecure - and often unpatchable. (2014). Retrieved August 18, 2017 from https://goo.gl/a7sxjn

[24] Craig A Shue, Andrew J Kalafut, Mark Allman, and Curtis R Taylor. 2012. On building inexpensive network capabilities. *ACM SIGCOMM Computer Communication Review* 42, 2 (2012), 72–79.

[25] TP-Link Technologies Co. 2017. TP-Link TL-WR1043ND V1 Router. (2017). https://goo.gl/OLwXYY.

[26] TP-Link Technologies Co. 2017. TP-Link Wi-Fi Smart Plug HS100. (2017). Retrieved August 18, 2017 from https://goo.gl/3IMBR5

[27] Veracode. 2017. The Internet of Things: Security Research Study. (2017). Retrieved August 18, 2017 from https://www.veracode.com/sites/default/files/Resources/Whitepapers/internet-of-things-whitepaper.pdf

[28] Tianlong Yu, Vyas Sekar, Srinivasan Seshan, Yuvraj Agarwal, and Chenren Xu. 2015. Handling a Trillion (Unfixable) Flaws on a Billion Devices: Rethinking Network Security for the Internet-of-Things. In *Proceedings of the 14th ACM Workshop on Hot Topics in Networks (HotNets-XIV)*. ACM, New York, NY, USA, Article 5, 7 pages. https://doi.org/10.1145/2834050.2834095