A PRINCIPLED APPROACH TO MEASURING THE IOT ECOSYSTEM

BY

DEEPAK KUMAR

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Doctoral Committee:

Associate Professor Michael Bailey, Chair
Professor Nikita Borisov
Assistant Professor Adam Bates
Assistant Professor Gang Wang
Assistant Professor Zakir Durumeric, Stanford University

ABSTRACT

Internet of Things (IoT) devices combine network connectivity, cheap hardware, and actuation to provide new ways to interface with the world. In spite of this growth, little work has been done to measure the network properties of IoT devices. Such measurements can help to inform systems designers and security researchers of IoT networking behavior in practice to guide future research.

Unfortunately, properly measuring the IoT ecosystem is not trivial. Devices may have different capabilities and behaviors, which require both active measurements and passive observation to quantify. Furthermore, the IoT devices that are connected to the public Internet may vary from those connected inside home networks, requiring both an external and internal vantage point to draw measurements from. In this thesis, we demonstrate how IoT measurements drawn from a single vantage point or mesaurement technique lead to a biased view of the network services in the IoT ecosystem. To do this, we conduct several real-world IoT measurements, drawn from both inside and outside home networks using active and passive monitoring.

First, we leverage active scanning and passive observation in understanding the Mirai botnet—chiefly, we report on the devices it infected, the command and control infrastructure behind the botnet, and how the malware evolved over time. We then conduct active measurements from inside 16M home networks spanning 83M devices from 11 geographic regions to survey the IoT devices installed around the world. We demonstrate how these measurements can uncover the device types that are most at risk and the vendors who manufacture the weakest devices. We compare our measurements with passive external observation by detecting compromised scanning behavior from smart homes. We find that while passive external observation can drive insight about compromised networks, it offers little by way of concrete device attribution. We next compare our results from active external scanning with active internal scanning and show how relying solely on external scanning for IoT measurements under-reports security important IoT protocols, potentially skewing the services investigated by the security community. Finally, we conduct passive measurements of 275 smart home networks to investigate IoT behavior. We find that IoT device behavior varies by type and devices regularly communicate over a myriad of bespoke ports, in many cases to speak standard protocols (e.g., HTTP). Finally, we observe that devices regularly offer active services (e.g., Telnet, rpcbind) that are rarely, if ever, used in actual communication, demonstrating the need for both active and passive measurements to properly compare device capabilities and behaviors.

Our results highlight the need for a confluence of measurement perspectives to comprehensively understand IoT ecosystem. We conclude with recommendations for future measurements of IoT devices as well as directions for the systems and security community informed by our work.

To Amma, Appa, and Prakash.

ACKNOWLEDGMENTS

First and foremost, I want to thank my advisor, Professor Michael Bailey, for his support and guidance. I could write a sizable tome on how much I have learned both as a researcher and as a person from you, but I will spare the reader the gory details here. What I will say is this: you taught me that doing good work is important, but not as important as being kind. I will take that with me throughout every aspect of my life.

I would also like to thank my committee members—Professor Nikita Borisov, Assistant Professor Adam Bates, Assistant Professor Gang Wang, and Assistant Professor Zakir Durumeric—for their time and advice. I would especially like to thank Zakir, who has relentlessly encouraged me to "never stop digging" through the years we have known each other.

As anyone who has conducted measurement research knows, this thesis is the product of far more than my efforts alone. I must thank my labmates for everything from our spirited discussions to our quarantined pictionary games: Zane Ma, Simon Kim, Paul Murley, Joshua Reynolds, Suyup Kim and Yi Zhou—you all are rock stars. This thesis would also not be possible without Joshua Mason, our Network Security Research Group (NSRG) research scientist, who constantly reminds me that great researchers can be funny too.

And finally, to my very long list of collaborators over the years: Ariana Mirian, Alex Halderman, Manos Antonakakis, Tim April, Matt Bernhard, Elie Bursztein, Jaime Cochran, Luca Invernizzi, Michalis Kallitsis, Chaz Lever, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, Yi Zhou, Zhengping Wang, Matthew Hyder, Joey Dickinson, Gabrielle Beck, David Adrian, Dave Tian, Nolen Scaife, Adam Bates, Kevin Butler, Surya Bakshi, Andrew Miller, Riccardo Paccagnella, Eric Hennenfent, Kelly Shen, Benton Case, Deepali Garg, Galina Alperovich, Dmitry Kuznetsov, Rajarshi Gupta, Rohan Subramanian, Meishan Wu, Martin Shelton, Emily Stark: thank you for your ideas, for your support, and especially to the industry folk—for your data.

Last but certainly not least, I would not be here without my family. My parents, Sudha Kumar and Kumar Naryanan, have stood by me whenever I tell them I am changing my career path. That may still happen, but at least now I am done with school. To my brother, Prakash Kumar, who has unwaveringly supported me through everything: we will have to celebrate with a round of Smash Bros or Pokemon Showdown. I could also not have done this without the countless friends who have kept me smiling and laughing over the years, and to those who understood when I told them I was too busy to hang out. Well, I'm done now. Let's hang out?

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

The Internet has grown tremendously in the last decade. The number of Internet connected devices has greatly increased, from 2 billion devices in 2006 to an estimated 200 billion devices today [1]. Internet speeds have significantly improved [2], enabling faster and more reliable communication worldwide. Advancements in cloud-computing have granted developers easy access to high-end computation, storage, and communication [3]. Coupled with Internet growth are advancements in our computing devices themselves. Devices have shrunk in size and increased in capability; flagship mobile devices today offer faster computation, more storage, and more memory than high-end laptops from just five years prior at a fraction of the cost. Ubiquitous sensing enables innovation in a variety of cyber-physical systems.

It comes as no surprise that the combination of these advancements would eventually find their way into consumer products. Aptly named "The Internet of Things (IoT)"—these devices combine network connectivity, cheap hardware, and control into physical systems to provide users with new ways to interface with the world. Examples of these systems can now be found almost everywhere. Smart TVs and streaming devices, like the Roku TV and Google Chromecast, enable us easy access to high quality programming without the need for cable television [4, 5]. Voice assistants, such as the Amazon Echo and Google Home, provide us access to gaming, banking services, and even fully fledged home automation using only our voices [6]. Smart thermostats and cameras allow full control over the comfort, security, and safety of our homes. These are just a handful of examples. IoT devices have permeated our daily lives and shifted the ways in which we interact with computing. Outside of sheer innovation, these devices have also brought large economic development: the IoT industry is estimated to add $11 trillion dollars to the U.S. economy by 2025 [7].

In spite of this growth, there is little work done in measuring the network properties and behaviors of IoT devices. Taken alone, these measurements can inform us about the network services that IoT devices offer and subsequently, how devices use those services in practice, driving our understanding of the types of IoT devices deployed in the world and the vendors that manufacture them. Beyond this, they can also inform other researchers, like those in the systems and security communities, about how devices behave in practice to inform future security systems and guide future research.

Properly measuring the IoT ecosystem, however, is not trivial. Typically, researchers turn to active measurements of the public IPv4 space to understand the distributions and security posture of Internet connected devices. Tools like ZMap [8], Censys [9], Shodan [10], and Masscan [11] have transformed our access to these kinds of data, with full IPv4 scans available almost daily and freely to researchers. However, active scanning measurements are not without limitations. Although active

scanning enumerates the network services available on devices in a network, it cannot provide insight into how frequently those services are used in practice, or which devices communicate via those services.

Conversely, passive measurements—analogous to that of a network tap or a large darknet—can help researchers answer questions beyond simply enumerating device capabilities. Instead, researchers can focus on the behavior of devices: answering questions about the network layer services that devices use to frequently communicate and why. Research in passive measurement has driven insight into many aspects of device behavior, from inferring DoS activity [12] to checking user security behaviors [13]. Unfortunately, passive measurements taken in isolation may miss all capabilities of devices in a network. For example, even though a device communicates solely on port `80/HTTP`, it may also offer other services like `23/Telnet` or `22/SSH` that could place it at a higher security risk.

Of equal importance to varying measurement techniques (e.g., active versus passive) are drawing measurements from representative *vantage points*. In the context of IoT, measurements drawn only from the public Internet are limited, as many IoT devices are hidden behind network address translators (NATs) to internal networks. For example, it is unlikely to find devices expressly designed for home automation (e.g., smart lightbulbs) connected directly to the public Internet. To capture the properties of these devices, we must also draw measurements from *inside* home networks. However, measurements of the inside of networks are challenging to achieve. Collecting such measurements requires the voluntary participation of users as well as soliciting enough participants to measure these properties at scale. The value of these measurements, however, is clear: local network traffic often contains detailed information that is unavailable from application layer scanning alone (e.g., MAC addresses, DHCP leases, and UPnP discovery probes), offering a closer perspective to IoT device capabilities and behavior.

Thesis Statement: Understanding the network properties of IoT devices requires a combination of active and passive monitoring from both external and internal vantage points to the devices' local networks. A failure to consider all measurement perspectives leads to biased view of the network services in the IoT ecosystem.

In this thesis, we present measurements of the IoT ecosystem using active and passive techniques drawn from both inside and outside home networks. In the process, we compare each combination of vantage point and technique and demonstrate how relying on a single combination of vantage point and technique leads to differences in network services offered by devices. We argue that relying on a single vantage point and technique obscures the complex interplay between devices, vendors, and device behavior, and that only a confluence of measurement perspectives can adequately capture the networking behavior of the ecosystem as a whole. Our results have direct implications for future measurements of the IoT ecosystem, but can also inform security researchers and practitioners

about the fractured mix of architectures, devices, vendors, and behaviors that lead to a weakened IoT ecosystem.

We will leverage the described techniques and vantage points to conduct our IoT measurements. Chapter 2 shows how active probing of the public IPv4 space can be used to identify weak IoT devices and vendors, while Chapter 4 describes its shortcomings. Chapter 3 demonstrates how active probing inside networks can drive insight into the vendors that frequently produce insecure IoT devices. In contrast, Chapter 5 shows how passive observation both internally and externally can augment active scanning to provide deeper insight into network behaviors of IoT devices.

1.1 SUMMARY OF RESULTS

This thesis contains three separate measurements of the home IoT ecosystem from varied vantage points and techniques and highlights the benefits each measurement has on understanding the IoT ecosystem.

1.1.1 Measuring the Impact of Mirai

The Mirai botnet, composed primarily of embedded and IoT devices, took the Internet by storm in late 2016 when it overwhelmed several high-profile targets with massive distributed denial-of-service (DDoS) attacks. In Chapter 2, we measure the Mirai botnet from a myriad of measurement perspectives, including both through active scanning for device attribution and passive measurement through honeypots and a DNS tap of a large, U.S. ISP.

- Active probing can be used to identify 31.5% of the IoT devices scanning for the Mirai signature on the public Internet. We observe that vulnerable devices were primarily networked storage, routers, cameras, media devices, and printers. We show that many of the worlds top consumer electronic manufacturers—such Huawei, Dahua, ZTE, Cisco, ZyXEL, and MikroTik lacked protections to mitigate a threat like Mirai.

- We reverse engineer thousands of malware samples drawn from our passive, low-interaction honeypots, and track the evolution of the Mirai malware itself. We observe Mirai variants for 7 different architectures and show how the malware grew to support new infection vectors and targets over time.

- Through external, passive observation of DNS requests to Mirai's command and control (C2) infrastructure, we identify clusters of suspicious domains beyond the ones explicitly included

in malware samples, document evidence of competing botmaster control, and attribute C2 infrastructure to observed attacks.

1.1.2 Measuring Home IoT Networks with Active Scanning

In Chapter 3, we provide the first large-scale empirical analysis of IoT devices in real-world homes by leveraging data collected from user-initiated network scans of 83M devices in 16M households from 11 geographic regions. Our analysis is driven by active scans of devices *inside* home networks. We find the following:

- IoT adoption is widespread globally: on several continents, more than half of households already have at least one IoT device. In North America, this number is even higher—66.3% of homes have at least one IoT device.
- IoT devices in homes are manufactured by over 14.3K device vendors, of which 29% appear in only one home. Some device types are dominated by a handful of vendors (for example, Amazon and Google for voice assistants), while others are split across many vendors. Certain vendors are also only prevalent in certain regions, adding to the complexity of mitigating harm in the ecosystem.
- The security posture of devices varies heavily across regions, device types, and vendors. For example, although only 17% of TP-Link devices in North America have guessable passwords, nearly half do in Eastern Europe and Central Asia, highlighting the challenges in addressing these security flaws: there cannot be a one-size-fits-all approach.
- Passive, external observation through a large darknet can inform researchers of malicious scanning behaviors of home networks, however, are limited in their ability to attribute flaws to specific devices.
- IoT devices are staple, end-user products. However, for most homes, the types of devices adopted are not the ones actively discussed by the security community—illustrating a gap in where our work can be most effective.

1.1.3 Limitations of External Scanning

In Chapter 4, we combine our datasets from Chapter 2 and Chapter 3. We then outline the limitations of relying only on external scanning of the public IPv4 space to measure the IoT ecosystem, we did in Chapter 2 and as others have done in their research [14]. We find:

- Active, external scanning of the public IPv4 space under-reports the prevalence of several IoT critical protocols. For example, `8443/MQTT` is frequently used in IoT devices and appears on 1.5% of devices inside home networks, but on near 0% of IoT devices connected publicly. Relying only on external scanning to network insight into IoT devices skews our understanding of the IoT ecosystem.

- Although distributions of device types are different in aggregate between external scanning and internal scanning, we find that IoT device identification through HTTPS only matches closely the distribution of device types found inside home networks ($\rho = 0.97$). Even in this case, the external IoT device type population excludes several device type (like work appliances) which have a high prevalence inside local networks.

1.1.4 Measuring Home IoT Behavior with Passive Observation

To conclude this thesis, we measure the home IoT ecosystem via both passive observation and active observation *inside* homes. We instrument Princeton IoT Inspector [15] to collect passive traces of communication within homes and actively probe IoT devices inside them. We recruit 275 participants to measure IoT devices in their homes for a period of 3 weeks. We detail our results in Chapter 5. We find:

- Device behavior varies by device type and aligns with its function. For example, smart home IoT devices like lightbulbs and switches communicate with only a small number of devices on the local network (14%), while storage devices communicate with more than half of the devices on the network (52%).

- IoT devices communicate over a myriad of specialized ports, in many cases to speak standard protocols (e.g., HTTP). In investigating these ports, we find 44% are largely offered by a single manufacturer for a specific device type, which highlights a lack of standardization in networking behavior and presents complications for comprehensive active measurements.

- 73% of IoT devices have at least one active service (observed through active scanning), however, devices utilize only an average of 19% of the services they offer, highlighting that device capabilities through active probing are much wider than device behavior through passive observation alone.

- External, passive observation of IoT devices captures only a small fraction of the behaviors of IoT devices internally and skews heavily towards a handful of popular protocols: DNS, HTTP, and HTTPS, highlighting that external behavior does not match internal device behavior.

Taken together, this thesis highlights the need for a confluence of vantage points and techniques in measuring network properties of the IoT ecosystem. Beyond measurement, our results also inform the security and systems communities about future directions in improving IoT safety. Our results in Chapter 3 highlight a gap between the devices investigated by the security research community and those with the lowest security profiles, pointing to an area for security researchers to have immediate impact. Our results from Chapter 5 show how IoT devices receive control commands from many other devices besides smartphones (voice assistants, other IoT devices), which can inform how to build better isolation schemes for hardening local networks. Finally, we also show how device behavior varies by device type, which opens up possibilities for systems designers to improve existing IoT policy engines [16, 17]. We detail these lessons learned and future directions in Chapter 6.

1.2 THESIS ROADMAP

The remainder of this thesis is composed of five chapters. In Chapter 2, we conduct active and passive measurements of the Mirai botnet. In Chapter 3, we demonstrate how active measurements drawn from inside 16M help to uncover a fractured IoT ecosystem that spans device types, vendors, and world regions. In Chapter 4, we combine our datasets from the previous two chapters and highlight the limitations of external, active scanning. Chapter 5 demonstrates how passive measurements can be used to infer device behavior, and presents a comparison between active and passive measurements in real-world smart homes. In Chapter 6, we conclude with implications of this thesis for the measurement and security communities. We hope the results of this thesis will enable principled research in IoT measurement, and prove useful for the measurement and security communities as we continue to measure and improve the growing IoT ecosystem.

CHAPTER 2: MEASURING THE IMPACT OF MIRAI

2.1 INTRODUCTION

Starting in September 2016, a spree of massive distributed denial-of-service (DDoS) attacks temporarily crippled Krebs on Security [18], OVH [19], and Dyn [20]. The initial attack on Krebs exceeded 600 Gbps in volume [18]—among the largest on record. Remarkably, this overwhelming traffic was sourced from hundreds of thousands of some of the Internet's least powerful hosts—Internet of Things (IoT) devices—under the control of a new botnet named Mirai.

While other IoT botnets such as BASHLITE [21] and Carna [22] preceded Mirai, the latter was the first to emerge as a high-profile DDoS threat. What explains Mirai's sudden rise and massive scale? A combination of factors—efficient spreading based on Internet-wide scanning, rampant use of insecure default passwords in IoT products, and the insight that keeping the botnet's behavior simple would allow it to infect many heterogeneous devices—all played a role. Indeed, Mirai has spawned many variants that follow the same infection strategy, leading to speculation that "IoT botnets are the new normal of DDoS attacks" [23].

In this chapter, we investigate the precipitous rise of Mirai and the fragile IoT ecosystem it has subverted. We draw from a diverse set of measurement vantage points and techniques, including active Internet-wide banner scans, IoT honeypots, C2 milkers, and large passive observation from a DNS tap of a large, U.S. ISP. We focus on three aspects of Mirai's rise: the devices that comprised the botnet, the evolution of Mirai, and which malware variants were used in conducting attacks. These unique datasets enable us to conduct the first comprehensive measurements of the Mirai botnet and posit technical and non-technical defenses that may curb future attacks.

We track the outbreak of Mirai and find the botnet infected nearly 65,000 IoT devices in its first 20 hours before reaching a steady state population of 200,000–300,000 infections. These bots fell into a narrow band of geographic regions and autonomous systems, with Brazil, Columbia, and Vietnam disproportionately accounting for 41.5% of infections. We confirm that Mirai targeted a variety of IoT and embedded devices ranging from DVRs, IP cameras, routers, and printers, but find Mirai's ultimate device composition was strongly influenced by the market shares and design decisions of a handful of consumer electronics manufacturers.

By statically analyzing over 1,000 malware samples, we document the evolution of Mirai into dozens of variants propagated by multiple, competing botnet operators. These variants attempted to improve Mirai's detection avoidance techniques, add new IoT device targets, and introduce additional DNS resilience. We find that Mirai harnessed its evolving capabilities to launch over 15,000 attacks against not only high-profile targets (e.g., Krebs on Security, OVH, and Dyn), but

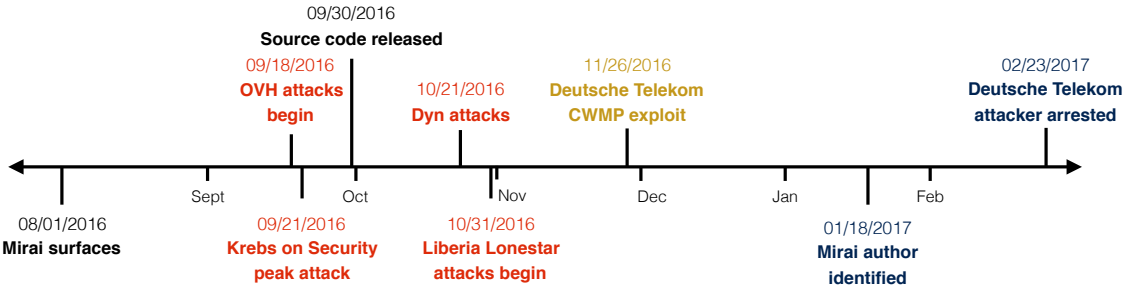


Figure 2.1: **Mirai Timeline**—Major attacks (red), exploits (yellow), and events (black) related to the Mirai botnet.

also numerous game servers, telecoms, anti-DDoS providers, and other seemingly unrelated sites. While DDoS was Mirai's flavor of abuse, future strains of IoT malware could leverage access to compromised routers for ad fraud, cameras for extortion, network attached storage for bitcoin mining, or any number of applications. Mirai's reach extended across borders and legal jurisdictions, and it infected devices with little infrastructure to effectively apply security patches. This made defending against it a daunting task.

Finally, we look beyond Mirai to explore the security posture of the IoT landscape. We find that the absence of security best practices—established in response to desktop worms and malware over the last two decades—has created an IoT substrate ripe for exploitation. However, this space also presents unique, nuanced challenges in the realm of automatic updates, end-of-life, and consumer notifications. Without improved defenses, IoT-based attacks are likely to remain a potent adversarial technique as botnet variants continue to evolve and discover new niches to infect. In light of this, Mirai seems aptly named—it is Japanese for "the future."

This chapter includes a subset of work that appeared at the USENIX Security Symposium in 2017.

2.2 THE MIRAI BOTNET

Mirai is a worm-like family of malware that infected IoT devices and corralled them into a DDoS botnet. We provide a brief timeline of Mirai's emergence and discuss its structure and propagation.

Timeline of events Reports of Mirai appeared as early as August 31, 2016 [24], though it was not until mid-September, 2016 that Mirai grabbed headlines with massive DDoS attacks targeting Krebs on Security [18] and OVH [25] (Figure 2.1). Several additional high-profile attacks later targeted DNS provider Dyn [20] and Lonestar Cell, a Liberian telecom [26]. In early 2017, the actors surrounding Mirai came to light as the Mirai author was identified [27]. Throughout our

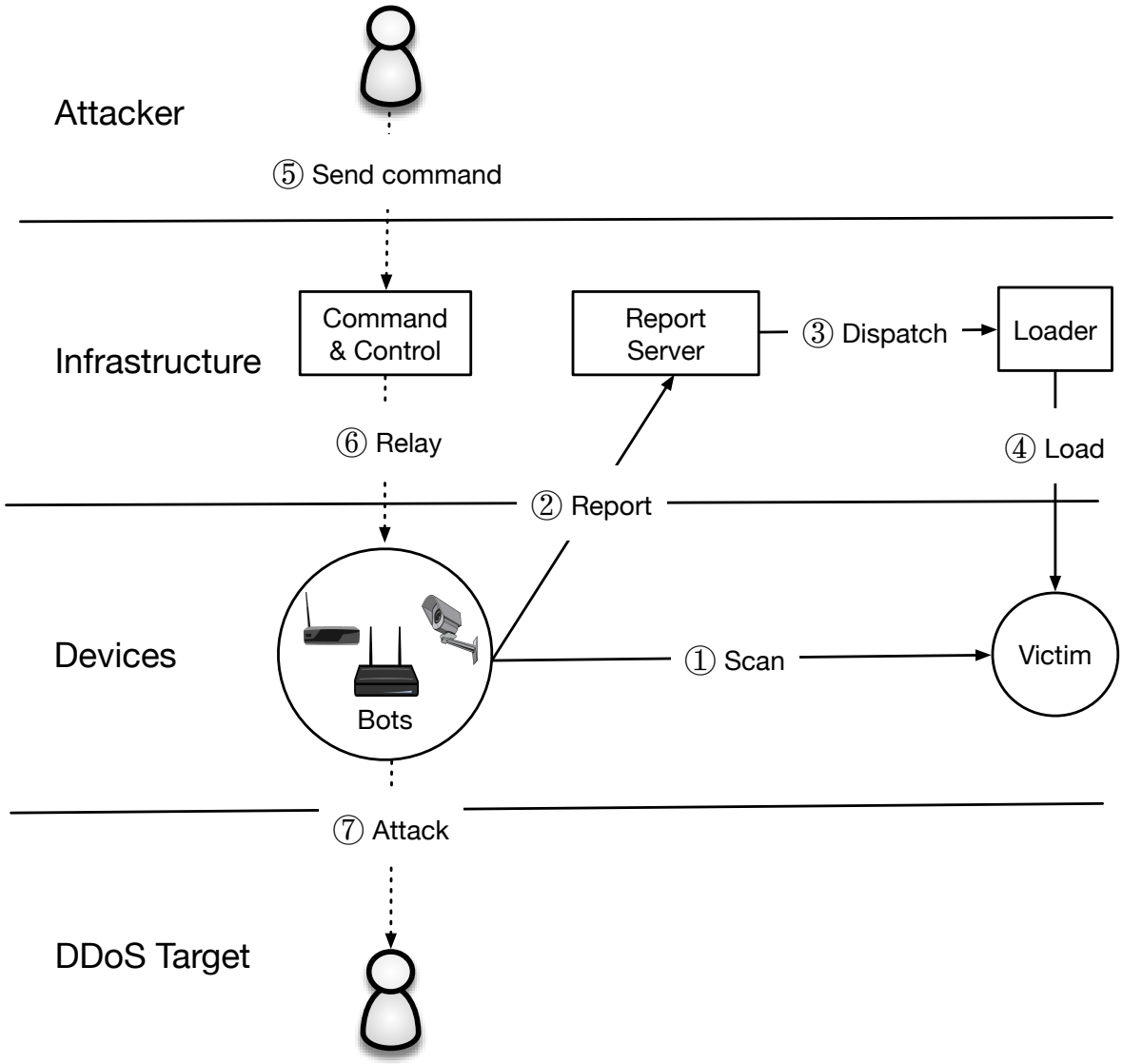


Figure 2.2: **Mirai Operation**—Mirai bots scan the IPv4 address space for devices that run telnet or SSH, and attempt to log in using a hardcoded dictionary of IoT credentials. Once successful, the bot sends the victim IP address and associated credentials to a report server, which asynchronously triggers a loader to infect the device. Infected hosts scan for additional victims and accept DDoS commands from a command and control (C2) server.

study, we corroborate our measurement findings with these media reports and expand on the public information surrounding Mirai.

Another significant event in this timeline is the public release of Mirai's source code on hackforums.net [28]. We rely on this code to develop our measurement methodology (Section 3.2). Furthermore, as we detail later (Section 2.5), this source code release led to the proliferation of Mirai variants with competing operators. One notable variant added support for a router exploit

through CPE WAN Management Protocol (CWMP), an HTTP-based protocol that enables auto-configuration and remote management of home routers, modems, and other customer-premises equipment (CPE) [29]. This exploit led to an outage at Deutsche Telekom late November 2016 [30], with the suspected attacker later arrested in February 2017 [31]. In this work, we track Mirai's variants and examine how they influenced Mirai's propagation.

Botnet structure & propagation We provide a summary of Mirai's operation in Figure 2.2, as gleaned from the released source code. Mirai spread by first entering a *rapid scanning* phase (①) where it asynchronously and "statelessly" sent TCP SYN probes to pseudorandom IPv4 addresses, excluding those in a hard-coded IP blacklist, on Telnet TCP ports 23 and 2323 (hereafter denoted TCP/23 and TCP/2323). If Mirai identifies a potential victim, it entered into a *brute-force login* phase in which it attempted to establish a Telnet connection using 10 username and password pairs selected randomly from a pre-configured list of 62 credentials. At the first successful login, Mirai sent the victim IP and associated credentials to a hardcoded *report server* (②).

A separate *loader program* (③) asynchronously infected these vulnerable devices by logging in, determining the underlying system environment, and finally, downloading and executing architecture-specific malware (④). After a successful infection, Mirai attempted to conceal its presence by deleting the downloaded binary and obfuscating its process name in a pseudorandom alphanumeric string. As a consequence, Mirai infections did not persist across system reboots. In order to fortify itself, the malware additionally killed other processes bound to TCP/22 or TCP/23, as well as processes associated with competing infections, including other Mirai variants, `.anime` [32], and Qbot [33]. At this point, the bot listened for attack commands from the command and control server (C2) while simultaneously scanning for new victims.

Malware phylogeny While not directly related to our study, the Mirai family represents an evolution of BASHLITE (otherwise known as LizardStresser, Torlus, Gafgyt), a DDoS malware family that infected Linux devices by brute forcing default credentials [21]. BASHLITE relied on six generic usernames and 14 generic passwords, while the released Mirai code used a dictionary of 62 username/password pairs that largely subsumed BASHLITE's set and added credentials specific to consumer routers and IoT devices. In contrast to BASHLITE, Mirai additionally employed a fast, stateless scanning module that allowed it to more efficiently identify vulnerable devices.

2.3 METHODOLOGY

Our study of Mirai leverages a variety of network vantage points: a large, passive network telescope, Internet-wide scanning, active Telnet honeypots, logs of C2 attack commands, passive

Role	Data Source	Collection Site	Collection Period	Data Volume
Growth and size	Network telescope	Merit Network, Inc.	07/18/2016–02/28/2017	370B packets, avg. 269K IPs/min
Device composition	Active scanning	Censys	07/19/2016–02/28/2017	136 IPv4 scans, 5 protocols
Ownership & evolution	Telnet honeypots	AWS EC2	11/02/2016–02/28/2017	141 binaries
	Telnet honeypots	Akamai	11/10/2016–02/13/2017	293 binaries
	Malware repository	VirusTotal	05/24/2016–01/30/2017	594 binaries
	DNS—active	Georgia Tech	08/01/2016–02/28/2017	290M RRs/day
	DNS—passive	Large U.S. ISP	08/01/2016–02/28/2017	209M RRs/day
Attack characterization	C2 milkers	Akamai	09/27/2016–02/28/2017	64.0K attack commands
	DDoS IP addresses	Akamai	09/21/2016	12.3K IP addresses
	DDoS IP addresses	Google Shield	09/25/2016	158.8K IP addresses
	DDoS IP addresses	Dyn	10/21/2016	107.5K IP addresses

Table 2.1: **Data Sources**—We utilized a multitude of data perspectives to empirically analyze the Mirai botnet.

DNS traffic, and logs from DDoS attack targets. In this section, we discuss our data sources and the role they play in our analysis. We provide a high-level summary in Table 2.1.

2.3.1 Active Scanning

While Mirai is widely considered an IoT botnet, there has been little comprehensive analysis of infected devices over the botnet's entire lifetime. In order to determine the manufacturer and model of devices infected with Mirai, we leveraged Censys [9], which actively scans the IPv4 space and aggregates application layer data about hosts on the Internet. We focused our analysis on scans of HTTPS, FTP, SSH, Telnet, and CWMP between July 19, 2016 and February 28, 2017.

A number of challenges make accurate device labeling difficult. First, Mirai immediately disables common outward facing services (e.g., HTTP) upon infection, which prevents infected devices from being scanned. Second, Censys scans often take more than 24 hours to complete, during which devices may churn to new IP addresses. Finally, Censys executes scans for different protocols on different days, making it difficult to increase label specificity by combining banners from multiple services. We navigated these constraints by restricting our analysis to banners that were collected within twenty minutes of scanning activity (the time period after which we expire a scan). This small window mitigates the risk of erroneously associating the banner data of uninfected devices

Protocol	Banners	Devices Identified	
HTTPS	342,015	271,471	(79.4%)
FTP	318,688	144,322	(45.1%)
Telnet	472,725	103,924	(22.0%)
CWMP	505,977	35,163	(7.0%)
SSH	148,640	8,107	(5.5%)
Total	1,788,045	587,743	(31.5%)

Table 2.2: **Devices Identified**—We identified device type, model, and/or vendor for 31.5% of active scan banners. Protocol banners varied drastically in device identifiability, with HTTPS certificates being most descriptive, and SSH prompts being the least.

with Mirai infections due to DHCP churn.

Post-filtering, our dataset included 1.8 million banners associated with 1.2 million Mirai-infected IP addresses (Table 2.2). We had the most samples for CWMP, and the least for SSH. We caution that devices with open services that are not closed by Mirai (e.g., HTTPS and FTP) can appear repeatedly in Censys banner scans during our measurement window (due to churn) and thus lead to over counting when compared across protocols. As such, we intentionally explored protocols in isolation from one another and limited ourselves to measurements that only consider relative proportions rather than absolute counts of infected hosts.

Finally, we processed each infected device's banner to identify the device manufacturer and model. We first applied the set of regular expressions used by Nmap service probes to fingerprint devices [34]. Nmap successfully handled 98% of SSH banners and 81% of FTP banners, but matches only 7.8% of the Telnet banners. In order to increase our coverage and also accommodate HTTPS and CWMP (which Nmap lacks probes for), we constructed our own regular expressions to map banners to device manufacturers and models. Unfortunately, we found that in many cases, there was not enough data to identify a model and manufacturer from FTP, Telnet, CWMP, and SSH banners and that Nmap fingerprints only provide generic descriptions. In total, we identified device type and/or model and manufacturer for 31.5% of banners (Table 2.2).

2.3.2 Telnet Honeypots

To track the evolution of Mirai's capabilities, we collected binaries installed on a set of Telnet honeypots that masqueraded as vulnerable IoT devices. Mechanically, we presented a BusyBox shell [35] and IoT-consistent device banner. Our honeypots logged all incoming Telnet traffic and downloaded any binaries that attackers attempted to install on the host via `wget` or `tftp` (the methods of infection found in Mirai's original source). In order to avoid collateral damage, we blocked all other outgoing requests (e.g., scanning and DoS traffic).

We logged 80K connection attempts from 54K IP addresses between November 2, 2016 and February 28, 2017, collecting a total 151 unique binaries. We filtered out executables unrelated to Mirai based on a YARA signature that matched any of the strings from the original source code release, leaving us with 141 Mirai binaries. We supplemented this data with 293 binaries observed by honeypots operated by Akamai, which served a similar purpose to ours, but were hosted on a different public cloud provider. As a final source of samples, we included 594 unique binaries from VirusTotal [36] that we scanned for using the YARA rules mentioned above. In total, we collected 1,028 unique Mirai samples.

We analyzed the binaries for the three most common architectures—MIPS 32-bit, ARM 32-bit, and x86 32-bit—which account for 74% of our samples. We extracted the set of logins and passwords, IP blacklists, and C2 domains from these binaries, identifying 67 C2 domains and 48 distinct username/password dictionaries (containing a total 371 unique passwords).

2.3.3 Passive & Active DNS

Following the public release of Mirai's source code, competing Mirai botnet variants came into operation. We disambiguated ownership and estimate the relative size of each Mirai strain by exploring passive and active DNS data for the 67 C2 domains that we found by reverse engineering Mirai binaries. We also leveraged our DNS data to map the IP addresses present in attack commands to victim domain names.

From a large U.S. ISP, we obtained passive DNS data consisting of DNS queries generated by the ISP's clients and their corresponding responses. More specifically, we collected approximately 209 million resource records (RRs)—queried domain name, and associated RDATA—and their lookup volumes aggregated on a daily basis. For our active DNS dataset, we obtained 290 million RRs per day from Thales, an active DNS monitoring system [37]. Both datasets cover the period of August 1, 2016 to February 28, 2017.

Using both passive and active DNS datasets, we performed DNS *expansion* to identify shared DNS infrastructure by linking related historic domain names (RHDN) and related historic IPs (RHIPs) [38]. This procedure began with the seed set of C2 domains and IPs extracted during reverse engineering of our honeypotted binaries. For a given seed `foo.com`, we identified the IP addresses that `foo.com` previously resolved to and added them to a growing set of domains and IPs. We additionally performed the reverse analysis, starting from an IP and finding any domain names that concurrently resolved it. Thus, even from a single domain name, we iteratively expanded the set of related domain names and IP addresses to construct a graph reflecting the shared infrastructure used by Mirai variants. In total, we identified 33 unique DNS clusters that we explore in detail in Section 2.5.

2.3.4 Attack Commands

To track the DDoS attack commands issued by Mirai operators, Akamai ran a "milker" from September 27, 2016–February 28, 2017 that connected to the C2 servers found in the binaries uploaded to their honeypots. The service simulated a Mirai-infected device and communicated with the C2 server using a custom bot-to-C2 protocol, which was reverse engineered from malware samples prior to source code release. In total, Akamai observed 64K attack commands issued by 484 unique C2 servers (by IP address). We note that a naive analysis of attack commands overestimates the volume of attacks and targets: individual C2 servers often repeat the same attack command in rapid succession, and multiple distinct C2 servers frequently issued the same command. To account for this, we heuristically grouped attack commands along two dimensions: by shared C2 infrastructure and by temporal similarity. We collapsed matching commands (i.e., tuples of attack type, duration, targets, and command options) that occur within 90 seconds of each other, which yielded 15,194 attacks from 146 unique IP clusters. Our attack command coverage includes the Dyn attack [20] and Liberia attacks [26]. We did not observe attack commands for Krebs on Security and OVH, which occurred prior to the milker's operation.

2.3.5 DDoS Attack Traces

Our final data source consists of network traces and aggregate statistics from Akamai and Google Shield (the providers for Krebs on Security) and Dyn. These attacks cover two distinct periods in Mirai's evolution. We used this data to corroborate the IP addresses observed in attacks versus those found scanning our passive network telescope, as well as to understand the volume of traffic generated by Mirai. From Akamai, we obtained an aggregate history of all DDoS attacks targeting Krebs on Security from 2012–2016, as well as a small sample of 12.3K IPs related to a Mirai attack on September 21, 2016. For Google Shield, we shared a list of IP addresses observed by our network telescope and in turn received aggregate statistics on what fraction matched any of 158.8K IP addresses involved in a 1-minute Mirai HTTP-flood attack on September 25, 2016. Finally, Dyn provided us with a set of 107.5K IP addresses associated with a Mirai attack on October 21, 2016.

2.4 DEVICES AND GEOGRAPHY

In this section, we detail the botnet's composition, including the devices it ultimately infected and where in the world those devices were.

2.4.1 Device Composition

While cursory evidence suggested that Mirai targets IoT devices—Mirai's dictionary of default usernames and passwords included routers, DVRs, and cameras [39], and its source compiled to multiple embedded hardware configurations—we provide an in-depth analysis of both the intended device targets and successful infections.

To understand the types of devices that Mirai targeted, we analyzed the credentials hardcoded into the binaries we collected. We observed a total 371 unique passwords, and through manual inspection, we identified 84 devices and/or vendors associated with these passwords. Many passwords were too generic to tie to a specific device (i.e., "password" applies to devices from a large number of manufacturers), while others only provided information about underlying software (e.g., "postgres") and not an associated device. The devices we identified were primarily network-attached storage appliances, home routers, cameras, DVRs, printers, and TV receivers made by dozens of different manufacturers (Table 2.3).

Mirai's intended targets do not necessarily reflect the breakdown of infected devices in the wild. We leveraged the device banners collected by Censys to determine the models and manufacturers of infected devices. Our results across all five protocols indicate that security cameras, DVRs, and consumer routers represent the majority of Mirai infections (Table 2.4). The manufacturers responsible for the most infected devices we could identify are: Dahua, Huawei, ZTE, Cisco, ZyXEL, and MikroTik (Table 2.5). We note that these results deviate from initial media reports, which stated that Mirai was predominantly composed of DVRs and cameras [40, 41, 42].

Our data indicates that some of the world's top manufacturers of consumer electronics lacked sufficient security practices to mitigate threats like Mirai, and these manufacturers will play a key part in ameliorating vulnerability. Unfortunately, as discussed in the previous section, the menagerie of devices spanned both countries and legal jurisdictions, exacerbating the challenge of coordinating technical fixes and promulgating new policy to safeguard consumers in the future.

Password	Device Type	Password	Device Type	Password	Device Type
123456	ACTi IP Camera	klv1234	HiSilicon IP Camera	1111	Xerox Printer
anko	ANKO Products DVR	jvbzd	HiSilicon IP Camera	Zte521	ZTE Router
pass	Axis IP Camera	admin	IPX-DDK Network Camera	1234	Unknown
888888	Dahua DVR	system	IQinVision Cameras	12345	Unknown
666666	Dahua DVR	meinsm	Mobotix Network Camera	admin1234	Unknown
vizxv	Dahua IP Camera	54321	Packet8 VOIP Phone	default	Unknown
7ujMko0vizxv	Dahua IP Camera	00000000	Panasonic Printer	fucker	Unknown
7ujMko0admin	Dahua IP Camera	realtek	RealTek Routers	guest	Unknown
666666	Dahua IP Camera	1111111	Samsung IP Camera	password	Unknown
dreambox	Dreambox TV Receiver	xmhdipc	Shenzhen Anran Camera	root	Unknown
juantech	Guangzhou Juan Optical	smcadmin	SMC Routers	service	Unknown
xc3511	H.264 Chinese DVR	ikwb	Toshiba Network Camera	support	Unknown
OxhlwSG8	HiSilicon IP Camera	ubnt	Ubiquiti AirOS Router	tech	Unknown
cat1029	HiSilicon IP Camera	supervisor	VideoIQ	user	Unknown
hi3518	HiSilicon IP Camera	<none>	Vivotek IP Camera	zlxx.	Unknown
klv123	HiSilicon IP Camera				

Table 2.3: **Default Passwords**—The 09/30/2016 Mirai source release included 46 unique passwords, some of which were traceable to a device vendor and device type. Mirai primarily targeted IP cameras, DVRs, and consumer routers.

CWMP (28.30%)		Telnet (26.44%)		HTTPS (19.13%)		FTP (17.82%)		SSH (8.31%)	
Router	4.7%	Router	17.4%	Camera/DVR	36.8%	Router	49.5%	Router	4.0%
		Camera/DVR	9.4%	Router	6.3%	Storage	1.0%	Storage	0.2%
				Storage	0.2%	Camera/DVR	0.4%	Firewall	0.2%
				Firewall	0.1%	Media	0.1%	Security	0.1%
Other	0.0%	Other	0.1%	Other	0.2%	Other	0.0%	Other	0.0%
Unknown	95.3%	Unknown	73.1%	Unknown	56.4%	Unknown	49.0%	Unknown	95.6%

Table 2.4: **Top Mirai Device Types**—We list the top types of infected devices labeled by active scanning, as a fraction of Mirai banners found in Censys. Our data suggests that consumer routers, cameras, and DVRs were the most prevalent identifiable devices.

CWMP (28.30%)		Telnet (26.44%)		HTTPS (19.13%)		FTP (17.82%)		SSH (8.31%)	
Huawei	3.6%	Dahua	9.1%	Dahua	36.4%	D-Link	37.9%	MikroTik	3.4%
ZTE	1.0%	ZTE	6.7%	MultiTech	26.8%	MikroTik	2.5%		
		Phicomm	1.2%	ZTE	4.3%	ipTIME	1.3%		
				ZyXEL	2.9%				
				Huawei	1.6%				
Other	2.3%	Other	3.3%	Other	7.3%	Other	3.8%	Other	1.8%
Unknown	93.1%	Unknown	79.6%	Unknown	20.6%	Unknown	54.8%	Unknown	94.8%

Table 2.5: **Top Mirai Device Vendors**—We list the top vendors of infected Mirai devices labeled by active scanning, as a fraction of Mirai banners found by Censys. The top vendors across all protocols were primarily camera, router, and embedded device manufacturers.

ID	Max Lookup Vol.	Notes
6	61,440	Attacked Dyn, other gaming related attacks
1	58,335	The original botnet. Attacked Krebs on Security, OVH
2	36,378	Attacked Lonestar Cell. Scans TCP/7547 and TCP/5555, removes DoD from blacklist, adds DGA
13	9,657	—
7	9,467	Scans TCP/7547

Table 2.6: **Cluster Size Estimate and Characteristics**—We highlight the top five clusters by max single-day lookup volume within a large U.S. ISP, which provides an indicator of their relative size. Each cluster is additionally labeled with observed evolutionary patterns and associated attacks.

2.5 OWNERSHIP AND EVOLUTION

After the public release of Mirai's source code in late September 2016, multiple competing variants of the botnet emerged. We analyze the C2 infrastructure behind Mirai in order to uncover the relationships between strains, their relative sizes, and the evolution of their capabilities.

2.5.1 Ownership

In order to identify the structure of Mirai command and control servers, we turned to active and passive DNS data, which we used to cluster C2 IPs and domains based on shared network infrastructure. Seeding DNS expansion with the two IPs and 67 domains that we collected by reverse engineering Mirai binaries, we identified 33 independent C2 clusters that shared no infrastructure. These varied from a single host to the largest cluster, which contained 112 C2 domains and 92 IP addresses. We show the connectivity of the top six clusters by number of C2 domains in Figure 2.3. The lack of shared infrastructure between these clusters lends credence to the idea that there are multiple active bot operators during our study period.

While Figure 2.3 provides a rough sense of Mirai C2 complexity, it does not indicate the number of bots that each cluster controlled. To estimate botnet membership, we measured the DNS lookup volume per cluster. In Figure 2.4, we show the top clusters of domains based on the volume of DNS lookups at a large, name-redacted ISP. This single perspective is not comprehensive, but it allows us to observe the rise and fall of different botnets over time, and may provide a hint of their relative sizes. A prime example is cluster 1, which was the initial version of the Mirai botnet involved in the early, high-profile attacks on Krebs on Security and OVH. Although it dominated in lookup volume in late September and early October, it gave way to newer clusters, 2 and 6, in mid-October. We provide a list of the largest clusters by lookup and their unique characteristics in Table 2.6.

While we cannot conclusively link each of these clusters to distinct operators, we note that each

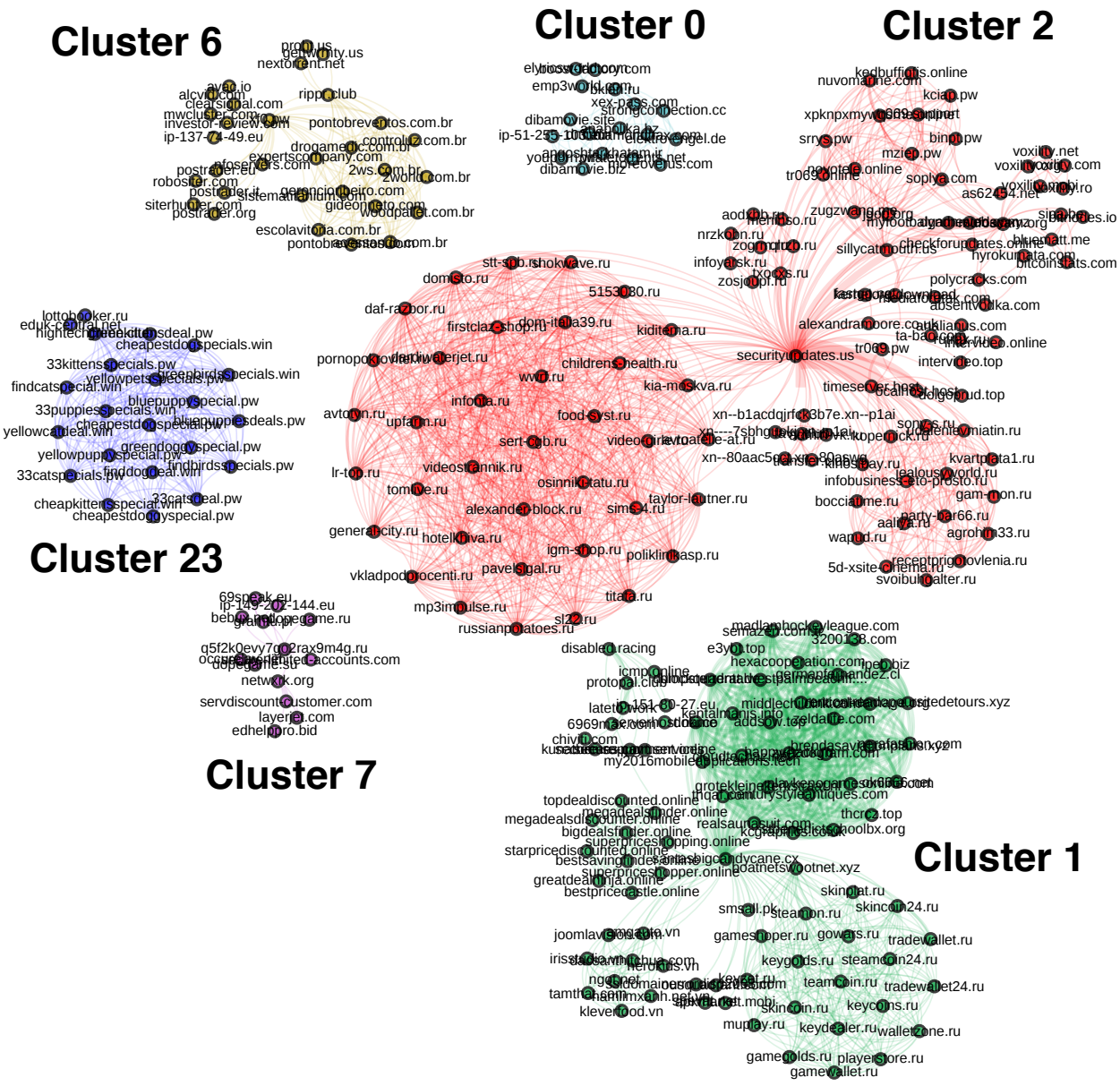


Figure 2.3: **C2 Domain Relationships**—We visualize related C2 infrastructure, depicting C2 domains as nodes and shared IPs as edges between two domains. The top six clusters by C2 domain count consisted of highly connected components, which represent agile, long-lived infrastructures in use by botmasters.

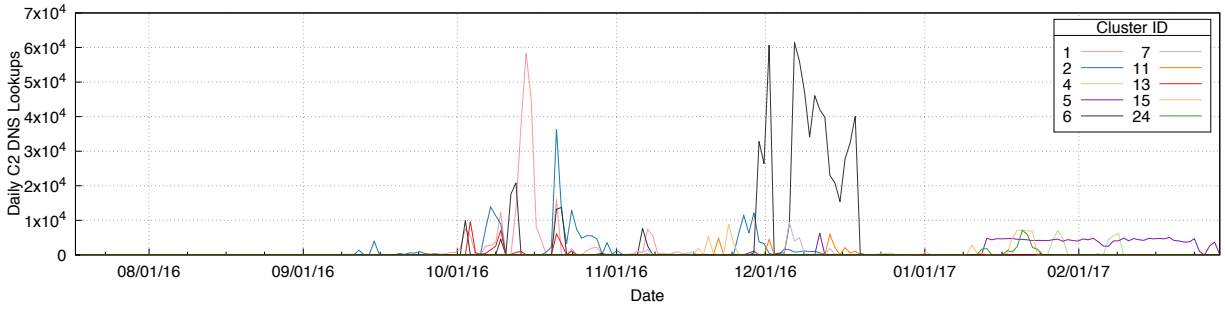


Figure 2.4: **C2 Cluster Lookup Volume**—The DNS lookup volume of C2 DNS clusters in a large U.S. ISP establishes the relative size of the botnet behind each cluster and chronicles its rise and fall. Note, for example, cluster 1 which represents the original botnet in use for the early high profile attacks on Krebs and OVH and the emergence of a myriad of clusters after the public source release.

cluster utilized independent DNS infrastructure and evolving malware, underscoring the challenge of defending against these attacks through bespoke mitigations. Our results also confirm the recent findings of Lever et al., who observed that the naming infrastructure used by malware is often active weeks prior to its operation [43]. In all cases, the first occurrence of DNS/IP lookup traffic for a cluster far preceded the date that the domains were used as C2 infrastructure for the botnet. For example, even though the peak lookup for cluster 2 occurred on October 21, 2016, the first lookup of a C2 domain in this cluster occurred on August 1, 2016 (Table 2.6). This also significantly predated the first binary collected for this cluster (October 24, 2016), and the first attacks issued by the cluster (October 26, 2016). These results suggest that careful analysis of DNS infrastructure can potentially guide preventative measures.

2.5.2 Evolution

Although the Mirai ecosystem exploded after the public source code release on September 30, 2016, this was not the botnet's first major evolutionary step. Between August 7, 2016 and September 30, 2016—when the source code was publicly released—24 unique Mirai binaries were uploaded to VirusTotal, which we used to explore the botnet's initial maturation. Several key developments occurred during this period. First, we saw the underlying C2 infrastructure upgrade from an IP-based C2 to a domain-based C2 in mid-September. Second, the malware began to delete its executing binary, as well as obfuscate its process ID, also in mid-September. We additionally saw a number of features added to make the malware more virulent, including the addition of more passwords to infect additional devices, the closing of infection ports TCP/23 and TCP/2323, and the aggressive killing of competitive malware in a sample collected on September 29, 2016.

After the public release, we observed the rapid emergence of new features, ranging from improved

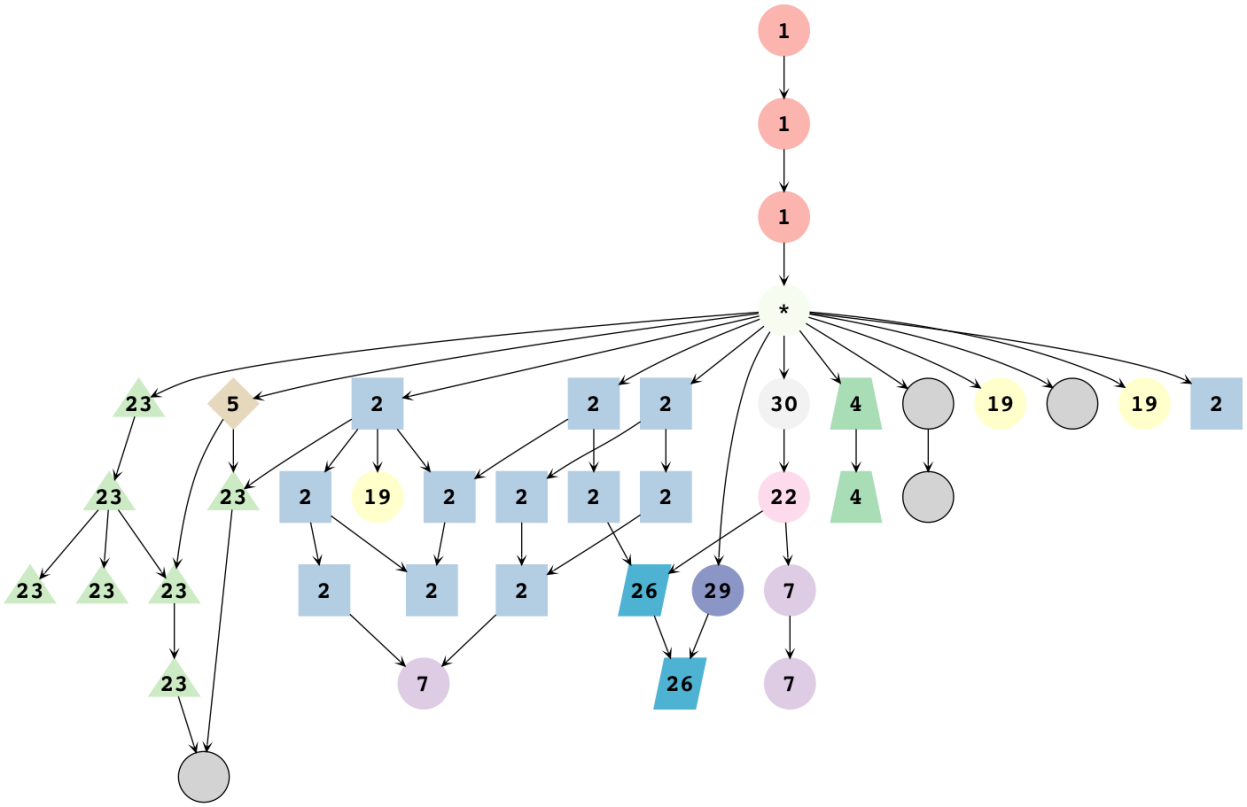


Figure 2.5: **Password Evolution**—The lineage of unique password dictionaries, labeled with their associated clusters, depicts many malware strains modifying the default credential list to target additional devices. The node marked (*) indicates the released source code password dictionary and serves as the foundation for the all divergent password variants

infection capabilities to hardened binaries that slow reverse engineering efforts. Between November 2, 2016 and February 28, 2017, we observed 48 new sets of usernames and passwords, as well as changes to the IP blacklist. We note that while many actors modified the set of credentials, they often did so in different ways (Figure 2.5). This is true for other features as well. In one example, a variant evolved to remove U.S. Department of Defense blocks from the initial scanning blacklist. The malware further evolved to use new infection mechanisms. Most notably, in late November 2016, Mirai variants began to scan for TCP/7547 and TCP/5555, two ports commonly associated with CWMP [29, 44]. Additionally, one malware strain began to using domain generation algorithms (DGA) in the place of a hardcoded C2 domain, though this feature was short lived. By November 2016, packed binaries had emerged.

Techniques to improve virulence and to aide in reliability were not simply limited to the client binaries. We found evidence of operators using DNS to avoid or attempt to evade detection as well. Recent work by Lever et al. demonstrated how attackers abuse the residual trust inherited by domains to perform many, seemingly unconnected types of abuse [45]. Mirai was no different from

other types of malware—we found evidence that at least 17% of Mirai domains abused residual trust. Specifically, these domains expired and were subsequently re-registered before they were used to facilitate connections between bots and C2 servers. This serves as a reminder that although Mirai is unique in many ways, it still shares much in common with the many threats that came before it.

By combining the malware we observed with our DNS data, we can also measure the evolution of the C2 clusters in Table 2.6. We note that cluster 2—the third largest by lookup volume—evolved to support many new features, such as scanning new ports TCP/7547 and TCP/5555, adding DGA, and modifying the source code blacklist to exclude Department of Defense (DoD) blocks. This is not to say, however, that evolution guaranteed success. Cluster 23, which can be seen clearly in Figure 2.5, evolved very rapidly, adding several new passwords over its active time. Despite this evolution, this cluster was 19th out of 33 clusters in terms of lookup volume over time and was unable to capture much of the vulnerable population. We also note that not all successful clusters evolved either; for example, cluster 6, which showed no evolutionary trend from its binaries, received the highest lookup volume of all the clusters.

2.6 MIRAI'S DDOS ATTACKS

The Mirai botnet and its variants conducted tens of thousands of DDoS attacks during our monitoring period. We explore the strategies behind these attacks, characterize their targets, and highlight case studies on high-profile targets Krebs on Security, Dyn, and Liberia's Lonestar Cell. We find that Mirai bore a resemblance to booter services (which enable customers to pay for DDoS attacks against desired targets), with some Mirai operators targeting popular gaming platforms such as Steam, Minecraft, and Runescape.

2.6.1 Types of Attacks

Over the course of our five month botnet infiltration, we observed Mirai operators issuing 15,194 DDoS attack commands, excluding duplicate attacks (discussed in Section 3.2). These attacks employed a range of different resource exhaustion strategies: 32.8% were volumetric, 39.8% were TCP state exhaustion, and 34.5% were application-layer attacks (Table 2.7). This breakdown differs substantially from the current landscape of DDoS attacks observed by Arbor Networks [46], where 65% of attacks are volumetric, 18% attempt TCP state exhaustion, and 18% are higher-level application attacks. While amplification attacks [47] make up 74% of attacks issued by DDoS-for-hire booter services [48], only 2.8% of Mirai attack commands relied on bandwidth

Attack Type	Attacks	Targets	Class
HTTP flood	2,736	1,035	A
UDP-PLAIN flood	2,542	1,278	V
UDP flood	2,440	1,479	V
ACK flood	2,173	875	S
SYN flood	1,935	764	S
GRE-IP flood	994	587	A
ACK-STOMP flood	830	359	S
VSE flood	809	550	A
DNS flood	417	173	A
GRE-ETH flood	318	210	A

Table 2.7: **C2 Attack Commands**—Mirai launched 15,194 attacks between September 27, 2016–February 28, 2017. These include [A]pplication-layer attacks, [V]olumetric attacks, and TCP [S]tate exhaustion, all of which are equally prevalent.

amplification, despite built-in support in Mirai's source code. This absence highlights Mirai's substantial capabilities despite the resource constraints of the devices involved.

2.6.2 Attack Targets

Studying the victims targeted by Mirai sheds light on its operators. We analyzed the attack commands issued by Mirai C2 servers (as detailed in Section 3.2) to examine who Mirai targeted. In total, we observed 15,194 attacks issued by 484 C2 IPs that overlapped with 24 DNS clusters (Section 2.5). The attacks targeted 5,046 victims, comprised of 4,730 (93.7%) individual IPs, 196 (3.9%) subnets, and 120 (2.4%) domain names. These victims ranged from game servers, telecoms, and anti-DDoS providers, to political websites and relatively obscure Russian sites (Table 2.8).

The Mirai source code supports targeting of IPv4 subnets, which spreads the botnet's DDoS firepower across an entire network range. Mirai issued 654 attacks (4.3%) that targeted one or more subnets, with the three most frequently targeted being Psychz Networks (102 attacks, 0.7%), a data center offering dedicated servers and DDoS mitigation services, and two subnets belonging to Lonestar Cell (65 combined attacks, 0.4%), a Liberian telecom. We also saw evidence of attacks that indiscriminately targeted large swathes of the IPv4 address space, including 5 distinct `/8` subnets and one attack on `/0` subnet—the entire IPv4 space. Each of the `/8` and `/0` subnets, (with the exception of the local `10.0.0.0/8`) contain a large number of distributed network operators and total IP addresses, which drastically exceed the number of Mirai bots. As such, the Mirai attacks against these subnets likely had modest impact.

If we exclude targeted subnet (due to their unfocused blanket dispersion across many networks), we find that Mirai victims were distributed across 906 ASes and 85 countries. The targets were

Target	Attacks	Cluster	Notes
Lonestar Cell	616	2	Liberian telecom targeted by 102 reflection attacks.
Sky Network	318	15, 26, 6	Brazilian Minecraft servers hosted in Psychz Networks data centers.
1.1.1.1	236	1,6,7,11,15,27,28,30	Test endpoint. Subject to all attack types.
104.85.165.1	192	1,2,6,8,11,15,21,23,26,27,28,30	Unknown router in Akamai's AS.
feseli.com	157	7	Russian cooking blog.
minomortaruolo.it	157	7	Italian politician site.
Voxility hosted C2	106	1,2,6,7,15,26,27,28,30	C2 domain from DNS expansion. Exists in cluster 2 seen in Table 2.6.
Tuidang websites	100	—	HTTP attacks on two Chinese political dissidence sites.
execrypt.com	96	—	Binary obfuscation service.
auktionshilfe.info	85	2,13	Russian auction site.
houtai.longqikeji.com	85	25	SYN attacks on a former game commerce site.
Runescape	73	—	World 26 of a popular online game.
184.84.240.54	72	1,10,11,15,27,28,30	Unknown target hosted at Akamai.
antiddos.solutions	71	—	AntiDDoS service offered at `react.su`.

Table 2.8: **Mirai DDoS Targets**—The top 14 victims most frequently targeted by Mirai run a variety of services. Online games, a Liberian cell provider, DDoS protection services, political sites, and other arbitrary sites match the victim heterogeneity of booter services. Many clusters targeted the same victims, suggesting a common operator.

heavily concentrated in the U.S. (50.3%), France (6.6%), the U.K. (6.1%), and a long tail of other countries. Network distribution was more evenly spread. The top 3 ASes—OVH (7.8%), Cloudflare (6.6%) and Comcast (3.6%)—only accounted for 18.0% of victims.

The three most frequently targeted victims were Liberia's Lonestar Cell (4.1%), Sky Network (2.1%), and `1.1.1.1` (1.6%). We examine Lonestar Cell in depth in Section **??**. Sky Network is a Brazilian company that operates servers for Minecraft (a popular game), which is hosted by Psychz Networks. The attacks against Psychz began on November 15, 2016 and occurred sporadically until January 26, 2017. `1.1.1.1` was likely used for testing [49]. Additional game targets in the top 14 victims included a former game commerce site longqikeji.com, and Runescape, another popular online game. The prevalence of game-related targets along with the broad range of other otherwise unrelated victims shares many characteristics with previously studied DDoS booter services [50].

For volumetric and TCP state exhaustion attacks, Mirai optionally specified a target port, which implied the type of service targeted. We find a similar prevalence of game targets—of the 5,450 attacks with a specified port, the most commonly attacked were 80 (HTTP, 37.5%), 53 (DNS, 11.5%), 25565 (commonly Minecraft servers [51, 52], 9.2%), 443 (HTTPS, 6.4%), 20000 (often DNP3, 3.4%), and 23594 (Runescape game server, 3.4%).

Interestingly, the 7th most common attack target was an IP address hosted by Voxility that

was associated with one of the Mirai C2 servers, and we note that 47 of 484 Mirai C2 IPs were themselves the target of a Mirai DDoS attack. By clustering these 484 C2 IPs by attack command, we identified 93 unique clusters, of which 26 (28%) were targeted least once. This direct adversarial behavior reaffirms the notion of multiple, competitive botnet operators.

2.7 DISCUSSION

Mirai has brought into focus the technical and regulatory challenges of securing a menagerie of consumer-managed, interfaceless IoT devices. Attackers are taking advantage of a reversal in the last two decades of security trends especially prevalent in IoT devices. In contrast to desktop and mobile systems, where a small number of security-conscious vendors control the most sensitive parts of the software stack (e.g. Windows, iOS, Android)—IoT devices are much more heterogeneous and, from a security perspective, mostly neglected. In seeking appropriate technical and policy-based defenses for today's IoT ecosystem, we draw on the experience of dealing with desktop worms from the 2000s.

Security hardening The Mirai botnet demonstrated that even an unsophisticated dictionary attack could compromise hundreds of thousands of Internet-connected devices. While randomized default passwords would be a first step, it is likely that attacks of the future will evolve to target software vulnerabilities in IoT devices much like the early Code Red and Confickr worms [53, 54]. To mitigate this threat before it starts, IoT security must evolve away from default-open ports to default-closed and adopt security hardening best practices. Devices should consider default networking configurations that limit remote address access to those devices to local networks or specific providers. Apart from network security, IoT developers need to apply ASLR, isolation boundaries, and principles of least privilege into their designs. From a compliance perspective, certifications might help guide consumers to more secure choices as well as pressure manufacturers to produce more secure products.

Automatic updates Automatic updates—already canonical in the desktop and mobile operating system space—provide developers a timely mechanism to patch bugs and vulnerabilities without burdening consumers with maintenance tasks or requiring a recall. Automatic updates require a modular software architecture by design to securely overwrite core modules with rollback capabilities in the event of a failure. They also require cryptographic primitives for resource-constrained devices and building PKI infrastructure to support trusted updates. Apart from these challenges, patching also requires the IoT community to actively police itself for vulnerabilities, a potentially burdensome responsibility given the sheer diversity of devices. Bug bounties can help in this

respect: roughly 25% of all vulnerabilities patched by Chrome and Firefox came from bug bounties in 2015 [55], while Netgear launched a bug bounty for its router software in January, 2017 [56]. In the event of a zero-day exploit that disables automatic updates, IoT developers must provide a secure fallback mechanism that likely requires physical access and consumer intervention.

The Deutsche Telekom infection and subsequent fix provide an excellent case study of this point. DT's routers had a vulnerability that enabled the botnet to spread via its update mechanism, which provides a reminder that basic security hardening should be the first priority. However, since DT did have an automatic update mechanism, it was also able to patch devices rather swiftly, requiring minimal user intervention. Implementing automatic updates on IoT devices is not impossible, but does take care to do correctly.

Notifications Notifications via out-of-band channels serve as a fallback mechanism to bring devices back into security compliance or to clear infections. Recent examples include alerting device administrators via CERT bulletins, emailing the abuse contact in WHOIS records, and in-browser warnings to site owners that a page is compromised [57, 58, 59]. Notifications in the IoT space are complicated to say the least. IoT devices lack both a public indication of ownership and an established communication channel to reach consumers. Were consumers reachable, there must also be a clear and simple update path to address the problem. As a minimum alternative, IoT devices could be required to register an email address with the manufacturer or with a unified, interoperable monitoring platform that can alert consumers of serious issues. This is a space where IoT requires non-technical intervention. The usability challenge of acting on notifications remains an open research problem.

Facilitating device identification Even when device models or firmware versions are known to be vulnerable, detecting such devices on the network can be extremely difficult. This made our investigation more challenging, but it also makes it hard for network operators to detect vulnerabilities in their or their customers' devices. To mitigate this, IoT manufacturers could adopt a uniform way of identifying model and firmware version to the network—say, encoding them in a portion of the device's MAC address. Disclosing this information at layer 2 would make it visible to local network operators (or to the user's home router), which could someday take automated steps to disable remote access to known-vulnerable hardware until it is updated. Achieving this in a uniform way across the industry would likely require the adoption of standards.

Defragmentation Fragmentation poses a security (and interoperability) risk to maintaining and managing IoT devices. We observed numerous implementations of Telnet, FTP, and HTTP stacks during scanning. The IoT community has responded to this challenge by adopting a handful of

operating systems, examples of which include Android Thing, RIOT OS, Tock, and Windows for IoT [60]. This push towards defragmentation would abstract away the security nuances required of our prescriptive solutions.

End-of-life Even with security best practices in mind, end-of-life can leave hundreds of thousands of in-use IoT devices without support. Lack of long-term support will yield a two class system of protected and unprotected devices similar to the current state of Windows XP machines [61]. Over time, the risk that these devices pose to the Internet commons will only grow unless taken offline.

2.8 RELATED WORK

Since as early as 2005, the security community has been working to understand, mitigate, and disrupt botnets [62]. For example, Zand et al. proposed a detection method based on identifying command and control signatures [63], and Gu et al. focused on analyzing network traffic to aid in detection and mitigation [64, 65]. Unfortunately, mitigation remains a difficult problem as botnets often evolve to avoid disruption [66].

This work follows in a long line studies that have analyzed the structure, behavior, and evolution of the botnet ecosystem [67, 68, 69, 70, 71, 72, 73]. Bailey et al. note that each technique used in understanding botnets has a unique set of trade offs, and only by combining perspectives can we fully analyze the entire picture [74]. This observation and the seminal work of Rajab et al., implicating botnet activity in 27% of all network telescope traffic, inspire our approach [75].

Botnets have historically been used to launch DDoS attacks, and there exists a parallel set of studies focusing on characterizing and defending against these attacks [76, 77], as well as estimating their effect [78]. In response to the recent growth of amplification attacks, there have been several studies investigating vulnerable amplifiers [47, 79, 80]. As DDoS attacks and infrastructure are becoming more commonplace, attention has turned to exploring the DDoS for hire ecosystem [48].

Since the emergence of IoT devices, security researchers have warned of their many inherent security flaws [81]. Researchers have found that IoT devices contain vulnerabilities from the firmware level [82, 83] up to the application level [84, 85, 86, 87]. Mirai is also not the first of its kind to target IoT devices—several precursors to Mirai exist, all of which exploit the weak password nature of these devices [22, 33, 88, 89, 90]. As a result of these widespread security failures, the security community has been quick to design systems to secure these kinds of devices. In one example, Fernandes et al. proposed Flowfence, which enables data flow protection for emerging IoT frameworks [91]. Much more work is needed if we are to understand and secure this new frontier.

In this work, we utilize a multitude of well-established botnet measurement perspectives, which substantiate concerns about IoT security. We demonstrate the damage that an IoT botnet can inflict

upon the public Internet, eclipsing the DDoS capabilities of prior botnets. We use previously introduced solutions as guidelines for our own proposals for combating the Mirai botnet, and IoT botnets at large.

2.9 LIMITATIONS AND CONCLUSION

The Mirai botnet was the first highly publicized security incident that primarily focused on IoT and embedded device. It took the Internet by storm in late 2016 when it overwhelmed several high-profile targets with some of the largest distributed denial-of-service (DDoS) attacks on record. Although our measurements were able to track its emergence, evolution, and the devices it targeted and infected, our analysis suffered from several limitations. A major limitation is that, our device identification capabilities limited by what is available through active probing of external services—we were only able to identify 31.5% of the devices scanned, with vastly differing identification rates per protocol. Beyond this, we observe many devices were routers—which may or may not be the device that was ultimately infected. As routers serve as gateways to internal networks, it is possible that a device *behind the NAT* was compromised as part of the Mirai scanning operation, and that this was lost using only our external measurement perspective. In the next chapter, we address these limitations by presenting IoT distributions captured from inside the home.

We find that while IoT devices present many unique security challenges, Mirai's emergence was primarily based on the absence of security best practices in the IoT space, which resulted in a fragile environment ripe for abuse. As the IoT domain continues to expand and evolve, we hope Mirai serves as a call to arms for those concerned about the security of an IoT-enabled world.

CHAPTER 3: MEASURING HOME IOT NETWORKS WITH ACTIVE SCANNING

3.1 INTRODUCTION

In the previous chapter, we saw how active scanning of the public IPv4 space is useful in identifying IoT device vendors, device types, and their server capabilities. However, we also observed two important shortcomings. The first is a growing challenge to identify devices from their public fingerprint—we could only identify 31.5% of devices out of all collected banners during our measurement period. The second is that the majority of devices identified were routers, which in this case serve as a measurement "black hole". What this means is that we cannot if the router itself was compromised, or a device *behind* the router was compromised, sitting inside the network that the router serves as an exit point for. In fact, in the context of home IoT devices, most such devices do not sit publicly on the Internet, but rather are hidden behind NATs, obscured from public scanning.

Beyond DDoS attacks, the weak security posture of many popular IoT devices has enabled attackers to conduct a litany of attacks, such as compromising local networks [92, 93] and breaking into homes [94, 95]. However, despite much attention to IoT in the security community [94, 96, 97, 98, 99], there has been little investigation into what devices consumers are adopting and how they are configured in practice.

In this chapter, we provide a large-scale empirical analysis of 83M IoT devices in 16M real-world homes. We partner with Avast Software, a popular antivirus company, whose consumer security software lets customers scan their local network for IoT devices that support weak authentication or have remotely exploitable vulnerabilities. inventory of devices it finds.

Leveraging data collected from user-initiated network scans in 16M households that have agreed to share data for research and development purposes, we describe the current landscape of IoT devices and their security posture.

IoT devices are widespread. More than half of households have at least one IoT device in three global regions and in North America more than 66% of homes have a network-connected device. Media devices like smart televisions are most common in seven of eleven global regions, but there is significant variance otherwise. For example, surveillance cameras are most popular in South and Southeast Asia, while work appliances prevail in East Asia and Sub-Saharan Africa. Home assistants are present in 10% of homes in North America but have yet to see significant adoption in other markets. There is a long tail of 14K total manufacturers, but surprisingly we find that 90% of devices worldwide are produced by only 100 vendors. A handful of companies like Apple, HP, and Samsung dominate globally, but there also exist a set of smaller vendors with significant regional

adoption. For example, Vestel, a Turkish manufacturer, is the third largest media vendor in North Africa and the Middle East, but has negligible broader adoption.

A surprising number of devices still support FTP and Telnet with weak credentials. In Sub-Saharan Africa, North Africa, the Middle East, and Southeast Asia, around half of devices support FTP and in Central Asia, nearly 40% of home routers use Telnet. Similar to the regional differences in device type and manufacturer popularity, there are dramatic differences in the use of weak credentials. For example, while less than 15% of devices with FTP allow weak authentication in Europe and Oceania, more than half do in Southeast Asia and Sub-Saharan Africa. Interestingly, this is not entirely due to manufacturer preference. While less than 20% of TP-Link home routers allow access to their administration interface with a weak password in North America, nearly half do in Eastern Europe, Central Asia, and Southeast Asia. About 3% of homes in our dataset are externally visible and more than half of those have a known vulnerability or weak password.

Our results indicate that IoT is not a security concern of the future, but rather one of today. We argue that there already exists a complex ecosystem of Internet-connected embedded devices in homes worldwide, but that these devices are different than the ones considered by most recent work. We hope that by shedding light on the devices consumers are purchasing, we enable the security community to develop solutions that are applicable to today's homes.

This chapter in its entirety appeared at the USENIX Security Symposium in 2019.

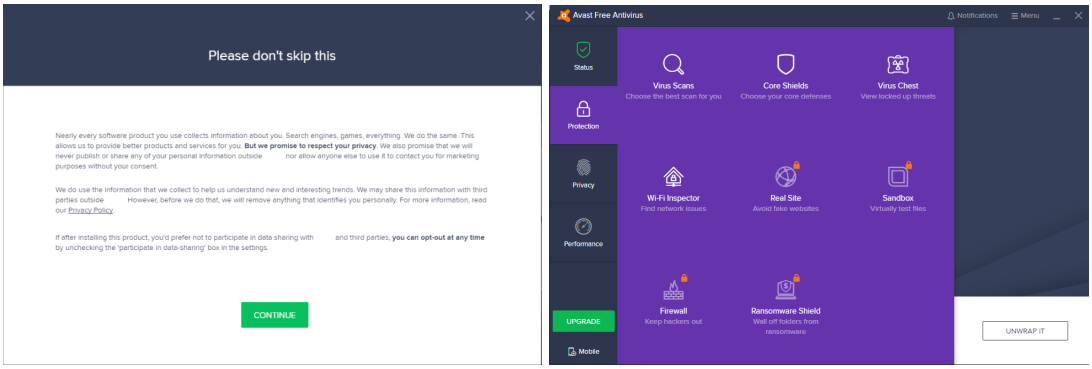
3.2 METHODOLOGY AND DATASET

Our study leverages several network vantage points, including data collected from Avast, a passive network telescope, and active Internet-wide scans. In this section, we discuss these datasets and the role they play in our analysis.

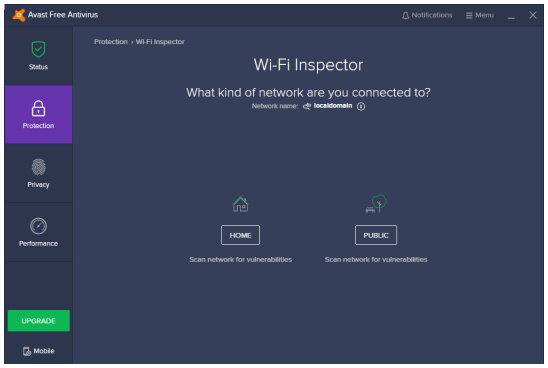
3.2.1 WiFi Inspector

Avast Software is a security software company that provides a suite of popular antivirus and consumer security software products like *Avast Free Antivirus*. Avast software is sold on a freemium model: the company provides a free basic version of their product and charges for more advanced versions. Avast estimates that their software runs on 160 M Windows and 3 M Mac OS computers, and makes up approximately 12% of the antivirus market share [100].

As of 2015, all antivirus products from Avast include a tool called *WiFi Inspector* that helps users secure IoT devices and other computers on their home networks. WiFi Inspector runs locally on the user's personal computer and performs network scans of the local subnet to check for devices that



(a) **Data Sharing Consent** (b) **WiFi Inspector Drawer**



(c) **WiFi Inspector Initiation**

Figure 3.1: **WiFi Inspector**—WiFi Inspector allows users to scan their local network for insecure IoT devices. Data sharing back to Avast for research purposes is an explicit part of the installation process, and presented to the user in plain English. For ease of reading, we duplicate the text shown in panel (a) in Appendix A.1.

accept weak credentials or have remotely exploitable vulnerabilities. Scans can also be manually initiated by the end user. WiFi Inspector alerts users to security problems it finds during these scans and additionally provides an inventory of labeled IoT devices and vulnerabilities in the product's main interface (Figure 3.1). We next describe how WiFi Inspector operates:

Network Scanning To inventory the local network, WiFi Inspector first generates a list of scan candidates from entries in the local ARP table as well through active ARP, SSDP, and mDNS scans. It then probes targets in increasing IP order over ICMP and common TCP/UDP ports to detect listening services.[1] Scans terminate after the local network has been scanned or a timeout occurs.

[1] WiFi Inspector scans several groups of TCP/UDP ports: common TCP ports (e.g., 80, 443, 139, 445); TCP ports associated with security problems (e.g., 111, 135, 161); common UDP ports (e.g., 53, 67, 69); and ports associated with services that provide data for device labeling (e.g., 20, 21, 22). When hosts are timely in responding, the scanner will additionally probe a second set of less common ports (e.g., 81–85, 9971). In total, the scanner will target up to 200 ports depending on host performance. The scanner will identify devices so long as they are connected to the network.

After the discovery process completes, the scanner attempts to gather application layer data (e.g., HTTP root page, UPnP root device description, and Telnet banner) from listening services.

Detecting Device Types To provide users with a human-readable list of hosts on their network, WiFi Inspector runs a classification algorithm against the application-and transport-layer data collected in the scan. This algorithm buckets devices into one of fourteen categories:

1. Computer
2. Network Node (e.g., home router)
3. Mobile Device (e.g., iPhone or Android)
4. Wearable (e.g., Fitbit, Apple Watch)
5. Game Console (e.g., XBox)
6. Home Automation (e.g., Nest Thermostat)
7. Storage (e.g., home NAS)
8. Surveillance (e.g., IP camera)
9. Work Appliance (e.g., printer or scanner)
10. Home Voice Assistant (e.g., Alexa)
11. Vehicle (e.g., Tesla)
12. Media/TV (e.g., Roku)
13. Home Appliance (e.g., smart fridge)
14. Generic IoT (e.g., toothbrush)

We consider devices in the latter eleven categories to be IoT devices for the remainder of this work. Because the classifier greatly affects the results of this work, we describe the algorithm in detail in Section 3.2.2.

Manufacturer Labeling To generate a full device label, WiFi Inspector combines device type with the device's manufacturer (e.g., Nintendo Game Console). Avast determines manufacturer by looking up the first 24 bits of each device's MAC address in the public IEEE Organizationally Unique Identifier (OUI) registry [101]. We note that at times, the vendor associated with a MAC address is the manufacturer of the network interface rather than the device. For example, MAC addresses associated with some Sony Playstations belong to either FoxConn or AzureWave, two major electronic component manufacturers, rather than Sony. In this work, we manually resolve and document any cases that required grouping manufacturers together.

Checking Weak Credentials WiFi Inspector checks for devices that allow authentication using weak credentials by performing a dictionary-based attack against FTP and Telnet services as well as web interfaces that use HTTP basic authentication. When possible, WiFi Inspector will also try to log into HTTP-based administration interfaces that it recognizes. The scanner attempts to log in with around 200 credentials composed of known defaults (e.g., `admin/admin`) and commonly used strings (e.g., `user`, `1234`, `love`) from password popularity lists, leaks, vendor and ISP default lists, and passwords checked by IoT malware. WiFi Inspector immediately notifies users about devices with guessable logins.

Checking Common Vulnerabilities In addition to checking for weak credentials, WiFi Inspector checks devices for vulnerability to around 50 recent exploits that can be verified without harming target devices (e.g., CVE-2018-10561, CVE-2017-14413, EDB-ID-40500, ZSL-2014-5208, and NON-2015-0211). Because there is bias towards more popular manufacturers in these scans, we do not provide ecosystem-level comparisons between different vulnerabilities.

3.2.2 Device Identification Algorithm

A significant portion of our work is based on identifying the manufacturers and types of IoT devices in homes. We describe the algorithm that Avast has developed in this section:

Classifier WiFi Inspector labels device type (e.g., computer, phone, game console) through a set of expert rules and a supervised classification algorithm, both of which run against network and application layer data. Classification is typically possible because manufacturers often include model information in web administration interfaces as well as in FTP and Telnet banners [102]. Additionally, devices broadcast device details over UPnP and mDNS [103]. WiFi Inspector uses expert rules—regular expressions that parse out simple fields (e.g., telnet banner or HTML title)—to label hosts that follow informal standard practices for announcing their manufacturer and model. This approach, while not comprehensive, reliably identifies common devices [14, 102]. WiFi Inspector contains approximately 1,000 expert rules that are able to identify devices from around 200 manufacturers. We show a sample of these rules in Table 3.1. However, these rules only identify 60% of devices from a random sample of 1,000 manually-labeled devices. To categorize the remaining devices, WiFi Inspector leverages an ensemble of four supervised learning classifiers that individually classify devices using network layer-data, UPnP responses, mDNS responses, and HTTP data. Therefore, when identifying a device, WiFi Inspector first tries the expert rules, and in the case of no match, next applies the ensemble of four supervised classifiers.

The network classifier is built using a random forest, which aggregates the following network features of a device:

1. MAC address
2. Local IP address
3. Listening services (i.e., port and protocol)
4. Application-layer responses on each port
5. DHCP `class_id` and hostname

The UPnP, mDNS and HTTP classifiers leverage raw text responses. The classifier treats each response as a bag-of-words representation, and uses TF-IDF to weight words across all responses. This representation is fed as input to a Naïve Bayes classifier.

Training and Evaluation To train the supervised algorithm, Avast collected data on approximately 500K random devices from real-world scans. 200K of these were manually classified through an iterative clustering/labeling process, where experts clustered devices based on network properties and labeled large clusters, winnowing and re-clustering until all devices were labeled. The remaining 300K devices were labeled using the expert rules. To tune model parameters, we performed five-fold cross-validation across the original training set. However, because the initial clustering was used to help identify devices in the clustering/labeling step, the dataset is not used for validation. Instead, Avast curated a validation set of 1,000 manually labeled devices, whose labels were never used for training. The final classifier achieves 96% accuracy and 92% coverage with a 0.80 macro average F1 score (Table 3.2). We mark devices we cannot classify as "unknown".

Protocol	Field	Search Pattern	Device Type Label	Confidence
DHCP	Class ID	`(?i)SAMSUNG[- :_]Network[- :_]Printer`	Printer	0.90
UPnP	Device Type	`.*hub2.*`	IoT Hub	0.90
HTTP	Title	`(?i)Polycom - (?:SoundPoint IP )?(?:SoundStation IP )?`	IP Phone	0.85
mDNS	Name	`(?i)_nanoleaf(?:api|ms)?\._tcp\.local\.`	Lighting	0.90

Table 3.1: **Example Device Classification Rules**—Our device labeling algorithm combines a collection of 1,000 expert rules and a supervised classifier, both of which utilize network and application layer data. Here, we show a few examples of these expert rules, which provide 60% coverage of devices in a random sample of 1,000 devices.

Classifier	Coverage	Accuracy	Macro F1
Supervised Ensemble	0.91	0.95	0.78
Network	0.89	0.96	0.79
UPnP	0.27	0.91	0.37
mDNS	0.05	0.94	0.25
HTTP	0.14	0.98	0.23
Final Classifier	0.92	0.96	0.80

Table 3.2: **Device Classifier Performance**—Our final classifier combines the supervised classifier and expert rules, and achieves 92% coverage and 96% accuracy against a manually labeled set of 1,000 devices.

3.2.3 Avast Dataset

Avast collects aggregate data about devices, vulnerabilities, and weak credentials from WiFi Inspector installations of consenting users for research and development purposes. Users are informed about this data collection in simple English when they install the product (Figure 3.1) and can opt out at any time. We worked with Avast to analyze *aggregate data* about the types of devices in each region. No individual records or personally identifiable information was shared with our team. Although WiFi Inspector supports automatic vulnerability scans, we only use data from user-initiated scans in this paper so that we can guarantee that users knowingly scanned their networks. In addition, we exclude scans of public networks by only analyzing networks that were marked as home networks in Windows during network setup. We detail the ethical considerations and our safeguards in Section 3.2.6.

We specifically analyze data about devices found in scans run between December 1–31, 2018 on Windows installations. This dataset consists of data about 83 M devices from 15.5 M homes spanning 241 countries and territories, and 14.3 K unique manufacturers. For installations with multiple scans during this time period, we use the latest scan that found the maximum number of devices. We aggregate each country into 11 regions, defined by ISO 3166-2 [104]. As shown in Table 3.3, WiFi Inspector is more popular in Europe and South America than in North America. Because of this market share, as well as significant regional differences in IoT deployment, we discuss regions separately.

Threats to Validity While WiFi Inspector is installed in a significant number of homes, the dataset is likely colored by several biases. First, the data is predicated on users installing antivirus software on their computers. There is little work that indicates whether users with antivirus software have more or less secure practices. Second, we only analyzed data from installations on Windows machines due to differences between Mac and Windows versions of the software. This may skew

Region	Homes		Devices	
North America	1.24 M	(8.0%)	9.2 M	(11.1%)
South America	3.2 M	(20.9%)	18 M	(21.6%)
Eastern Europe	4.2 M	(27.2%)	18.8 M	(22.6%)
Western Europe	2.9 M	(19.1%)	15 M	(18.0%)
East Asia	543 K	(3.5%)	3 M	(3.7%)
Central Asia	107 K	(0.7%)	500 K	(0.6%)
Southeast Asia	813 K	(5.3%)	3.6 M	(4.3%)
South Asia	824 K	(5.3%)	6.6 M	(7.7%)
N. Africa, Middle East	1.2 M	(7.5%)	6.1 M	(7.3%)
Oceania	124 K	(0.8%)	680 K	(0.8%)
Sub-Saharan Africa	266 K	(1.7%)	1.8 M	(2.2%)

Table 3.3: **Regional Distribution of Homes**—The 15.5M homes and 83M devices in our dataset are from geographically diverse regions. Because this breakdown is representative of Avast market share rather than organic density of homes and devices, we limit our analysis to within individual regions.

the households we study to different socioeconomic groups or introduce other biases. Third, WiFi Inspector *actively notifies* users about problems it finds. As a result, users may have patched vulnerable hosts, changed default passwords, or returned devices to their place of purchase. This may skew our results to indicate that homes included in this study are more secure than in practice.

3.2.4 Network Telescope

While WiFi Inspector scans can identify the types of devices present in home networks, the data does not provide any insight into whether devices have been compromised. To understand whether devices are infected and scanning to compromise other devices (e.g., as was seen for Mirai [102]), we consider the IP addresses scanning in a large network telescope composed of approximately 4.7 million IP addresses. We specifically analyze the traffic for a 24 hour period on January 1, 2019 for scan activity using the methodology discussed by Durumeric et al. [105]: we consider an IP address to be scanning if it contacts at least 25 unique addresses in our telescope on the same port within a 480 second window. In total, we observe 1.7 M scans from a total of 529 K unique IP addresses from 1.4 billion packets during our measurement period. Of the 500,716 homes scanned by WiFi Inspector on this day, 1,865 (0.37%) were found scanning on the network telescope.

3.2.5 Internet-Wide Scanning

We further augment the WiFi Inspector data with data collected from Internet-wide scans performed by Censys [106] to understand whether the vulnerabilities present on gateways (i.e., home routers) could be remotely exploitable. Similarly to our network telescope data, we investigate the intersection between Censys and Avast data for a 24-hour period on January 30, 2019 to control for potential DHCP churn. We also check whether devices that accept weak credentials for authentication present login interfaces on public IP addresses. We discuss the results in Section 3.4.

3.2.6 Ethical Considerations

WiFi Inspector collects data from inside users' homes. To ensure that this data is collected in line with user expectations, we only collect statistics about homes where the user explicitly agreed to share data for research purposes. This data sharing agreement is not hidden in a EULA, but outlined in simple English. We show the dialogue where users acknowledge this in Figure 3.1. We note that this is an explicit *opt-out* process. The data sharing agreement is the last message shown to the user before the main menu, meaning users do not need to wait and remember to turn off data collection at a later time.

In order to keep up to date information on the devices in a home, WiFi Inspector runs periodic, automated scans of the local network. Automated scans do not perform any vulnerability testing or password weakness checks; they only identify devices through banners and MAC addresses. We limit our analysis to homes where a user explicitly *manually initiated* a network scan.

To protect user privacy and minimize risk to users, Avast only shared aggregate data with our team. This data was aggregated by device manufacturer, region, and device type. The smallest region contained over 100,000 homes. We never had access to data about individual homes or users; no personally identifiable information was ever shared with us. Avast did not collect any additional data for this work, nor did they change the retention period of any raw data. No data beyond the aggregates in this paper will be stored long term.

Internally, Avast adheres to a strict privacy policy: all data is anonymized and no personally identifiable information is ever shared with external researchers. All handling of WiFi Inspector data satisfies personal data protection laws, such as GDPR, and extends to data beyond its territorial scope (i.e., outside of the European Union). Specific identifiers like IP addresses are deleted in accordance with GDPR and only collected when explicitly necessary for the security function of the product.

	IoT	Media/TV		Work Appl		Gaming		Voice Asst.		Surveil.		Storage		Automat.		Wearable		Other IoT	
Region	Homes	Homes	Devices	H	D	H	D	H	D	H	D	H	D	H	D	H	D	H	D
North America	66.3%	42.8	44.9	32.7	28.0	16.0	12.0	9.5	7.5	3.9	3.7	2.7	1.7	2.3	1.9	0.2	0.1	0.4	0.2
South America	31.7%	20.5	51.7	7.5	24.0	4.3	9.8	0.1	0.3	4.6	13.3	0.3	0.6	0.0	0.1	0.0	0.1	0.1	0.2
Eastern Europe	25.2%	16.8	50.2	6.0	23.6	2.7	7.6	0.2	0.6	2.5	14.0	1.2	3.4	0.1	0.4	0.0	0.1	0.0	0.0
Western Europe	53.5%	40.2	59.0	14.0	18.9	7.5	9.2	1.8	2.3	3.8	5.6	2.5	3.2	1.3	1.6	0.0	0.0	0.0	0.0
East Asia	30.8%	12.2	25.8	14.9	44.5	6.3	12.1	0.9	1.6	2.2	9.1	3.1	6.5	0.1	0.2	0.1	0.2	0.0	0.1
Central Asia	17.3%	13.5	54.2	1.6	12.0	0.6	2.4	0.0	0.2	2.4	30.3	0.2	0.8	0.0	0.0	0.0	0.1	0.0	0.0
Southeast Asia	21.7%	9.0	25.4	7.5	31.2	1.0	2.7	0.2	0.5	7.8	37.0	0.9	2.7	0.1	0.2	0.1	0.3	0.0	0.0
South Asia	8.7%	2.5	16.6	2.7	24.2	0.4	2.4	0.1	0.8	4.1	54.5	0.2	1.1	0.0	0.2	0.0	0.2	0.0	0.0
N. Africa, M. East	19.1%	9.4	35.7	5.1	26.2	1.8	6.4	0.1	0.3	5.2	28.5	0.7	2.4	0.0	0.2	0.0	0.2	0.0	0.1
Oceania	49.2%	30.7	46.6	19.8	25.9	10.1	12.7	3.2	4.2	3.0	5.3	3.5	4.3	0.7	0.9	0.1	0.2	0.0	0.0
Sub-Saharan Africa	19.7%	6.9	21.7	10.9	49.9	2.5	7.1	0.1	0.4	2.8	18.0	0.8	2.3	0.1	0.3	0.1	0.3	0.0	0.1

Table 3.4: **IoT in Homes**—We show the percent of households that have one or more of each type of IoT device and the percent of devices (in gray) in each region that are of a certain type. For example, 42.8% of homes in North America have at least one media device and 44.9% of North American IoT devices are media devices. For the presence of any IoT device, we only report the percent of homes with an IoT device.

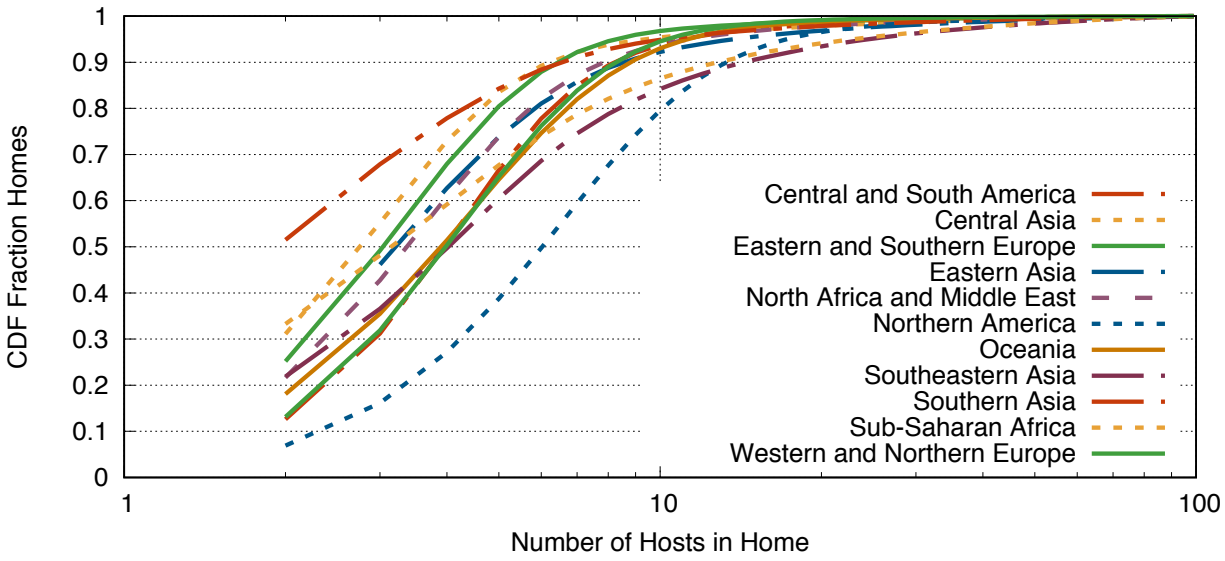


Figure 3.2: **Devices per Region**—There is significant variance in device usage across regions. The largest presence is in North America, where homes have a median seven hosts. Conversely, homes in South Asia have a median two hosts. The number of devices per home starts at two as all homes require at least one computer and one router to be included.

3.3 IOT IN HOMES

It is vital that the security community understands the types of IoT devices that consumers install and their respective regional distributions given their increasing security and privacy implications. In this section, we provide one of first large-scale analyses of these devices based on scans from 15.5 M homes.

The presence of IoT devices varies by region. For example, while more than 65% of homes in North America have an IoT device, fewer than 10% of homes in South Asia do (Figure 3.2). Media devices (i.e., smart TVs and streaming devices) are the most common type of device in seven of the eleven regions, in terms of both presence in homes (2.5%–42.8%) and total number of devices (16.6%–59.0%). Four regions differ: surveillance devices are most common in South and Southeast Asia, while work appliances are most common in East Asia and Sub-Saharan Africa. We show the most popular devices in each region in Table 3.4.

Despite differences in IoT popularity across regions, there are strong correlations between regions for the *types* of devices that are popular.[2] In other words, the most popular types of devices are similar across regions. Still, certain pairs of regions differ. For example, homes in all Asian regions

[2]To quantify the preference for difference types of devices across regions, we leverage a Spearman's rank correlation test across each pairwise region, taking the rank ordered list of device types for each region as input (Table 3.5). Per Cohen's guidelines, we find all regions rank ordered distributions are strongly correlated (>0.7 coefficient) with p-values < 0.05 [107], indicating little change in the rank order of device type distributions across regions.

	N. America	S. America	E. Europe	W. Europe	East Asia	Central Asia	SE Asia	South Asia	N. Africa, ME	Oceania	S-S Africa
North America	–	81	88	92	88	76	77	81	87	93	86
South America	81	–	87	85	90	85	88	87	90	90	92
E. Europe	88	87	–	95	95	93	93	94	98	98	96
W. Europe	92	85	95	–	90	88	83	87	92	95	89
East Asia	88	90	95	90	–	90	93	92	93	98	99
Central Asia	76	85	93	88	90	–	93	90	94	90	93
Southeast Asia	77	88	93	83	93	93	–	99	95	96	95
South Asia	81	87	94	87	92	90	99	–	97	92	95
N. Africa, Middle East	87	90	98	92	93	94	95	97	–	96	95
Oceania	93	90	98	95	98	90	96	92	96	–	96
Sub-Saharan Africa	86	92	96	89	99	93	95	95	95	96	–

Table 3.5: **Regional Similarities**—We calculate the similarity regions by computing the Spearman's rank correlation test over each region's rank order list of most popular types of devices. We show the most similar region (green) and least similar region (red) by row. Correlation coefficients presented are out of 100. In all cases, p-values were < 0.05.

are least similar to homes in North America. On the other hand, homes in geographically similar regions (e.g., South Asia and Southeastern Asia) are highly correlated, even when they differ from the global distribution. The fact that distinct regions have unique preferences for device types points to deeper differences between regions, making it harder to reason about IoT in aggregate and more challenging to generalize findings from one region to others.

We also considered the relative popularity of types of devices within each region. Even in areas with similar rank order popularity, the proportion of device types in those regions varies (Figure 3.3). We compute a pairwise proportion test across each region to quantify the differences between regions and find that nearly all regions have varying proportions of IoT device types, except when a device type accounts for fewer than 1% of devices. We discuss each region below.

3.3.1 North America

North America has the highest density of IoT devices of any region: 66.3% of homes have an IoT device compared to 34% globally. Similar to other regions, media devices (e.g., TVs and streaming boxes) and work appliances account for the most devices in North American homes. Nearly half of homes have one media device and one third have a work appliance (Table 3.4). Media devices are also the most prolific, accounting for 44.9% of IoT devices in North America. In contrast, work appliances only account for 28% of devices (Table 3.4). There is a long tail of manufacturers that produce media devices in the U.S., and the most popular vendor, Roku, only accounts for 17.4% of

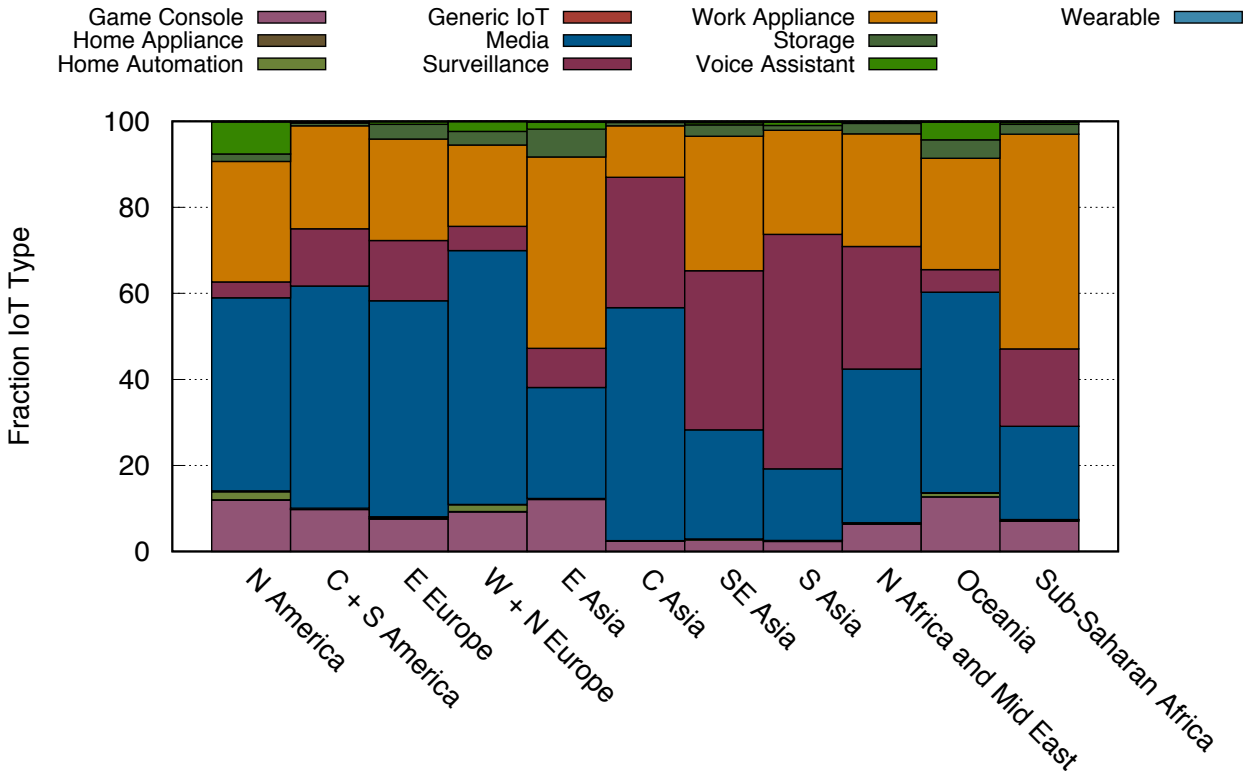


Figure 3.3: **IoT Device Distribution by Region**—IoT device type distributions vary between different geographic regions. For example, Surveillance devices are most prevalent in Asia, whereas Home Automation devices only appear in North America and Europe.

media devices (Table A.1). Second most popular is Amazon (10.2%). In contrast, there are only a handful of popular work appliance vendors—HP is the most common and accounts for 38.7% of work appliances in North America.

Though popular in every region, a considerably higher number of homes in North America contain a game console. This is one of the reasons that a smaller fraction of IoT devices are media-related than in Western and Northern Europe. There are three major vendors of game consoles: Microsoft (39%), Sony (30%),[3] and Nintendo (20%).

North America is the only region to see significant deployment of home voice assistants like Amazon Echo [108] and Google Home [109]. Nearly 10% of homes now have a voice assistant and the device class accounts for 7.5% of IoT devices in the region. Two thirds of home assistants are Amazon produced, the remaining one third are Google devices. North America is also one of the only region to see automation devices, which are present in 2.5% of homes. There are four major manufacturers in this space, Nest[4] (44.2%), Belkin (15.1%), Philips (14.4%), and Ecobee (9.8%).

[3]Sony PlayStation devices are split across three vendors in this distribution primarily due to their network cards being manufactured by two third party vendors, Azurewave (11.6%) and Foxconn (9%).

[4]A classification error misclassifies Nest products as mobile devices. We manually correct this in our analysis since

These vendors sell products such as the Nest Thermostat [110], Wemo smart plug [111], Philips Hue Smart Lights [112], and the Ecobee Smart Thermostat [113].

The relative ranking of IoT device type popularity generally does not change as more IoT devices are added to North American homes. To quantify this, we calculate the Spearman rank correlation for each pairwise set of homes based on the number of devices and observe only slight deviations from the overall regional distribution. As more devices are added to the network, the correlation coefficients for North America hover between 0.98–1.0, indicating minimal change. Despite minimal change in the relative ranking of IoT device types, we note that the fraction of each device type does vary as more IoT devices are added to the home. For example, for homes with one IoT device, voice assistants make up only 3.9% of all devices, down from 7.3% across all homes. Game consoles are also more popular in homes with only one IoT device, up from 13.9% to 16.5%.

3.3.2 Central and South America

South American homes are the least similar to North America of any region (Table 3.5). While the most common types of IoT devices in both regions are media devices (51.7% vs 44.9%) and work appliances (24% vs 28%), significantly fewer South American homes have an IoT device (32% vs 66%) and there are significantly more surveillance devices: 13.3% vs 3.7% of devices (Table 3.4). Prior research uncovered that there is an increased reliance on surveillance devices in Brazil and surrounding regions to deter violence [114, 115], which may offer one explanation. The only other device type we commonly see are game consoles (9.8% of devices). No other class appears in more than a fraction of a percent of homes.

The vendor distribution of media devices in Central and South America differs from the global distribution. Two vendors appear in the top 5 for this region that do not appear in any other region. First is Arcadyan, a Taiwanese company that primarily manufactures cable boxes in this category, and is often found in LG Smart TVs. The second is Intelbras, a Brazilian company that manufactures DVRs and smart video players. Intelbras accounts for 11% of the surveillance cameras in the region, though they are third to Hikvision and Dahua.

3.3.3 Europe

Eastern and Western Europe are both most similar to Oceania, primarily due to the three regions sharing a similar fraction of storage devices (Table 3.4). Still, the regions vary in terms of their IoT usage: 53.5% of Western European homes have at least one IoT device, compared to 25.2% in Eastern European homes.

Nest does not sell mobile devices.

Manufacturers in Western Europe are similar to the global distribution with a handful of exceptions. Sagemcom and Free, two French companies that sell media boxes and IP cameras, are the first and third largest media vendors in Western Europe, accounting for 15.7% and 9.3% of all devices compared to 5.7% and 3.2% globally. The markets of both companies are highly localized, as 99% of their devices in our dataset are located in Western and Northern Europe. In other device categories, such as work appliances, game consoles, and home automation, there is limited variance from the global distribution. Outside of North America and Oceania, Western Europe is the only other region where more than 1% of homes have a home automation device.

There are significantly more surveillance devices in Eastern Europe than Western Europe (14% versus 5.6% of devices). Eastern Europe is also unlike most other regions in that its rank ordered device type distribution changes as more IoT devices are added over time. For homes with one IoT device, surveillance devices only make up 5.3% of all IoT devices, but this changes drastically for homes with 3 IoT devices, where the number of surveillance devices shoots up to 13.8%. The fraction of surveillance devices continually increases as more IoT devices are added to Eastern European homes. In homes with 10 IoT devices, surveillance devices are the most popular device, accounting for 42.7% of all devices.

3.3.4 Asia

We analyze the four regions (East, Central, South, and Southeast) of Asia separately as they have different IoT profiles. For example, surveillance devices make up 54.5%, 37%, and 30.3% of devices in South, Southeast, and Central Asia (Figure 3.3), whereas only 9.1% of devices are surveillance related in East Asia. This is not due to a large number of homes with cameras, but rather that other types of IoT devices are sparse. For example, only 9% of S.E. Asian Homes and 2.5% of South Asian homes contain a media device whereas more than 40% homes in North America and Western Europe do. Similar to other regions, Hikvision is the most prevalent vendor of surveillance devices in S.E. Asia and South Asia, making up 25.8% and 34.7% of surveillance devices in each region respectively. Unlike other regions, a private[5] vendor accounts for 15.5% of all surveillance devices in Southern Asia.

East and Central Asia are more similar to Eastern Europe and Africa than they are to South and Southeast Asia. East Asia, for example, is most similar to Sub-Saharan Africa because its largest device type is work appliances, which make up 44.5% of the devices in the region. Central Asia more closely follows Eastern Europe with media devices accounting for 54.2% of devices. All Asian regions do have one thing in common: they are all the least similar to North American homes,

[5]Private vendors are ones that have paid an additional fee to IEEE to keep their MAC address mapping off of the public OUI list.

indicating fundamental differences in IoT device usage between the Asian countries and North America.

3.3.5 Africa and Middle East

The North Africa, Middle East (combined) region is most similar to Eastern Europe. Media devices are the most prevalent, appearing in 9.4% of homes and accounting for 35.7% of devices. Again, we observe a local media vendor with a large presence: Vestel, a Turkish TV manufacturer, is the third largest media vendor after Samsung and LG. Surveillance devices make up 28.5% of their overall devices, and appear in 5.2% of homes. Sub-Saharan Africa is distinct in that work appliances are most popular (50% of devices). 11% homes in the region have at least one work appliance. The most popular vendor is HP (33.6%), followed by a long tail of other manufacturers.

3.3.6 Oceania

Oceania ranks third to North America and Western Europe in terms of fraction of homes that contain an IoT device (49.2% of homes). Similar to other regions, the most popular device type in the region are media devices, which are found in 30.7% of homes. This is followed by work appliances (19.8% of homes) and gaming consoles (10.1% of homes). Oceania is one of the only regions that contains home automation devices, appearing in 0.7% of homes in our dataset. Similar to North America and Western Europe, Oceania has a moderate number of voice assistant devices, which appear in 3.2% of homes and account for 4.2% of all devices. Unlike North America and Western Europe, homes in Oceania contain many networked storage devices. They account for 4.3% of all devices, which is most similar to homes in Eastern Europe and East Asia.

3.3.7 IoT Device Vendors

While we find devices from 14.3K unique vendors, 90% of all devices globally are manufactured by 100 vendors (Figure 3.4). Globally, there are 4,157 vendors (29%) that only appear in one home. Unlike device type distributions, which are consistent across region, vendor distributions vary heavily across device type (Figure 3.5). Some device types are dominated by a small handful of vendors. For example, Amazon and Google account for over 90% of voice assistant devices globally. Other device types like media devices and surveillance devices are split across many vendors. Media devices are the most heterogeneous by vendor: the top 10 vendors only account for 60% of devices.

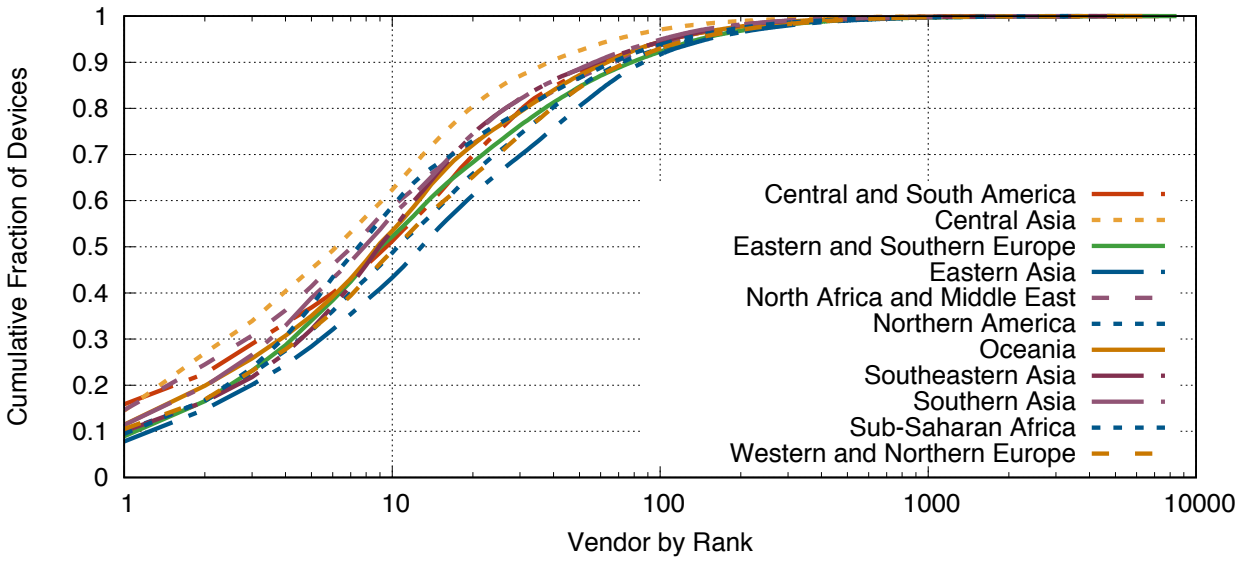


Figure 3.4: **Vendors per Region**—There are a long tail of vendors per region. In all regions, 100 vendors account for more than 90% of devices and 400 vendors account for 99%.

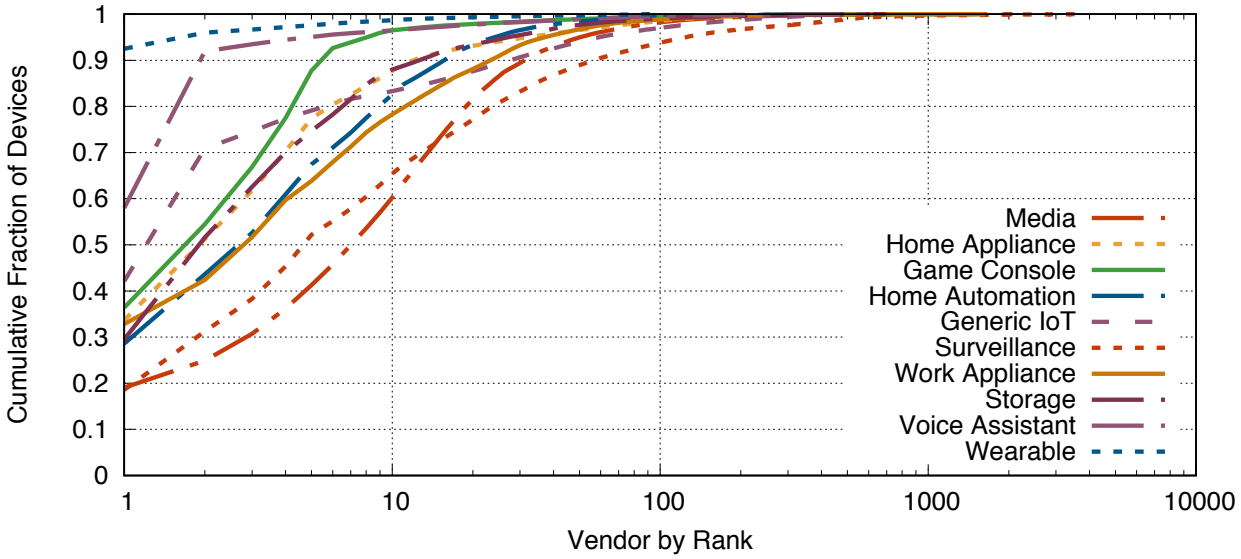


Figure 3.5: **IoT Vendors per Region and Device Type**—Some device types are almost entirely dominated by one or two vendors. For example, Amazon and Google produce 91.9% of voice assistants and Hikvision produces 18.6% of surveillance devices.

Regional differences in vendor preferences may cause the observed variance in vendor distributions across device types. To measure this, we compute the pairwise Spearman's correlation for each vendor distribution across every pair of regions (e.g. vendor distribution for voice assistants in North America vs. East Asia). We then aggregate[6]over device type by taking the average correlation

Device Type	Mean Correlation	Top-10 Mean Correlation
Game Console	**0.43**	**0.49**
Voice Assistant	**0.23**	**0.26**
Home Automation	**0.98**	**0.98**
Surveillance	0.07	**0.28**
Work Appliance	0.04	**0.22**
Storage	0.05	-0.03
Media	0.04	0.09
Router	0.01	0.02
Mobile Device	0.01	0.03

Table 3.6: **Vendor Correlation by Device Type**—We show the mean correlation in rank ordered vendor distributions per device type across every pair of regions across all vendors as well as the top 10 vendors in each category. The correlations in bold are statistically significant, and indicate consistency in vendors for these device types across all regions in our dataset.

across each pair of regions (Table 3.6).

We observe that device types dominated by a handful of vendors globally (Figure 3.5) show moderate to strong correlations across all regions, indicating stability in popular vendors across geographic areas. For example, game consoles are dominated by three major players (Microsoft, Sony, Nintendo) in almost every region across the world. In contrast, there are a number of device types, such as media and storage devices, for which there are no correlations across region, even when looking only at the top 10 vendors. This indicates that for these device types, regions have differing vendor preferences. This result aligns with our investigation of individual regions, where we observed many regions prefer local media vendors that are less prevalent in the global distribution.

3.4 HOME SECURITY

Beyond understanding the landscape of IoT devices, we investigate the security profile of devices in homes, including devices that allow weak authentication, the security profile of home routers, and the presence of homes that exhibit scanning behavior on a large darknet.

Many IoT devices act as embedded servers: 67.5% of devices provide at least one TCP- or UDP-based service. Many of these services are not surprising—network printers necessarily run

[6]We note that correlation coefficients are not additive, so to aggregate we convert the respective correlation r-values to z-values using a Fisher's Z transform [116], take the average of the Z values, and convert back to an r-value. In addition, we could only compare rank order for vendors who appeared in all 11 regions in the dataset. There were three device categories (wearables, home appliances, generic IoT) for which no vendors appeared in all regions; we could not compute correlations in these cases.

Port	Service	Devices	Port	Service	Devices
1900	UPnP	46.2%	139	SMB	10.6%
80	HTTP	45.7%	8443	HTTPS Alt.	9.5%
5353	mDNS	39.2%	8009	HTTP Alt.	9.3%
8080	HTTP Alt.	26.9%	445	SMB	8.7%
443	HTTPS	21.1%	7676	Custom	8.2%
9100	JetDirect	19.5%	49152	–	7.9%
515	LPR	16.5%	21	FTP	7.8%
631	IPP	11.8%	5000	UPnP	7.8%
554	RTSP	11.8%	23	Telnet	7.1%
8008	HTTP Alt.	11.1%			

Table 3.7: **Popular IoT Services**—We show the common open ports in IoT devices in our dataset. The most popular protocols are related to device discovery (UPnP, mDNS) and device administration (HTTP, HTTPS).

services like IPP. However, we also note that devices commonly support older protocols like Telnet (7.1% of IoT devices) and FTP (7.8%). The most common protocol is Universal Plug and Play (UPnP), which is prevalent on 46.2% of devices. We also observe HTTP and mDNS on nearly half of devices. We show the top protocols in Table 3.7.

3.4.1 Weak Device Credentials

WiFi Inspector identifies devices that allow authentication with weak default credentials by attempting to log in to FTP and Telnet services with a small dictionary of common default credentials (Section 3.2). We find that 7.1% of IoT devices and 14.6% of home routers support one of these two protocols. Of those, 17.4% exhibit weak FTP passwords and 2.1% have weak Telnet passwords. In both cases, `admin/admin` is most common and accounts for 88% of weak FTP and 36% of weak Telnet credentials (Table 3.8). The credential is used by FTP devices from 571 vendors and from 160 Telnet vendors.

Regions vary in terms of vulnerable IoT device populations. In the smallest case, 14.7% of FTP devices in Western Europe support weak default credentials while more than 55% of FTP devices in Sub-Saharan Africa that are weak. A similar, though not as drastic range exists for Telnet. North America has the smallest vulnerable population of Telnet devices (0.5%), Central Asia and South America share the largest vulnerable Telnet population (4.9% of all IoT Telnet devices), primarily because of their reliance on surveillance devices, which have the weakest Telnet profile of all IoT devices.

Nearly all of the IoT devices that support FTP are work appliances (76%), storage (9.1%), media (7.6%), and surveillance devices (5.1%). Media and surveillance devices appear in the list due

Credential	%
admin/admin	88.3%
admin/	5.9%
Administrator/	1.4%
sysadm/sysadm	0.9%
root/	0.7%
root/root	0.4%
user/	0.4%
meo/meo	0.3%
admin/password	0.3%
admin/ttnet	0.3%
other	1.0%

(a) Weak FTP Credentials

Credential	%
admin/admin	35.6%
root/xc3511	16.0%
vodafone/vodafone	10.4%
guest/guest	7.8%
admin/1234	7.5%
root/hslwificam	3.9%
root/vizxv	3.7%
root/oelinux123	2.2%
admin/4321	1.8%
admin/_	1.6%
other	9.5%

(b) Weak Telnet Credentials

Table 3.8: **Most Popular Weak FTP and Telnet Credentials**—`admin/admin` accounts for the 88.3% and 35.6% of the weak FTP and Telnet credentials.

to their global popularity—unlike storage and work appliances where 29% and 23% of devices support FTP respectively, only 1% of media devices and 4% of surveillance devices support FTP. This aligns with the business need for work and storage devices to facilitate user file transfer, and also explains why there is little variance in the types of devices that support FTP across regions.

Storage devices are the device type most likely to support FTP, though only a small fraction of them use weak credentials. There are two regional exceptions—East Asia and Sub-Saharan Africa (Table 3.9), which exhibit 12.2% and 10.1% of storage devices with weak credentials respectively. We observe this is primarily due to one vendor, ICP Electronics, which has a large market presence in the two regions: 12.1% and 10.1% of storage devices in East Asia and Sub-Saharan Africa respectively. 74% of ICP storage devices exhibit weak default credentials.

A surprising number of home routers also support FTP (10.2%). TP-Link is responsible for the most routers with weak FTP credentials (Table 3.10)—regions that rely on TP-Link routers thus have a higher rate of devices with weak FTP credentials. Of all TP-Link devices across all regions, 9.3% offer an open FTP port, and 62.8% of those devices are protected by weak credentials.

Unlike FTP, there is little reason for any IoT devices to support Telnet in 2019. Yet, we find both that surveillance devices and routers consistently support the protocol. Surveillance devices have the weakest Telnet profile, with 10.7% of surveillance devices that support Telnet exhibiting weak credentials. This aligns with anecdotal evidence that suggests that these kinds of devices are easy to hack [102].

Region	FTP										Telnet						HTTP
	All IoT		Work Appl.		Surveillance		Router		Storage		All IoT		Surveillance		Router		TP-Link
	Vuln	Sup	Vuln	Sup	Vuln	Sup	Vuln	Sup	Vuln	Sup	Vuln	Sup	Vuln	Sup	Vuln	Sup	Vuln
North America	20.8	5.4	23.4	16.7	6.4	4.6	5.0	4.6	3.2	27.0	0.5	4.8	5.8	9.9	1.3	5.3	16.8
South America	39.0	7.4	42.0	27.8	13.1	2.9	11.9	9.3	4.8	25.9	4.9	8.6	18.9	16.6	1.6	13.2	42.3
Eastern Europe	31.6	9.9	40.7	30.9	9.8	5.8	16.2	12.6	6.6	31.2	3.0	8.9	9.3	19.4	2.3	20.9	48.9
Western Europe	14.7	6.5	23.6	19.9	7.2	5.1	4.4	7.4	5.5	26.4	1.0	4.2	8.1	7.5	2.1	3.3	23.6
East Asia	36.0	17.3	41.5	32.0	6.9	5.5	4.4	7.5	12.2	36.7	0.4	13.8	4.7	13.0	0.9	19.9	23.8
Central Asia	29.5	3.0	64.2	10.2	9.9	2.7	53.9	15.7	3.8	35.1	4.9	6.7	6.4	16.1	7.3	37.6	47.3
Southeast Asia	50.4	7.4	59.5	25.4	7.4	1.4	21.0	14.8	5.8	37.7	3.6	12.1	6.3	12.4	2.0	18.1	43.7
South Asia	33.7	13.4	38.6	36.6	5.4	2.4	6.8	11.1	4.2	35.4	2.9	14.6	7.6	13.7	0.9	19.3	21.4
Oceania	14.7	9.2	16.2	29.9	5.0	4.2	28.2	13.4	6.7	25.0	0.7	7.8	5.7	14.8	0.9	17.1	19.9
N. Africa, M. East	44.6	9.8	53.4	30.4	7.5	2.6	33.7	23.9	8.2	25.9	4.8	11.1	10.5	17.3	1.7	26.6	24.0
Sub-Saharan Africa	55.3	15.4	61.5	27.2	10.8	5.1	23.6	12.5	10.1	35.4	1.1	12.0	5.2	14.1	1.6	20.9	25.4

Table 3.9: **Weak Default Credentials by Region and Device Type**—We show the weak FTP and Telnet device population by region and device type, highlighting both the fraction of devices that support (Sup) each protocol as well as the fraction that are vulnerable with weak default credentials (Vuln). Some regions have a higher fraction of devices with weak credentials—in the largest case, 50% of FTP devices in Southeast Asia and 4.9% of all Telnet devices in Central Asia are weak. We further observe that the likelihood of having weak FTP credentials is correlated to weak Telnet credentials, indicating that the presence of weak credentials may be linked to weaker security posture in the region overall.

Vendor	% Open	% Weak	% of Weak
Ricoh	92.1%	71.2%	29.8%
Kyocera	91.7%	97.1%	26%
HP	7.3%	92.4%	24.5%
Sharp	89.4%	94.2%	6.4%
Canon	2.7%	79.3%	2.1%

(a) Work Appliance (FTP)

Vendor	% Open	% Weak	% of Weak
TP-Link	9.3%	62.8%	55.9%
Technicolor	22.9%	20.4%	9.6%
ZTE	9.9%	37.5%	9.5%
MicroTik	46.9%	13.0%	5.3%
D-Link	16.2%	10.9%	3.9%

(b) Router (FTP)

Vendor	% Open	% Weak	% of Weak
D-Link	38.9%	6.1%	33.0%
Huawei	13.6%	4.8%	18.7%
TP-Link	15.0%	1.4%	12.6%
Zyxel	53.5%	2.9%	12.1%
Intelbras	12.7%	26.4%	7.1%

(c) Router (Telnet)

Table 3.10: **Weak Vendors by Device Type**—We show the vendors that exhibit weak default credentials across each device type in our dataset sorted by the fraction of weak devices they contribute to their respective device types. For example, 71.2% of Ricoh printers that support FTP also support weak default credentials, and these make up 29.8% of all weak work appliances.

3.4.2 Home Routers

Nearly every home in our dataset has a home router. Similar to most types of IoT devices, there are regional differences and a long tail of vendors globally (Table 3.9). In total, we see home routers from 4.8 K vendors. TP-Link is the most popular manufacturer globally (15% of routers) and is the top provider in five regions: South America, Central Asia, Eastern Europe, South Asia, and Southeast Asia. Arris is the most popular router vendor in North America (16.4%)—likely because popular ISPs like Comcast supply Arris routers to customers. Huawei is the most popular vendor in Sub-Saharan and North Africa, accounting for 19.8% and 25.6% of all routers respectively.

Weak FTP/Telnet Credentials More than 93% of routers have HTTP administration interfaces on port 80. We also find that many routers support DNS over UDP (66.5%), UPnP (63.4%), DNS over TCP (42.1%), HTTPS (42.2%), SSH (19.7%), FTP (10.8%), and Telnet (14.6%). Of the devices that support FTP and/or Telnet, 12% have weak FTP and 1.6% have weak Telnet credentials. 1.2% of *all* routers exhibit a weak FTP credential and 0.2% exhibit of all routers have a weak Telnet credential. For FTP, TP-Link routers had the weakest profile: 55.3% of their routers with an open FTP port exhibited a weak credential. For Telnet, D-Link routers were the weakest—6% of all open routers had a weak credential, and 35.3% of all D-Link routers had an open Telnet port. We show a breakdown by region in Table 3.10.

Weak HTTP Administration Credentials WiFi Inspector attempts to login to the HTTP interfaces for devices from a small number of common vendors, including TP-Link—the most common router manufacturer. In our dataset, there are 3.8 M TP-Link home routers, of which 82% have an HTTP port open to the local network. WiFi Inspector was able to check for weak default credentials on 2.5 M (66%) of the devices with HTTP. Overall, 1.2 M (30%) of TP-Link routers exhibit weak HTTP credentials. Nearly all (99.6%) use `admin/admin`. The number of TP-Link routers with guessable passwords varies greatly across regions (Table 3.9). For example, only 6% of TP-Link routers in North America have weak passwords while around 45% do in South and Central Asia, and East and South Europe.

External Exposure To understand whether routers with weak default credentials are also exposed on the public Internet, we joined the WiFi Inspector dataset with Internet-wide scan data from Censys [106] for devices on a single day—January 30, 2019.[7] A small number of home routers host publicly accessible services: 3.4% expose HTTP, 0.8% FTP, 0.7% Telnet, and 0.8% SSH. Open gateways are primarily located in three regions—Central America (29.3%), Eastern Europe

[7] We perform this analysis for January because of GDPR restrictions on Avast data.

(20.6%), and Southeast Asia (17.2%). Of routers that are externally exposed, we find that 51.2% of them are exposed with a vulnerability—far higher than the fraction non-externally available routers in our dataset with a weakness or vulnerability (25.8%). The most popular router vendor in these regions is TP-Link, which is also the vendor responsible for the most externally exposed routers (19.7%). We note this is not simply because TP-Link is the largest vendor—a proportion test across regions shows that TP-Link routers appear in the set of externally exposed routers at a higher rate than that of non-externally exposed routers.

3.4.3 Scanning Homes

While scan data can provide insight into the vulnerability of hosts, it typically does not indicate whether hosts have been compromised. We analyzed the homes from WiFi Inspector that were seen performing vulnerability scans in a large network telescope (Section 3.2) on January 1, 2019 to better understand infected devices. Of the 500.7 K homes that WiFi Inspector collected data from that day, 1,865 (0.37%) homes were found to be scanning for vulnerabilities. Scans most frequently target TCP/445 (SMB, 26.7% homes) followed by TCP/23 (Telnet, 11.3%), TCP/80 (HTTP, 10.7%), and TCP/8080 (HTTP, 9.4%). In addition to checking credentials, WiFi Inspector also checks devices for a handful of recent, known vulnerabilities (CVEs, EDBs, and others). 1,156 (62%) of scanning homes contained at least one known vulnerability—conversely, 7.2 M (46.8%) non scanning homes in our dataset contain at least one known vulnerability. To test the differences between these populations, we used a proportions t-test at a confidence interval of 95%. We observe that the two sets are statistically significantly different (p-value: $2.31 * 10^{-39}$), indicating that scanning homes have a higher vulnerability profile than homes globally. This trend also holds for the number of vulnerable devices in scanning homes (9.7%) compared to homes globally (5.7%). Unfortunately, we were unable to determine why homes without known vulnerabilities were seen scanning. This is likely due to devices being compromised through means outside of our measurement vantage point, for example, vulnerabilities that we do not test for.

Although the overall vulnerability profile of devices in scanning homes is higher, this is not true of all specific vulnerabilities. Of the 25 vulnerabilities observed in scanning homes, 17 appeared at a ratio that was not statistically significantly different than devices globally. The remaining eight vulnerabilities were statistically significantly different, though six appear at a *smaller* rate in scanning homes than globally. The two vulnerabilities that appeared at a higher rate in scanning homes were both related to EternalBlue—a leaked NSA exploit targeting SMB on Windows that was primarily responsible for the WannaCry outbreak that impacted millions of Windows devices in 2017 [117]. Specifically, we identify 5.2% of devices within scanning homes that are vulnerable to EternalBlue, and further, 1.3% of devices in scanning homes are *already compromised*, and

communicating through a backdoor. This additionally explains some fraction of the SMB scanning we observed on the darknet, as machines compromised via EternalBlue often scan for other hosts running vulnerable SMB servers. We note that although these homes contain vulnerable devices, we cannot claim that they are scanning as a result of these devices—for one, we do not have full vulnerability coverage, and two, it is an outstanding challenge to attribute device behavior from our vantage point. Still, the presence of any scanning homes in general indicates a threat landscape larger than simply publicly accessible devices, and one that should be considered by the security community.

3.5 DISCUSSION

There are some immediate next steps. As outlined in Section 3.4, much of the devices that support weak credentials are manufactured by a handful of popular vendors across all regions (Table 3.10). The security community can start addressing these challenges by encouraging the largest offending vendors to adopt better security practices. On the policy end, law enforcement and legal entities have started to provide legal disincentives for weak security practices. In light of the Mirai attacks, the U.S. Federal Trade Commission has prompted legal action against D-Link [118] for putting U.S. consumers at risk.

A larger question remains on how to address the long tail of vendors. As described in Section 3.3, regions often have vastly different preferences for vendors across device types. As a result, working to improve the security of devices based solely on the global distribution may inadvertently leave smaller regions with divergent preferences less secure.

Finally, it is not immediately clear how to measure the impact of compromise on home security. In our work, we measured the prevalence of scanning, though this is just one indication of compromise. Furthermore, we only observed 0.37% of homes scanning; amounting to only 1.8 K homes on a single day. In spite of all the data collected within homes in this paper, we could not effectively identify why certain homes were compromised. Researchers have proposed systems to enable auditing of home IoT setups [99, 119], but there is still more work to be done.

3.6 RELATED WORK

Our work builds on research from a number of areas, primarily in home network measurement and IoT security.

Home Network Measurement Early research in home network measurement primarily focused on debugging networks—projects like Netalyzer [120] were conceived to enable users to debug their home Internet connectivity [121, 122, 123]. A number of follow on papers leveraged Netalyzer-like scans to investigate the state of devices in homes [103, 121, 124], as well to try and understand the implications of a connected home on user behavior [125].

Most similar to our work is presented by Grover et al. who installed home routers with custom firmware in 100 homes across 21 countries to measure the availability, infrastructure, and usage of home networks [126, 127]. Their work focuses on the network properties of home networks on aggregate, and also is able to measure networks continuously based on their position in the network. Our work instead focuses on the *devices* behind the NAT in their ubiquity and their security properties, with particular attention spent on IoT devices.

Recent work has built off of network scanning to enable rich device identification. Feng et al. built a system that leverages application layer responses to perform device identification without machine learning, similar to our hand curated expert rules [14]. This work has built off a number of papers that leverage banners and other host information to characterize hosts [128, 129, 130, 131, 132]. Other rule based engines have been used in other work on active, public scan data based on probing for application banners [102, 106].

Home IoT Security Home IoT security has been of recent interest to researchers in light of its growing security and privacy implications, from the systems level up through the application layer. Ma et al. investigated the rise of the Mirai botnet [102], which was largely composed of IoT devices compromised due to weak credentials and used to launch massive DDoS attacks. This is not isolated to only attackers—researchers have been breaking the home IoT devices since their conception [6, 133, 134, 135**?**]. Notably, Fernandes et al. outlined a number of challenges in Samsung SmartThings devices, from their access control policy to their third-party developer integration [94]. In response, researchers have built systems to enable security properties in home IoT, such as information flow tracking and sandboxing [96, 97], improving device authentication [136], and enabling auditing information [99, 119]. Most recently, Alrawi et al. synthesized the security of home IoT devices into a SoK, where they present a systematization of attacks and defense on home IoT and outline how to reason about home IoT risk [137].

Internet-Wide IoT Scanning There has been a wealth of recent work that has used Internet-wide scanning for security analysis, including analyzing embedded systems on the public Internet (e.g., [8, 102, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147]). In contrast to these works, we focus on devices inside of homes that are not visible through Internet-wide scanning.

3.7 CONCLUSION

In this chapter, we conducted the first large-scale empirical analysis of IoT devices on real-world home networks. Leveraging internal network scans of 83M IoT devices in 16M homes worldwide, we find that IoT devices are widespread. In several regions, the majority of homes now have at least one networked IoT device. We analyzed the types and vendors of commonly purchased devices and provided a landscape of the global IoT ecosystem. We further analyzed the security profile of these devices and networks and showed that a significant fraction of devices use weak passwords on FTP and Telnet, are vulnerable to known attacks, and use default HTTP administration passwords that are left unchanged by users. We hope our analysis will help the security community develop solutions that are applicable to IoT devices already in today's homes.

CHAPTER 4: LIMITATIONS OF EXTERNAL SCANNING

4.1 INTRODUCTION

In the previous chapter, we showed how IoT devices are now staple, end-user products in many regions. We showed that two thirds of North American homes contain at least one IoT device [148] and that IoT devices are dispersed globally. The research community has spent much time investigating home devices, analyzing device identification [14, 129, 132, 149, 150, 151], understanding device security and privacy [94, 98, 102, 137], and enumerating their networking capabilities [15, 148, 152]. However, there has been little investigation into how best to study and measure the IoT ecosystem. Prior research leveraged a multitude of measurement techniques and vantage points to investigate home IoT devices, leaving researchers with a fractured and sometimes disparate view of the ecosystem taken from different measurements.

Two competing vantage points have primarily driven IoT measurements at scale. The first is Internet-wide scanning, as we employed in Chapter 2, and that others have used to perform device identification and enumerate vulnerable devices [14, 102]. The second is private network scanning, which we leveraged in Chapter 3, and that others have used to capture the security and privacy risks of home IoT devices [15, 152]. Still, little is known about how these two vantage points compare with each other, specifically with regards to their IoT populations. How different are the IoT populations between external scanning and internal scanning, by device type, services offered, and scale?

In this chapter, we investigate these questions by comparing two vantage points: active, home network scans through a collaboration with Avast, and active external scans of the public IPv4 space, drawn from Censys [9]. In this context, active scanning means probing devices for their capabilities, through a SYN scan and application layer handshake. We collect scans from both datasets from a single day, May 2 2020, spanning 6.1 M IoT devices connected publicly and 1.8 M IoT devices connected locally, sourced from 941 K homes.

Internet-wide scanning and internal home scanning offer different distributions of IoT devices. Although routers are the most prevalent from both vantage points, device distributions internally are diverse, containing media devices (20.7%), work appliances (6.7%), and camera (3.4%). Conversely, IoT devices connected publicly skew mainly towards routers (92%), with smaller fractions of storage devices (4.3%) and media devices (1.5%). Many of the devices inside home networks that appear at a high frequency (e.g., media devices and cameras) have historically been vulnerable to security risks [] and have been used in high profile attacks (Chapter 2).

The services offered by devices also differs between the two vantage points. For example,

71.8% of IoT devices inside home networks have an open 80/HTTP port for remote communication, compared to only 15.8% of IoT devices connected publicly. Instead, externally connected devices are more likely to support 7547/CWMP, a router protocol; in part due to the skew in our dataset towards routers, and 443/HTTPS for secure communication via HTTP. Services offered by devices also differ between the two vantage points. In some cases, the external perspective of IoT devices under-reports the services that are available internally. For example, MQTT, a popular IoT control protocol, appears on near zero publicly accessible devices but 1.5% of devices internally, highlighting that relying only on certain vantage points may skew our understanding of the ecosystem.

Our results highlight the potential biases in measurements of home IoT devices and underscore the need for holistic measurements as the ecosystem continues to develop.

4.2 BACKGROUND AND RELATED WORK

Our work compares and contrasts several measurement techniques for understanding real-world IoT deployment. In this section, we describe the prior work in the space and the techniques used.

4.2.1 Active Measurement

There is a long history of researchers using active scanning for network measurement through tools like nmap [153]. Recently, active measurement has been used to measure a wide variety of topics using tools like ZMap [8], and Massscan [11]. These tools dramatically reduced the time needed to discover open services on the public Internet, and researchers have used to understand the impact of natural disasters [8], track high profile vulnerabilities [154], and understand attacker behavior [102]. Most related to our work, several studies have used scanning to uncover security problems in embedded systems and IoT devices [138, 139, 143, 146, 147]. Active measurement forms the foundation of search engines like Censys and Shodan [9].

4.2.2 Home IoT Measurements

Our work draws on early home network measurement infrastructure, which primarily focused on debugging home networks. Early projects like Netalyzer [120] helped users debug their home networks [121, 122, 123] and were subsequently used to measure device distributions in homes around the world [103, 121, 124]. Recent work in home IoT measurement has also leveraged active and passive measurement data to perform IoT device identification [128, 129, 130, 131, 132]. Recent tools, like IoT Inspector [15] (which we leverage in this work) offer users the ability to inspect their home networks for potential security and privacy concerns. Other recent work in home

IoT networks has primarily focused on giving users finer grained control over device to device communication and using SDN to enforce network policy [16, 17, 132, 155].

4.3 METHODOLOGY

Our study leverages measurements of IoT devices from both an internal and external vantage point. In this section, we detail our collection mechanisms for each data perspective and provide each source.

4.3.1 Active Internal Scans

We collect active, internal scans of home networks by partnering with Avast, a security software company that provides popular antivirus and consumer security software products. Avast estimates that their software runs on 160 M Windows and 3 M-Mac OS computers and accounts for 12% of the antivirus market share. Avast products contain a tool, called WiFi Inspector, which enables users to inventory their local network. To do this, WiFi Inspector collects all entries from the local computer's ARP table and subsequently probes devices over common TCP/UDP ports to detect listening services. It then performs device identification, using a combination of regular expression based rules and machine learning based on banners collected while probing each device and DHCP hostnames. In addition, Avast also stores the device type of each device found on the internal network, using a combination of regular expression based rules and supervised machine learning. We collect active, internal scan data for a single day—May 2, 2020, which amounts to 941 K homes and 1.8 M IoT devices inside of those homes.

4.3.2 Active External Scans

We collect active, external scan data from Censys [9], a company that performs active Internet-wide scans built onto of ZMap [8]. Censys scans the public, IPv4 space for many popular protocols [156] over both TCP and UDP and performs an application layer handshake if ports are available, for example, making an `HTTP GET /` request if the IP address is available on port 80. We collect data from the same 24 hour period on May 2, 2020, to control for potential DHCP churn when comparing to our active internal scan dataset. In total, our active external scan dataset contains scans for 122 M IP addresses which were each scanned on 49 different popular protocols, such as FTP, Telnet, SSH, and HTTP.

Port	Protocol	Avast Rank (%)	Censys Rank (%)
80	HTTP	1 (30.16%)	1 (46.57%)
445	SMB*	2 (25.82%)	20 (1.01%)
53	DNS	3 (14.22%)	7 (5.85%)
443	HTTPS	4 (12.35%)	2 (38.31%)
8080	HTTP	5 (6.36%)	6 (6.14%)
22	SSH	6 (4.97%)	4 (15.91%)
631	IPP	7 (3.79%)	24 (0.23%)
23	Telnet	8 (3.49%)	17 (2.38%)
21	FTP	9 (3.04%)	5 (9.07%)
7547	CWMP	10 (1.65%)	3 (19.34%)
8888	HTTP	11 (1.31%)	18 (2.11%)

Table 4.1: **Port Distributions in Aggregate**—Services as they appear from an external perspective do not always reflect the distribution of services inside home networks.
* We note that `445/SMB` appears highly on this list due to Avast's customer base, which skews towards Windows users.

4.4 COMPARING EXTERNAL AND INTERNAL VANTAGE POINTS

Active scanning of the public IPv4 space has served as a mechanism in much IoT measurement, such as device type attribution for security incidents [102] and building IoT device signatures [14]. However, many devices are obscured behind NATs, and scanning the public Internet does not provide a full picture of IoT devices. As a result, IoT research based on only the public IPv4 Internet is skewed towards devices that appear readily on the public Internet. In this section, we detail findings in comparing the device distributions and server capabilities of devices through an external vantage point (through Censys) and an internal vantage point (through Avast).

4.4.1 Comparing Port Distributions in Aggregate

To begin we simply compare the active services across all devices found on the public internet versus devices found inside home networks (Table 4.1). In aggregate, Avast and Censys share 49 ports that are scanned for active services over both TCP and UDP. We rank order these services by the fraction of devices that support each service and compute a Spearman's rank correlation test between the two lists. We find a correlation of $\rho = 0.61$, indicating a medium correlation between the two lists per Cohen's guidelines [107].

Some top services are shared across both the external and internal vantage points, for example, `80/HTTP` is the top service for both vantage points, however, they make up a much larger fraction of external hosts than internal hosts (46.6% vs. 30.2%). This is likely because many externally

Protocol	% IPv4 Responded	% IPs Labeled
7547/CWMP	19.34%	16.29%
21/FTP	9.07%	6.79%
22/SSH	15.91%	2.38%
23/Telnet	2.38%	20.62%
443/HTTPS	38.31%	3.71%

Table 4.2: **Active IoT Labeling**— We leveraged five protocols for identifying devices on the public IPv4 space. We show the fraction of IPs that support each protocol and the fraction of those IPs that we could ultimately label. We identified 6.1 M IoT devices on the public Internet. Devices that support Telnet and CWMP were the most descriptive in their banners, while devices that support SSH were the least descriptive.

connected devices are web servers that host content for users to browse to, whereas HTTP is typically only used inside local networks for remote configuration or remote control of devices. Conversely, some services appear much higher inside home networks than others—IPP (Internet Printing Protocol) appears on 3.8% of devices inside home networks, however, only appears on 0.2% of devices publicly. Some protocols, however, are near equivalent in both vantage points—8080/HTTP, 23/Telnet, and 8888/HTTP are some notable examples—highlighting some similarities between the two.

4.4.2 Collecting Publicly Available IoT Devices

Although port distributions vary across the external and internal perspectives, it is not particularly surprising. Naturally, the distribution of devices inside homes vastly differs from the distribution of devices outside them—for example, most homes do not have servers, whereas many publicly connected computing devices are servers. To address this shortcoming of comparison, we instead focus only on the IoT devices that are connected both internally and externally. We thus build a subset of active hosts that we can identify as IoT devices connected to the public Internet. Although several studies have leveraged active, external scan data for IoT device identification [14, 102, 151], we could not locate any freely available tools to perform this task. In lieu of a de facto solution for device identification, we implemented the technique described in Ma et al. which leverages banner data from five protocols—CWMP, FTP, SSH, Telnet, and HTTPS—to make predictions about device types [102]. To do this, we primarily utilize the regular expression database provided by the Nmap project [34], which has curated over 3000 regular expressions on these protocols with labeled banner data. Ma et al. augmented this dataset with labels manually curated for those found to be scanning in the Mirai botnet, particularly via HTTPS and CWMP, which we were able to obtain from the researchers. We note that Nmap splits their identification into 28 device types [34],

Port	Protocol	Avast Rank (%)	Censys Rank (%)
80	HTTP	1 (71.88%)	3 (15.84%)
53	DNS	2 (36.57%)	8 (2.55%)
443	HTTPS	3 (30.18%)	2 (26.7%)
8080	HTTP	4 (14.47%)	7 (5.85%)
445	SMB*	5 (12.57%)	11 (0.44%)
22	SSH	6 (12.17%)	6 (7.12%)
631	IPP	7 (9.92%)	12 (0.35%)
23	Telnet	8 (8.31%)	5 (9.23%)
21	FTP	9 (7.44%)	4 (11.56%)
7547	CWMP	10 (3.9%)	1 (59.29%)
8888	HTTP	11 (2.42%)	14 (0.27%)
8883	MQTT	12 (1.55%)	37 (0.0%)

Table 4.3: **Port Distribution Comparisons for IoT Devices**—We show the most prevalent services offered by IoT devices in our Avast and Censys datasets. Services as they appear from an external perspective do not always reflect the distribution of services inside home networks.
* We note that `445/SMB` appears highly on this list due to Avast's customer base, which skews towards Windows users.

however, Avast only contains 11 IoT device types. We aggregate Nmap's device types into the device types that Avast uses, combining for example "broadband router" and "router" into "router".

We compiled these rules into a device identification tool that runs on active data collected by Censys to build our IoT dataset. In the context of this work, we define an IoT device as *not* a general purpose computing device (e.g., a computer or server) or a mobile phone (e.g., iPhone or Android device). Table 4.2 shows both the fraction of IPs that support each protocol we investigated as well as the fraction that we could label. In sum, our Censys data consists of 114 M IP addresses, of which we were able to identify 5.3%, or 6,053,986 as an IoT device. We will release our dataset and device identification code as an open source package at publication time.

4.4.3 IoT Server Capabilities

Externally available IoT devices offer different services than those connected inside homes. Avast and Censys both scan devices for a common 44 ports that range over TCP and UDP for each respective IoT dataset. We rank order these services by the fraction of total devices that offer each service and compute a Spearman's rank correlation test, which is a non-parametric test used to identify the correlation between two ranked lists. We find a correlation of $r = 0.73$, indicating a medium correlation between the two lists per Cohen's guidelines [107], highlighting that although the lists are similar, they are not identical.

Device Type	% Internal Devices	% External Devices
Router	61.9%	92.3%
Media	20.7%	1.5%
Work Appliances	6.7%	0.7%
Camera	3.4%	0.6%
Generic IoT	1.1%	0.4%
Storage	0.9%	4.3%

Table 4.4: **Device Type Distributions**—Device type distributions internally and externally vary. Notably, internal networks contain more media devices (20.7%), work appliances (6.7%), and surveillance devices (3.4%) than the outside world.

The most commonly offered services inside homes are those that skew towards home-network services. Table 4.3 shows the services from Avast that appear on at least 1% of devices, and their corresponding rank in the Censys dataset. Many of the top active services are shared—for example, `80/HTTP`, `53/DNS`, and `443/HTTPS` all appear in the top 10 services for both perspectives. In some cases, the services that differ in popularity can be explained by their context. For example, `631/IPP`, Internet Printing Protocol, is more prevalent inside of homes (9.9%) than outside of homes (0.35%), as there are more printers within home networks than there are connected to the outside world. In a similar case, `8883/MQTT` is commonly used by companion apps to control and communicate with smart home IoT devices and is far more prevalent inside homes (1.55%) than outside of homes (0.0%). MQTT is associated with multiple vulnerabilities [157], and despite its low prevalence in external scans, its prevalence inside homes warrants closer attention from the research community.

4.4.4 Comparing Device Type Distributions

IoT device type distributions differ between the external and internal vantage points. Table 4.4 shows the fraction of devices that each device type accounts for in both our internal and external datasets. In both cases, routers are the most common device, accounting for 61.9% of devices inside homes and 92.3% of tagged devices on the public Internet. However, other device types are more prevalent inside homes, such as media devices (20.7%), work appliances (6.7%), and surveillance devices (3.4%). All of these device types appear on a smaller scale on the public Internet.

An important limitation of our IoT device identification on Censys data is that it relies on specific regular expressions over a fixed set of protocols. As such, device distributions taken in aggregate are not a random sample of IoT devices connected to the public Internet, but rather a biased perspective based on both the protocols scanned and the regular expressions used. For example, our collected IoT dataset is heavily skewed towards routers, in part because they are often easier to identify, but also because one protocol we scan for is `7547/CWMP`, which is typically only offered by routers. To

account for this bias, we analyze and compare device type distributions per protocol we used for active device identification. Table 4.5 shows the device distributions for both external and internal probes accounting for all protocols scanned. In our discussion, we explicitly exclude `7547/CWMP` as it only encompasses routers.

FTP ($r = 0.7$)		**SSH** (r **not presented**)		**Telnet** ($r = 0.7$)		**HTTPS** ($r = 0.97$)	
Internal	External	Internal	External	Internal	External	Internal	External
Router (73%)	Router (95%)	Router (88.3%)	Router (97.9%)	Router (82%)	Router (94.1%)	Router (73.3%)	Router (63%)
Work (20.2%)	Storage (2.3%)	Storage (4.2%)	Gen. IoT (0.2%)	Work (13.4%)	Work (3.7%)	Work (19.9%)	Storage (24%)
Storage (4.5%)	Camera (1.7%)	Work (3.2%)	–	Camera (2.7%)	Media (2.5%)	Storage (3.4%)	Media (8%)
Camera (1%)	Work (0.7%)	Camera (2%)	–	Media (2.7%)	Gen. IoT (0.2%)	Media (1.6%)	Camera (2.8%)
Media (1%)	Media (0.4%)	Media (1.4%)	–	Storage (0.2%)	Cam (0.02%)	Camera (1.3%)	Gen. IoT (2.2%)

Table 4.5: **Device Type Distributions per Protocol**—We show device type distributions split per protocol both within home networks and through active external scans. Distributions broadly differ, with home networks containing a larger fraction of media devices and work appliances, and external networks containing a larger fraction of storage devices. Notably, labeling devices via HTTPS proves the most descriptive in capturing a wide spread of IoT devices connected publicly.

FTP. Device type distributions for devices that support `21/FTP` are moderately correlated, with a Spearman rank correlation of $r = 0.7$. For external devices, we find that 9% of public IP addresses externally have an open FTP services at port 21, of which we could label 6.8% as a type of IoT device. These were mainly comprised of routers (95%), storage devices (2.3%), and cameras (1.7%). Devices that supported FTP on internal networks were also primarily routers (73%), however, work appliances (e.g., printers) were far more prevalent than they appeared externally, accounting for 20.2% of devices. Both perspectives shared a small fraction of camera and media devices, likely indicating these types of devices are unlikely to host an FTP server.

SSH. Devices that support `22/SSH` were the least descriptive in our external dataset—we were only able to identify 2.4% as an IoT device. These were primarily router devices (97.9%), which aligns with the internal perspective (88.3%). Unfortunately, we were unable to identify many other devices through SSH, indicating the protocol is not useful in measuring IoT devices on the public Internet using NMap regular expressions alone. Because of a lack of a full distribution, we do not present a correlation value for SSH.

Telnet. The smallest fraction of externally connected devices supported `23/Telnet` (2.4%), however, it was the most descriptive protocol—we were able to identify 20.6% of devices as an IoT device. Device type distributions were moderately correlated, with a Spearman rank correlation of $r = 0.7$. Again, routers are the most prevalent IoT device available in both internal (82%) and external (94.1%) networks. Unlike devices that support FTP, however, we find many media devices (2.5%) and work appliances (3.7%) connected to the public Internet that support Telnet. Given the prevalence of media devices and work appliances inside home networks, Telnet remains a useful protocol in capturing a diverse set of externally connected IoT devices.

HTTPS. The most diverse set of IoT devices comes from `443/HTTPS`. 38.3% of external host support HTTPS, of which we labeled 3.7% as an IoT device. Device type distributions were the strongest amongst all protocols, with a correlation of $r = 0.97$ to the distribution found inside home networks. Although routers are still the most prevalent device, they account for a smaller fraction of devices (63%) compared to both internal devices that support HTTPS (73%) and all other protocols externally. Instead, we find a larger fraction of networked storage devices (24%), media devices (8%), and cameras (2.8%) connected to the public Internet and available over HTTPS. We find `443/HTTPS` to be a particularly useful protocol in measuring IoT devices on the public Internet, especially as more devices rely on HTTPS for remote management services.

Port	Protocol	Fraction Devices
7547	CWMP	37.51%
80	HTTP	25.38%
443	HTTPS	17.76%
53	DNS	9.37%
8080	HTTP	8.85%
161	SNMP	8.77%
22	SSH	7.62%
21	FTP	5.18%
23	TELNET	4.24%
3389	RDP	1.75%

Table 4.6: **Externally Available Ports for Routers**—We find 6% of the homes scanned from our single day snapshot are externally available on at least one service and serve a variety of ports and protocols. CWMP is the most prevalent protocol, appearing on 37.5% of the externally available homes and indicating that many such devices are routers.

4.4.5 Homes from Public Scanning

In our dataset, approximately 56,550 (6%) of homes scanned from our single day snapshot were shared between Censys and Avast, meaning we obtained scan data from both the external perspective and the internal perspective. We expect many of these devices to be routers or gateways. To verify this hypothesis, we enumerate the distribution of services available on these machines in Table 4.6. The top service is `7547/CWMP`, a router management protocol, followed by `80/HTTP`, `443/HTTPS`, and `161/SNMP`, all of which are typically used routers to manage local networks.

However, routers simply serve as an entry point to networks which will typically contain other devices; we identify a median 4 devices per home in the intersection. To provide a lower bound estimate for the number of in-home devices that are obscured from active external scanning, we compute a simple calculation based on the number of externally available routers. Specifically, we start with the number of devices that externally support `7547/CWMP`, as this is a strong signal these devices are routers. These devices account for 22 M devices through active, external scans. We then scale this figure by the fraction of routers that are *not* externally exposed (94%), as well as the fraction of routers that do not support CWMP (62.5%), drawn from our measurements from Avast. This provides us with an estimate of the number of devices that support CWMP on the public Internet. Scaling this value by 4 (the median number of devices per home), we estimate that 3.9 B home devices are obscured from measurement through active external scanning. This largely eclipses the number of hosts that are available through a daily scan (115 M). Again, we note that this is a lower bound, as it only takes into account devices that support CWMP, and furthermore assumes that the distribution of homes from Avast is identical to a global home distribution. However, it

does underscore the need for deeper measurement into home networks as they continue to mature.

4.4.6 Limitations of Active Scanning

Our active measurements are not without limitations. Both our external and internal IoT device samples are biased due to their specific measurement mechanism. For example, `445/SMB` appears at rank 5 for Avast, only because their customer base skews towards Windows users. Similarly, available service comparisons are limited to those scanned by Censys—Avast scans for over 200 ports over TCP and UDP for which Censys scans are simply a subset. Notably, Censys does not scan for some protocols that are prevalent on local networks, like `1900/UPNP`, which is supported by 37.2% of devices inside of homes.

There is also a concern that we could not perform the same device identification techniques for in-home devices and external devices, potentially leading to additional skew in device distributions. At best, we restricted our analysis to scans done on specific ports, however, this remains an outstanding limitation of this work. Even considering this, identification for protocols were able to capture a diverse set of IoT devices, for example, those that had an available service over `443/HTTPS`, and can prove useful in leveraging external scan data as a proxy for home IoT measurement.

4.5 DISCUSSION AND CONCLUSION

In this chapter, we combined and compared two measurements of home IoT devices: active measurements of 6.1 M IoT devices collected from the public IPv4 space and active measurements of 1.8 M IoT devices within 941 K homes. We identified the differences in IoT populations between active external scanning and active internal scanning and provided data on device type distributions and offered services.

Our results echo a sentiment present in many aspects of Internet measurement—relying on a single data perspective alone may present a biased view of the measured ecosystem. To combat this, we complemented the measurements from an external perspective with an internal perspective, and observed relying only on the external perspective would under-report the prevalence of certain types of IoT devices, such as medias and cameras, many of which have historically posted security and privacy risks [5, 102]. Furthermore, external scanning under-reports several IoT critical protocols (such as MQTT, which has historically caused security issues), highlighting that relying only on the external perspective may skew our understanding of the prevalence of these services and devices. Finally, to further reduce the bias in measuring device distribution, we investigated IoT device type distributions for each protocol independently. We found that device identification through HTTPS provides the largest diversity in IoT device types and most closely resembles devices found in home

networks (Table 4.5), highlighting one area where external scanning may be useful as a proxy for IoT measurements at large.

Our measurement methodology, however, is not without limitations. The IoT population surveyed through active, external scanning is limited to only devices that have been tagged with specific regular expressions, either by us or through the NMap project. As a result, our IoT device population is inherently biased by the devices to those that are previously tagged, and likely misses newer IoT devices. It is thus challenging to conclude that the distributions presented here are absolute ground truth—rather, they serve as one comparison using current state of the art device identification techniques. Several device identification techniques that leverage machine learning on local networks have been reported in the last five years [148, 150]; one direction for future work would be to leverage models trained on SYN traffic from known in-home IoT devices and apply it to the public Internet to augment current regular expression based techniques.

Our measurements highlight the need for diverse vantage points when building IoT-focused systems. We hope our results will prove useful to future researchers as the IoT ecosystem continues to develop.

CHAPTER 5: MEASURING HOME IOT BEHAVIOR WITH PASSIVE OBSERVATION

5.1 INTRODUCTION

Our analysis of home IoT devices has thus far relied on *active scanning*, meaning we probe devices for their server capabilities and use the resultant data for analysis. However, IoT devices perform many tasks, such as device discovery, capability enumeration [**?**], and even relaying home automation events. Thus, beyond their capabilities as server devices, IoT devices are also clients on the network. Previous work has investigated IoT client behavior, with the focus on privacy [15]. Specifically, Huang et al. built IoT Inspector [15], which observes communication between IoT devices and the outside world. It provides users with a focused breakdown of the potentially privacy sensitive information that IoT devices send to ad networks and third-party data aggregation services. It does this by leveraging *passive observation*—IoT Inspector serves as a voluntary man-in-the-middle and records traffic between devices and the gateway. However, their system focuses specifically on communication outside the network—to the best of our knowledge, no one has studied what happen *inside* the NAT of smart home IoT networks.

In this chapter, we analyze the traffic of 275 smart homes over a three week period through an instrumented version of IoT Inspector. We augment the system by observing communication between *all* devices, not just between a single device and the gateway. This enables us observe IoT traffic in context, and characterize IoT device behavior today. In addition, we add active scanning to IoT Inspector, which we then leverage to compare the passive vantage point from the active point, showing that a single measurement technique alone cannot holistically capture the behavior of IoT devices.

Devices regularly communicate with other local devices: 65% of inspected devices communicated with at least one other device inside the network. Device networking behavior varies by device type: for example, for example, smart home devices communicate with a median 12% of devices on the network, while storage devices communicate with a median 56% of devices. Similarly, certain device types communicate frequently with other device types: smart home devices most frequently communicate with home assistants (48% of communication) while media devices most frequently communicate with other media device (51%).

Devices inside home networks support a host of services and protocols—73% of devices inside home networks offer at least one active service via TCP. Many of these services are used to run common services over bespoke ports determined by device type and manufacturer: 44% of ports used in communication are specific to a single manufacturer and device type. For example, Sonos speakers use port 1400 to host an HTTP administrator interface, and Google Home products use

port 8008 to receive HTTP-based commands. The lack of consistency in the networking services of home IoT devices complicates active measurements, which often rely on a priori knowledge of open ports to scan (e.g., the Censys scan list [156]).

Most alarming is that devices rarely use all services they offer: devices only use an average of 19% of services offered in normal operation. Worse, many unused services are security critical, such as `22/ssh` (12.5% of devices), `23/telnet` (2.3% of devices) and `111/rpcbind` (2.3%), increasing the attack surface of devices without offering new functionality. Relying only on active measurements may over-report the prevalence of certain ports and protocols even when they are rarely used.

Our results highlight potential biases in measurements of home IoT devices and underscore the need for holistic measurements as the ecosystem continues to grow. We will make our device-to-device measurements available to researchers and hope that our results will help to inform the research community as we continue to build tools to measure real-world IoT devices at scale.

5.2 BACKGROUND AND RELATED WORK

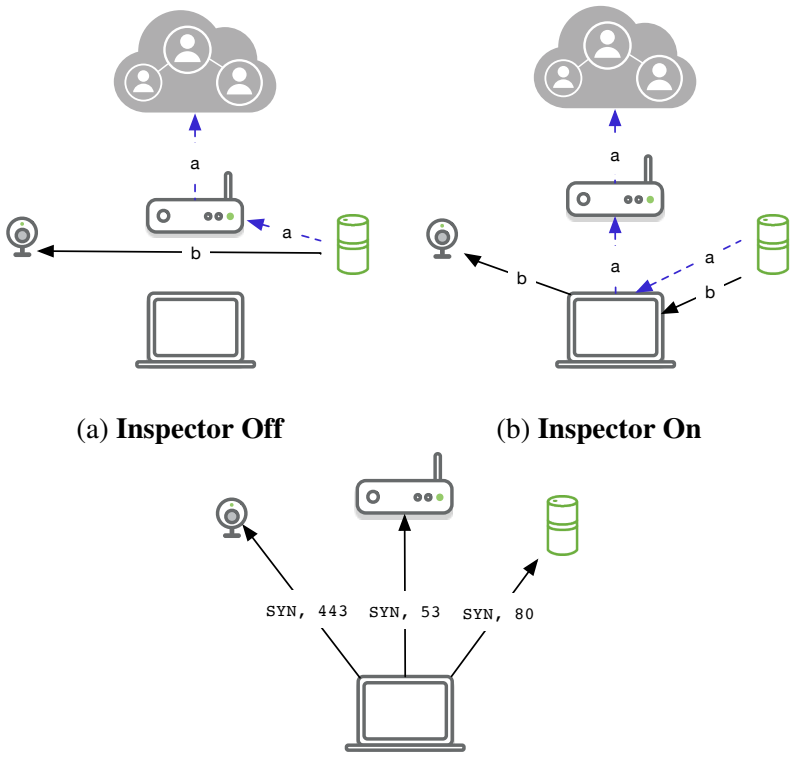
Our work compares two varying measurement techniques for understanding real-world IoT deployment. In this section, we describe prior work in the space and the techniques used.

5.2.1 Passive Measurement

Early Internet research leveraged passive measurements from telecommunication networks to measure traffic patterns and Internet delay times [158, 159]. Passive measurement has also been used to measure important network or security incidents in enterprise networks [160, 161]. For example, Zeek [161] and other similar intrusion detection systems use passive monitoring of network traffic to alert on security incidents and intrusions in real time. Studies have also leveraged passively collected darknet data to study security incidents like inferring DoS attacks at scale [12, 102] and tracking Internet wide scanning events [105, 162]. Most recently, DeKoven et al. leveraged passive data at on university residential network to investigate if following best practices adequately protects users from security threats [13].

5.2.2 Home IoT Measurements

Most similar to our work are three studies. The first, from Kumar et al. leveraged a corporate dataset of 83 M devices from around the world and characterized their device properties as well as their security profiles [148]. This data was primarily collected through active probing of devices on



(a) **Inspector Off** (b) **Inspector On**

(c) **Inspector SYN Scanning**

Figure 5.1: **IoT Inspector Workflow**—We extended IoT Inspector to capture not only external traffic (flow `a`, from an IoT device to a cloud service), but also to record internal traffic (flow `b`, device to device traffic). We also extended IoT inspector to perform an active, internal probe of all the devices connected to the internal network (Figure 5.1c).

the network. In contrast, work from Mazhar et al. instrumented the gateway software in 200 homes to collect passive data and investigate similar device properties [152]. Our vantage point enables us to compare and contrast these results directly within homes, and also outline the limitations of each approach in home IoT measurement. Finally, Huang et al. leveraged

5.3 METHODOLOGY

Our study primarily leverages Princeton IoT Inspector [15] to collect IoT traffic and device capabilities. Through IoT inspector, we are able to capture both active and passive measurements of IoT devices within the *same* home. IoT Inspector is an open-source tool that enables users to better understand their home IoT network traffic. Previously, IoT Inspector has been used to measure privacy tracking via smart TVs [5] and improve privacy through traffic shaping [163]. IoT Inspector works by serving as a voluntary man-in-the-middle between devices in the home network and the

gateway, thus logging any external communications that home IoT devices may make over the course of running the tool.

In order to leverage IoT Inspector to compare differing measurement techniques in homes, we augmented the tool to capture other measurement data. We first extended the tool's capability to not only collect external communications, but to also log all device-to-device traffic on the internal network. To do this, we had IoT Inspector serve as a man-in-the-middle between *all* devices, not just the device and the gateway, and logged metadata about each flow, such as the transport layer protocol, the ports communicated on, the number of bytes transferred, and which devices communicated with which other devices. We also labeled each device with its device type and device vendor by querying Fingerbank [164], an API that collects crowdsourced labeled fingerprints of devices and provides details based on a TCP handshake and DHCP lease data.

We also extended IoT inspector to collect active services offered by each device on the home network through a `SYN` scan. We could not scan every device on every port over TCP, as this may unintentionally cripple low powered IoT devices. We thus determined which ports to scan by leveraging those used by ZMap [8], and Avast WiFi Inspector, but augmented this list with ports that are typically used by IoT devices. To determine these, we leveraged an IoT dataset collected by Alrawi et al. which contains PCAPs and active scans of an IoT testbed containing 45 devices from 2018 [137]. In total, we extended IoT inspector to scan 329 ports on the internal network. A full diagram of our modified version of IoT Inspector is shown in Figure 5.1.

We deployed this version of IoT Inspector and collected data for a 3 week period, from May 8th to May 31st, 2020. In sum, our IoT Inspector dataset contains internal and external scans from 275 homes and covers 3,308 devices. We plan to release the dataset used in this chapter at publication time, and rolling IoT Inspector data is available for researchers.

5.3.1 Ethical Considerations

We levied a number of ethical considerations when conducting our work. Our usage of IoT Inspector is approved by our institution's IRB, and we follow industry-standard security and privacy practices. An important consideration is whether deploying our modified version of IoT inspector inside of a home may unintentionally break otherwise working IoT devices. Outside of thorough testing within local testbeds in our lab, we also clearly outline the risks of participating in this research whenever a new user installs IoT Inspector. Furthermore, per IoT Inspector's original design [15], we give users full control over the devices they wish to inspect—specifically, when a user installs IoT Inspector, no devices are inspected by default, and users can opt-in to having their device inspected by the tool. IoT Inspector does not capture any local traffic from devices that are not marked to be inspected. IoT Inspector also uses a heuristic to identify non-IoT devices (e.g.,

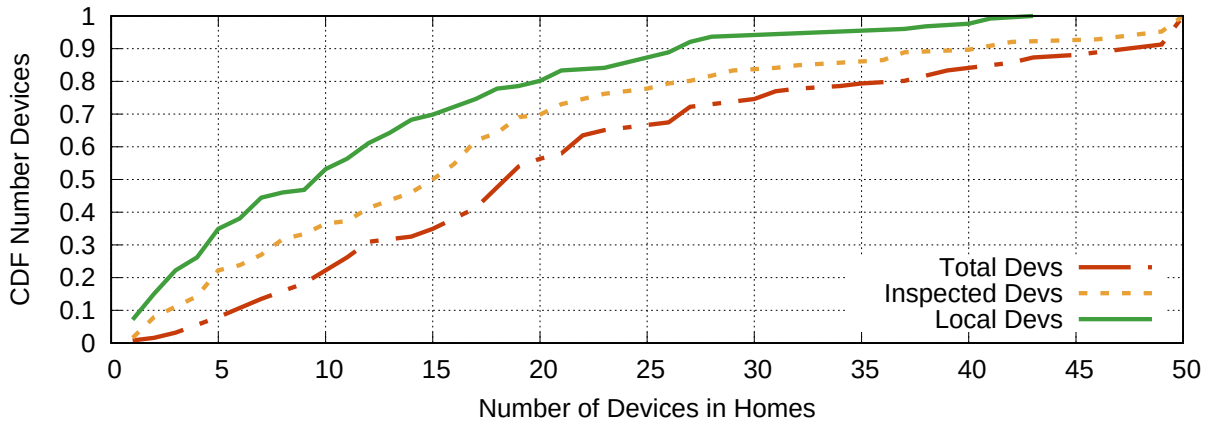


Figure 5.2: **Devices in Homes**—Homes in our dataset contain a median 19 devices, which is significantly higher than previous studies in smart homes. Of these, a median 15.5 (82%) are inspected by our tool, and of those, a median 10 (65%) generate local network traffic.

general purpose computers, cellphones) based on MAC addresses, and will not allow inspection of those devices without a manual override from the user.

5.4 PASSIVE OBSERVATION OF HOME IOT DEVICES

In this section, we detail our results collected through IoT Inspector in measuring 275 smart home networks containing 3,308 devices. These smart home networks are spread across 28 countries, including United States (46% of the networks), Canada (19%), and Sweden (15%). Using this dataset, we first present results in characterizing device behavior, quantifying the volume and types of traffic that are generated by home IoT devices. We then discuss the distinction between server capabilities (i.e., through active probing) and client behavior (i.e., through passive measurements.)

5.4.1 Homes and Devices in Aggregate

Homes in our dataset contain several devices connected to the local network. Figure 5.2 shows the absolute numbers of devices in each home, split by the total number of devices, the devices that were specifically inspected by the user, and the devices that ultimately generate local traffic. We note that users have the ability to inspect specific devices on the network, so our perspective contains only a fraction of the local network traffic occurring in each home. Homes contain a median 19 devices, with some homes containing up to 50 devices. We note that this is higher than homes measured in other contexts [148, 152], likely because IoT Inspector attracts power users of IoT devices [15]. As such, our results cannot be seen as a snapshot of an "average home", but rather

Device Type	Devices	Top OUI
Home IoT	768 (19.1%)	Espressif (30.9%)
Media	459 (16.8%)	Sonos (25.7%)
Voice Assistant	487 (12.1%)	Google (83%)
Camera	116 (2.8%)	Wyze (57%)
Television	102 (2.5%)	Roku (41%)
Work Appliance	68 (1.7%)	Xerox (50%)
Game Console	33 (0.8%)	Nintendo (64%)
Storage	28 (0.7%)	Synology (82%)
Unknown	1437 (35.7%)	—
Total	4030	—

Table 5.1: **IoT Device Type Distributions**—Homes in our dataset sport a wide range of devices and contain a median 19 devices, of which we could label 71% as an IoT device. The most common type of device are smart home IoT devices (19.1%), such as thermostats, smart plugs, and smart lights.

as an analysis of homes with heavier network deployments. Users inspected a median 15.5 (82%) devices in their homes, of which a median 10 (65%) generated local traffic during inspection. Users inspected devices in their homes for a median 307 minutes (5.1 hours).

Devices in our dataset span various device types and manufacturers. Table 5.1 shows a distribution of the types of devices in our dataset as well as the top vendor per category. We identify vendor by the Organizationally Unique Identifier (OUI) field of the device MAC address [101]. Although homes also contain other device types not listed here, such as mobile phones and computers, we specifically exclude general purpose computing devices as our focus is on home IoT traffic. The largest fraction of devices we could label were general smart home IoT devices (19.1%), such as thermostats (4%), lightbulbs (3.5%), and smart plugs (1.4%). This is followed by media devices (16.8%), like the Google Chromecast and Amazon Fire Stick, and home assistants (12.1%), such as the Google Home and Amazon Echo devices. In sum, our dataset consists of 2365 IoT devices produced by 190 manufacturers.

5.4.2 Local Device Communication

Devices frequently communicate with other devices on the local network. To quantify this, we track *flows*, which in this context are the 5-tuple combination of {`device_ip, device_port, remote_ip, remote_port, transport_protocol`}. Devices communicate with a median 15.8% other devices on the network, however, a long tail of devices are much chattier—10% of devices

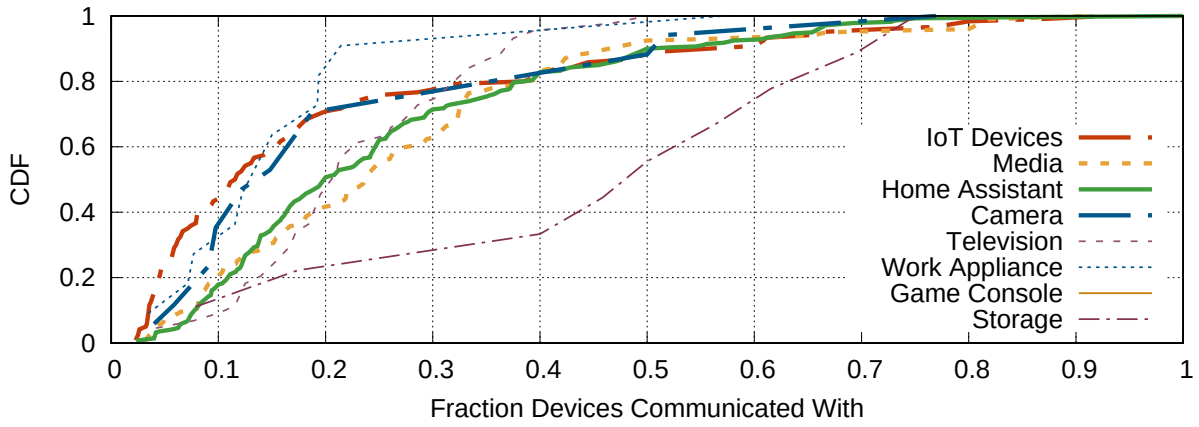


Figure 5.3: **Fraction Local Devices**—Devices in aggregate communicate with a median 15% of devices on the network, with variance aligned with their function. Storage devices, for example, communicate with a median 52% of devices, compared to smart home IoT devices which only communicate with a median 14% of devices.

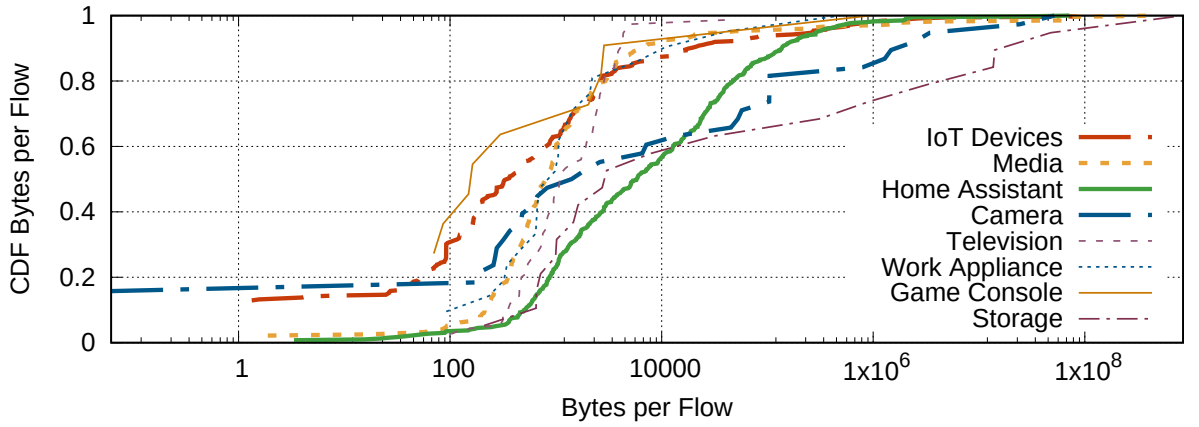


Figure 5.4: **Bytes per Flow**—Devices generate a median 1317 bytes per flow, again with variance across devices type. Voice assistants send the largest number of bytes per flow (5823 bytes/flow), while smart home IoT devices send the fewest (351 bytes/flow).

communicate with at least 50% of the devices on the network (Figure 5.4, Figure 5.3). Device flows are typically small in aggregate, with devices sending a median of 1317 bytes per flow.

Home IoT			Media			Voice Assistant			Camera			TV			Work			Gaming			Storage		
T.	**Fl.**	**D.**	**T.**	**Fl.**	**D.**	**T.**	**Fl.**	**D.**	**T.**	**Fl.**	**D.**	**T.**	**Fl.**	**D.**	**T.**	**Fl.**	**D.**	**T.**	**Fl.**	**D.**	**T.**	**Fl.**	**D**
Voice	48%	14%	Media	51%	47%	IoT	28%	5%	Comp.	31%	10%	TV	26%	23%	Media	83%	14%	Work	69%	10%	Voice	15%	11%
Comp.	11%	11%	Work	25%	3%	Media	17%	6%	Stor.	9%	4%	Voice	14%	11%	IoT	4%	4%	IoT	6%	16%	TV	9%	6%
Media	9%	9%	Comp.	13%	5%	Comp.	10%	10%	IoT	4%	11%	IoT	11%	7%	Work	0.6%	7%	Comp.	5%	10%	IoT	7%	10%
TV	6%	3%	Voice	5%	9%	Voice	9%	41%	Media	2%	7%	Comp.	11%	16%	Comp.	0.4%	18%	–	–	–	–		
IoT	4%	14%	IoT	1%	4%	TV	4%	2%	Work	0.2%	3%	Phone	6%	5%	–	–	–	–	–	–			

Table 5.2: **Device Type Communications**—IoT devices communicate with several other devices on the local network. We show the distribution of device communication per device type by type (T), sorted by fraction of flows (Fl.), and the fraction of total devices that particular type accounts for (D.)

Device behavior is distinct depending on device type. We observe differences in the volume of communication, the types of devices communicated with, and the ports used in communication across types. Table 5.2 shows the types of devices communicated with by fraction of flows and fraction of devices per device type. We discuss each device type in detail below:

Home IoT. Smart home IoT devices, such as lightbulbs and smart plugs, communicate with the fewest other devices on the local network, account for the plurality of devices in our dataset (19.1%). They communicate with a median 12% of devices on the network . They primarily communicate with other home IoT devices (14% of devices) and voice assistants (14%). Communication with voice assistants accounts for up the largest fraction of flows (48% of flows). Home IoT devices have a small network fingerprint, only sending a median of 351 bytes per flow, and primarily sending traffic over UDP (56%). Most communication with home IoT devices occurs over `443/HTTPS` (26%) and `80/HTTP` (13%), which are related to remote device control.

Media. Media devices, like Google Chromecasts or Sonos speakers, account for 16.8% of devices in our dataset. They communicate with a median 23% of devices on the network, mainly with other media devices (47% of devices, 51% of flows), voice assistants (9% of devices) and computers (5% of devices). Media devices send a median 820 bytes per flow and communicate over a long tail of ports specific to the manufacturer of the device. For example, Sonos speakers have a port open at `1400/HTTP` for remote control—we observe that to be their primary mechanism of communication.

Voice Assistants. Voice assistants, like the Amazon Echo and Google Home, account for 12.1% of devices in our dataset. Voice assistants communicate with a median 20% of devices on the network, and most prevalently communicate with other voice assistants (41% of devices) and computers (10%). For example, we observe Google Home devices frequently communicating over UDP for device discovery with other Google Home devices. Voice assistants send the most flows to IoT devices (28%), and communicate primarily over `443/HTTPS` (15%), `60000` (5.2%) and `8008/HTTP` (5%). Port `60000` is primarily used by Amazon devices, potentially for remote control[1].

Camera. Cameras communicate with a small fraction of devices on the network (median 15%) and primarily speak to IoT devices (11%) or computers (10%). In terms of number of flows, camera devices largely speak to computers (31% flows) or storage devices (9% of flows), likely to live stream or upload files taken from the device. Aligned with this, camera devices have a relatively

[1] https://forums.developer.amazon.com/questions/183535/network-congestion-from-192168491.html

high number of bytes per flow—1864 bytes per flow—compared to other specialized devices like IoT devices (351 bytes/flow) and gaming consoles (162 bytes/flow).

Television. Smart TVs communicate with a higher fraction of the network than devices in aggregate (21% of devices) and send a relatively volume of traffic per flow (1088 bytes/flow). Television devices communicate mainly with other television devices (23% devices, 26% of flows), computers (16% of devices, 11% of flows) and voice assistants (11% of devices, 14% of flow). Most television traffic in our dataset is generated by Roku TV, which enables a remote control port on `8060/HTTP`, and accounts for 65.6% of flows. However, TVs communicate over a long tail of services, with some TV flows occurring over `139/SMB` and `111/rpcbind`, both of which seem unlikely to be related to TV core function. Unfortunately, we do not collect application layer data through IoT Inspector and cannot attribute this traffic to malicious behavior, previous research has observed tracking behavior built into smart TVs [5]; understanding this ecosystem may be an area for future work.

Work Appliance. Work appliances, such as printers and scanners, account for only a small fraction of our dataset (1.7%), and communicate with a lower fraction of devices than devices in aggregate (12.2%). They primarily receive communication from computers (18% of devices) and media devices (14%), however, this is due to media devices regularly communicating with many devices on the network. Computers are typically used to send remote print jobs, either using a standard printing protocol (e.g., `631/IPP`) or through an open remote interface (`80/HTTP`).

Gaming. Gaming consoles account for a small fraction of our dataset (0.8%) and communicate infrequently with other devices (they account for 0.01% of measured flows in aggregate). From a network vantage point they appear to be largely self contained.

Storage. Storage devices appear the least frequently in our dataset (0.7% devices), however, they generate the second largest volume of traffic per flow (3022 bytes/flow) and communicate with the largest fraction of devices (52%) on the network. They primarily communicate over `554/RTSP`, with many storage devices likely serving as streaming box or for use in file transfer.

5.4.3 Characterizing Communication.

We expected most initiated traffic to be skewed towards simple discovery protocols (e.g., SSDP, UPnP) or UDP-based heartbeat protocols. Instead, we find that most initiated local flows were `TCP`-based (55%) on a variety of non-standard ports. Table 5.3 shows the most commonly used

Port	Service	Protocol	Frac. Devices
8009	TCP	HTTP	12.5%
80	TCP	HTTP	9.9%
1400	TCP	HTTP	4.8%
10001	UDP	–	4.6%
8008	TCP	HTTP	4.4%
9000	TCP	–	2.9%
8060	TCP	HTTP	2.2%
10001	TCP	–	2.0%
55443	TCP	–	2.0%
161	UDP	SNMP	1.9%

Table 5.3: **Port Distribution by Usage**—IoT devices communicate over a variety of standard and nonstandard ports, often to speak standardized protocols (e.g., HTTP). Of the top 10 services logged, 80% occurred over TCP.

ports from our flow collection, meaning these ports were the most actively used in communication on the local network. We observe 63% of local protocols leveraged TCP and 37% leveraged UDP. We manually investigated these flows, and identified many distinct, nonstandard ports used to speak popular protocols. For example, devices regularly use ports 80, 8008, 8009, and 8060 to host web servers. While it is not a new result that IoT devices leverage nonstandard ports for common protocols [15], it highlights an associated measurement challenge with IoT networks as vendors increasingly leverage bespoke ports.

In many cases, the usage of bespoke ports can be traced to a combination of manufacturers and device types. To illustrate this, we compute both the manufacturer distribution and device type distribution for each port used for communication in our dataset. We then take manufacturer and the device type that account for the *highest fraction of devices* for each port and plot them against each other in Figure 5.5. We observe four broad categories of ports, which we analyze per quadrant—quadrant I (upper-right), II (upper-left), III (bottom-left), and IV (bottom-right).

The largest fraction of ports (44%) fell into quadrant I, meaning they were largely used by a single device type and a single manufacturer. An example of this is port `1400/TCP`, which the company Sonos uses to host a web server for remote control. The second largest fraction was quadrant IV (28%), which are ports shared by a small number of device types but many manufacturers. One such example is `53/UDP`, which are typically offered by routers made by several vendors to service DNS queries. The third largest fraction were ports in quadrant III (24%), which are ports that are used by many device types and manufacturers. These are "standard" ports and protocols, for example, `80/TCP` and `443/TCP` are found in this quadrant. Finally, the fewest ports appear in quadrant II (4%), which are used by a singular manufacturer but spanning multiple device types.

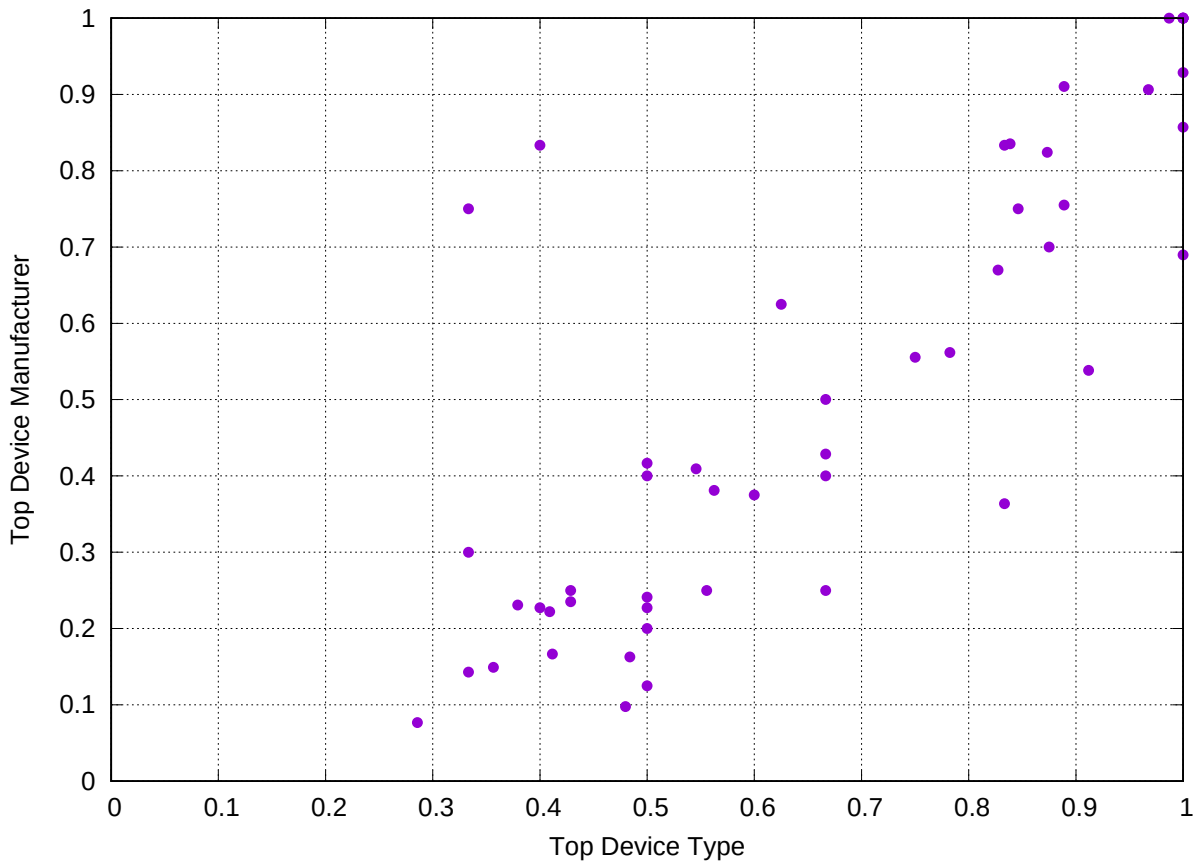


Figure 5.5: **Ports by Manufacturer and Device Type**—IoT devices communicate over a host of nonstandard ports and protocols, often dictated by their device type and manufacturer. Points in the upper right quadrant represent ports that are largely offered by a single manufacturer and for a specific device type. In our dataset, 44% of ports used in communication fell into this category, pointing to a fractured networking ecosystem with little standardization.

These results serve two functions. For one, the existence of bespoke ports can make device identification on home networks for some devices easier—a device running a webserver at port 1400 is likely a Sonos speaker, for instance. However, these results also highlight that without a priori knowledge of the devices in the network and their corresponding services, application layer insight remains an outstanding measurement challenge. Without complete knowledge of what devices use what ports and what ports serve what purposes, it could be difficult for network administrators—such as ISPs, home users, and vendors of home gateways—to develop comprehensive firewall rules that would block potentially malicious or sensitive traffic.

Device Type	Top External	Top Internal	Fraction Observed
Smart Home IoT	53/UDP (47.5%)	443/TCP (26%)	35%
Media	80/TCP (40%)	80/TCP (12.7%)	6%
Voice Assistant	53/UDP (56%)	443/TCP (15%)	27%
Camera	53/UDP (38%)	53/UDP (29%)	75%
Television	53/UDP (44%)	8060/TCP (66%)	22%
Work Appliance	53/UDP (82%)	80/TCP (48%)	12.5%
Game Console	3074/UDP (30.4%)	55280/TCP (16%)	–
Storage	53/UDP (19.3%)	554/UDP (17%)	42%

Table 5.4: **External vs. Internal Communication**—We show the distinctions between an external passive perspective and an internal passive perspective, as well as the fraction of used ports we observe from an external perspective alone. Devices have different external and internal services that they most commonly use, which varies by device type. A passive, external perspective alone would miss the rich internal communication that happens between devices—in the largest case, 94% of ports used by media devices would be missed by relying on a passive external perspective.

5.5 COMPARING INTERNAL AND EXTERNAL BEHAVIORS

Prior research has found that measuring the external footprint of a device through passive observation can product security and privacy insights. For example, using this view, Huang et al. were able to identify the IoT devices that communicate via TLS or use outdated version of TLS and highlight the tracking and advertising ecosystem that underlies home IoT devices [15]. Although these case studies are important in themselves, we note that the type of traffic generated by IoT devices inside home networks are significantly different than those generated to the outside world. This is not inherently surprising, but highlights that relying solely on passive observation from an external vantage point cannot adequately capture the full behavior of IoT devices.

The protocols that devices typically communicate with externally are 53/UDP (69%), 443/TCP (68%), and 80/TCP (43%), with a long tail of services that account for fewer than 20% of devices in our dataset. These services are used for for DNS, HTTPS, and HTTP-based communication to the outside world, which aligns with previous research into the top protocols used by IoT devices [137]. However, there is a significant difference between the protocols used to communicate externally and those used to communicate internally, varied again by device type. (Table 5.4). For example, although smart home IoT devices primarily communicate to the external world via DNS (53/UDP), they most often communicate inside home networks over HTTPS (443/TCP), which accounts for 26% of local flows with these devices. In other cases, we observe that external and internal behaviors align more closely. Camera devices in particular exhibited similar behavior inside home networks compared to their external fingerprint.

Beyond the top protocols, however, we find that some protocols would *never* be observed from

Port	Protocol	% Devices	Port	Protocol	% Devices
8008	HTTP	36%	445	SMB	6%
8443	MQTT	36%	7000	RTSP	5%
80	HTTP	31%	8888	HTTP	4%
443	HTTPS	17%	515	LPD	4%
8080	HTTP	12%	631	IPP	4%
1843	–	11%	554	RTSP	4%
1443	–	11%	9100	CUPS	3%
22	SSH	8%	8081	HTTP	3%
8060	–	6%	1080	SOCKS	3%
139	SMB	6%	6467	–	3%

Table 5.5: **Active Services on IoT Devices**—63% of IoT devices offer a TCP-based service on at least one of the ports we scanned for. Devices primarily run an HTTP or HTTPS based webserver for remote control, however, several services are offered.

a passive external perspective. Table 5.4 shows the fraction of services per device type that were observed using an external perspective alone. In some cases, relying only on an external perspective would miss almost all the behaviors of the device—for example, 94% of ports used by media devices were not captured through observing external communication. However, some device types are almost entirely encompassed by external observation, for example, 75% of services used by cameras are captured via external perspectives. We note camera devices also rarely communicate with other devices on the local network (15%), so the majority of their traffic is external facing.

These results highlight that IoT device behaviors cannot be captured by their external footprint alone, and are typically more complex than what is measured from that perspective alone.

5.6 COMPARING DEVICE BEHAVIOR AND CAPABILITIES

Previous studies of IoT deployments and measurements have primarily leveraged active scans [137, 148, 149], which typically involve a SYN scan and some application layer probing for labeling devices and enumerating device capabilities. We next answer a related question: what are the differences between offered services (i.e., server capabilities) compared to how the device is used in practice?

To begin, we conduct a SYN scan of IoT devices connected to the local network. We scan for a select number of ports drawn from previous studies of home IoT devices. Of the 2365 IoT devices inspected in our scans, we find 1756 (73%) respond to our SYN on at least one port. We note that because our scanning list is fixed a priori, we are likely under reporting the fraction of devices that have a TCP-based service available. IoT devices in our dataset run a median 2 services, however,

Device Type	Fraction Ports Used
Television	61%
Camera	41.3%
Router	37.5%
Gaming Console	25%
Smart Home IoT	17.1%
Media	16.8%
Work Appliance	13.3%
Voice Assistant	12.2%
Storage	12%

Table 5.6: **Fraction Ports Used by Device Type**—The fraction of used ports varies by device type. Television, cameras, and routers are more likely to support ports that were used in our measurements, compared to storage devices and voice assistants.

3% of devices in our dataset run more than 10 services, indicating a wide range of offered protocols. Table 5.5 shows the distribution of ports, protocols, and the fraction of devices that support each protocol. The most prevalent ports were `8008` (36%), `8443` (36%), `80` (31%) and `443` (17%), all of which are used to host web servers for remote control of the device.

We next compare these active services to the services that are used in-context, and observe that only an average of 19% of services offered per device are actively used. However, this may simply be a result of our measurement apparatus—for example, the port may only be used for very specific operations that occur infrequently, and may not have occurred in the time we captured traffic from the device. To control for this, we sorted each active port by the *fraction* of devices that did not leverage the port during communication, and only examined ports that were used by at least 10 devices in our dataset.

Similar to our previous results, the fraction of unused ports differs by device type (Table 5.6). Specialized devices, like storage and voice assistants, offered the least fraction of used ports, with 12% and 12.2% used respectively. Conversely, television and camera devices typically offered ports that were used in practice: 61% of ports offered by TVs were used in our measurements, and 41.3% ports offered by cameras were used in our measurements.

The vendors that built devices with the lowest fraction of used ports primarily build media devices and cameras (Table 5.7). In the largest case, Amcrest devices did used an average of (92%) of offered ports, and Roku devices used an average of 88% of ports offered by the devices. Conversely, several higher profile IoT vendors, such as Amazon, offer ports that do not serve device function. For example, Amazon voice assistants regularly come with `1080/TCP` and `8888/TCP` open, despite never using them in our measurements.

We next investigate what specific services are offered but least frequently used by devices in our

Vendor	Most Popular Type	Average Ports Used
Amcrest	Media	92%
Roku	Media	88%
Dahua	Camera	50%
Lite-On	Media	40%
Philips	Smart Home IoT	37%
Canon	Work Appliance	28%
SONY	Media	27%
Dish	Media	25%
Samsung	Media	23%
Vizio	Media	23%

Table 5.7: **Vendors with Highest Used Port Fraction**—The top vendors that serve used ports in our measurements primarily manufacturer media devices (70%). In the largest case, The top vendor, Amcrest, on average only did not use 8% of ports during our measurements.

Port	Unused	Expl?	Port	Unused	Expl?
22/SSH	100%	✗	7000	97%	✗
9100/CUPS	100%	✓	8443	97%	✓
8081/HTTP	100%	✗	1843	97%	✗
111/rpcbind	100%	✗	23/Telnet	96%	✗
6466	100%	✗	8888/HTTP	95%	✗
6467	100%	✗	515/LPD	94%	✓
8222	100%	✗	443/HTTPS	93%	✗
1080/SOCKS	100%	✗	139/SMB	92%	✓
8889	100%	✗	631/IPP	91%	✓

Table 5.8: **Unused Services on IoT Devices**—Many popular services offered by IoT devices remain unused during our measurement period. Although some services can be explained due to protocols that are used intermittently (e.g., a printing protocol), we observe many IoT devices offer services that are not used in normal function and place these devices at an elevated security risk. For example, 2.3% of devices support 23/Telnet, despite 96% of these devices never using the protocol once.

dataset. Table 5.8 shows the top 18 unused ports as well as the fraction of devices that did not use the port during our measurement period. We then manually investigated each port to identify if the lack of usage could be easily explained by our measurement methodology. We find that IoT devices offer many services that are rarely if ever used during our measurement period. For example, 12.5% of IoT devices offer an SSH server on port 22; however, we find this port was never used in our on any device in our measurements.

Some unused protocols could be explained due to bias from our measurement vantage point. For example, printing protocols, such as 9100/CUPS, 515/LPD, and 631/IPP may only be used when a

print job arrives, which may not have occurred during our measurements. However, some unused ports are more difficult to justify, especially when many of these ports are associated with security vulnerabilities. For instance, 2.6% of devices support `111/rpcbind`, which has a long chain of exploits and CVEs and is never used in our local flow measurements[2]. Similarly, 2.3% of devices support `23/Telnet`, which has been a consistent threat for IoT devices due common, weak default credentials [102]. These ports increase the attack surface of devices, and simply the existence of unused ports can lead to a host of security and privacy challenges [165].

These security and privacy risks are compounded by the challenges in identifying unused ports. First, researchers need to conduct both active scans (i.e., to identify open ports) and passive traffic analysis (i.e., to identify used ports). Moreover, many of the unused protocols are on bespoke ports, such as 10001 and 55443 (Section 5.4.3), further complicating the effort to develop firewall rules (e.g., by ISPs, home users, or automated systems on gateways) that can block potentially malicious traffic.

5.7 LIMITATIONS AND FUTURE WORK

In this section, we discuss the limitations of our work and outline areas of future work drawn from our results.

Exercising Full Device Behavior. Although IoT Inspector presents a useful snapshot of local network flows, it is only able to collect measurements while running, and users typically close the application after 307 minutes. In the context of this work, this points to a key limitations regarding our device behavior analysis: there is always a possibility that the full networking capabilities IoT devices are not exercised during our measurement period. Thus our results about devices not using every port for proper device function are limited to only our measurement collection period—it is possible that devices may have eventually used those services with longer measurement periods. In spite of that, we find it questionable that many IoT devices require more than a handful of network services to properly function, and argue that offering more services than required poses a security risk to those devices.

As such, an area of future work is in de-bloating the networking behavior of IoT devices by removing any networking function that are not core to the device behavior. Prior work has leveraged tools like symbolic execution and software analysis techniques to find bugs and security flaws in firmware and control applications [166, 167], however, little work exists in understanding the full networking capabilities of these devices in every action, either through network based fuzzing or

[2]https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-1816, https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-1349, https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-8779

protocol enumeration [168]. Recent work has investigated automatically generating firewall rules from IoT traffic, but much work needs to be done to understand if this can be done at scale [169].

Security vs. Usability The prevalence of device-to-device communications on the local network presents a unique challenge that highlights the tension between security and usability. On one hand, local communication appears to facilitate cooperation among devices. For example, 14% of Home IoT devices communicated with voice assistants presumably to allow for voice control of such devices. Many of these communications do not require prior authentication. Previous work has shown that any device or applications could communicate with, for instance, Google voice assistants on `8008/HTTP` [165] or Roku TVs on `8060/HTTP` [5].

Although this openness in communication may promote the usability of multiple IoT devices in tandem, it introduces potential security and privacy risks. As Acar et al. observed in a lab environment, a malicious IoT device or a malicious app on a computer may control (e.g., shutting down) or access sensitive information (e.g., precise location of users) from another device [165]. While our analysis does not have sufficient evidence for such malicious activities in the wild, the open and prevalent nature of device-to-device communication does raise an alarm and call for future research in this area.

Standardizing Networking Behavior. In our results, we observed IoT devices frequently communicating on a myriad of nonstandard ports for common protocols, such as HTTP. From a measurement perspective, this adds an additional challenge in measuring home IoT networks—it requires a priori knowledge of the services running on each port to measure application layer properties of the device. An area of future research is to properly enumerate these services (in-context) and build a growing set of port-to-protocol mapping for finer grained measurements with these devices.

Inferring Network Roles by Behavior. Many recent studies have focused on building better local network policies as home networks continue to grow in size [132, 155]. However, each study relies on knowing what devices are on the network a priori, and for the user to define "correct" and "incorrect" behavior on the network through specially crafted policies and rules. Our results suggest that IoT devices exhibit different behavior based on their device type. For example, smart home devices sent 94% fewer bytes per flow than voice assistants. Many of these behavioral properties of devices—the fraction of devices they communicate with, their outbound flow rates, the fraction of packets sent via TCP vs. UDP, their port distribution—may be useful in automatically inferring their device type as well as their default "normal" network behavior. Using these observations, an area of future work may be to leverage local network behavior of devices to automatically infer

and implement network policies that protect devices on the network, thereby limiting the potential damage of a compromised device.

5.8 CONCLUSION

In this chapter, we collected both passive and active measurements of 3,308 devices collected from 275 smart homes over a three-week period. We found that IoT device behavior varies by device type, and furthermore, manufacturers often leverage bespoke ports for common protocols, which complicates active measurements of the home IoT devices. Finally, we compared passive observation with active scans of local IoT networks, and identified several services that are offered by a wide spread of devices but rarely used, pointing to evidence of bloated networking capabilities on IoT devices. We conclude with a discussion of limitations and future work. We hope our results will be useful for the measurement and IoT communities as the ecosystem continues to develop.

CHAPTER 6: CONCLUSION AND FUTURE DIRECTIONS

In this thesis, we demonstrated how drawing network measurements from a single vantage point or technique leads to a biased view of the network services present in the IoT ecosystem. We performed measurements from several vantage points and techniques, and showed how the capabilities and behaviors of devices change based on the chosen measurement perspective.

In Chapter 2, we documented how active scanning could be used to identify devices that had been infected with Mirai malware, and showed that many of world's top commercial electronics vendors did not follow best practices and lacked adequate protection from the attack. We also showed how passive, external observation through a DNS tap aided in understanding the command-and-control infrastructure behind the botnet, lending evidence to competing botmasters and enabling us to track the impact of the vulnerability over time.

In Chapter 3, we measured leveraged active scans of the inside of 16M home networks and showed the diversity of IoT devices and vendors in regions around the world. We found that security threats, such as weak, guessable passwords on open services, are widespread and vary heavily based on device type, region, and vendor. Furthermore, we leveraged a large, passive darknet to quantify IoT device compromise, finding that while the darknet was able to help identify potentially compromise home networks, we could not use it to perform any deeper device attribution. Our results point to a fractured IoT ecosystem, and furthermore, that the types of devices studied by the security community are rarely the ones that are at larger risk of network vulnerability.

In Chapter 4, we combined our datasets from Chapter 2 and Chapter 3 and demonstrated that relying on active, external scanning alone under-reports IoT device types and security critical protocols that are prevalent inside homes. However, we also observe that active external scanning through only HTTPS served as a reasonable proxy for IoT device type distributions inside home networks, highlighting an opportunity for future research to collect a large array of IoT devices.

Finally, in Chapter 5, we conducted passive measurements from inside 275 smart homes, and observed that IoT devices regularly communicate with other IoT devices on the local network. Furthermore, we found that device behavior depends heavily on both the type of device and the manufacturer of the device. In comparing external device behavior to internal device behavior, we show that that a passive, external perspective would miss a significant fraction of the protocols commonly used by devices—in the largest case, 94% of protocols communicated by media devices would not be observed through passive, external observation. In combining an active and passive approach, we find that devices regularly support server capabilities that are rarely used in normal communication, with a median 50% of services unused per device during our measurements. Worse, many unused services are security critical (e.g., `23/Telnet`, `111/rpcbind`), increasing the attack

surface of these devices while adding little to no additional function.

6.1 LESSONS LEARNED AND FUTURE WORK

To conclude, we discuss some high-level lessons learned over the course of this research and implications for the measurement and security communities.

6.1.1 Recommendations for Measurement

With each measurement, we observed a fractured, heterogenous ecosystem spanning architectures, devices, vendors, behaviors, and world regions. We offer some suggestions and future directions for how to conduct these measurements:

Combining Measurement Techniques is Necessary. Measurements drawn from a single vantage point or technique cannot capture the complexity of the IoT ecosystem. As shown in Chapter 5, IoT devices often communicate on protocols that are different than the ones they offer as servers, and this behavior varies depending on external communication versus internal communication. As such, any future research should draw on multiple techniques—both active scanning and passive measurement—wherever possible when measuring the IoT ecosystem.

Local Network Measurement is Key. IoT measurements should be drawn from where the IoT devices are: inside local networks. As we showed in Chapter 5, IoT devices on local networks are relatively chatty, and access to this data can help to solve device identification which informs IoT device distributions. This is not to say external measurements cannot be useful—as we showed in Chapter 2, external measurements were used for device identification and attribution for IP addresses we observed to be scanning. However, our identification results were limited, as we were only able to label 31.5% of devices across all services and protocols. Comparatively, we were able to label (using active scanning) 91% of devices scanned inside local networks. In addition, we showed in Chapter 4 how IoT device distributions connected externally did not typically reflect the device type distributions inside home networks. Tools like IoT Inspector (Chapter 5) can aid in collecting measurements to drive future research.

6.1.2 Directions for Systems and Security

Beyond our discussion about proper measurement of the IoT ecosystem, our measurements themselves offer future directions and recommendations for the security community.

Improving Local Network Protections. We were surprised to learn that devices already communicate frequently over the local network, and not simply to perform menial tasks like device discovery or service enumeration. Instead, local network communication is rich, with devices communicating over many TCP-based application layer services to send data and issue control commands to other devices. We posit this will only increase as we add more devices to our local networks. As such, it becomes imperative to challenge our current local network assumptions: that all devices that enter are trusted, and any device can or should freely be able to communicate with other devices. Some work in this space has already been published [16, 17, 169], however, it is not informed by real world measurements or real world traffic. Many of these systems also make inherent assumptions about device network behavior that are not necessarily true—for example, HanGuard [170] and Hestia [17] both assume network level interaction primarily occurs between a device to a controller (e.g., smart hub, mobile device). Unfortunately, this is not true in practice—devices frequently communicate with other IoT devices on local network, in some cases to receive control commands (e.g., voice assistant to a smart home IoT device).

An area of future work is to leverage our measurements of in-home networks to automatically infer network policies that improve security without compromising functionality. We observed that many devices, for example, offer protocols that are never used in regular communication. New, defensive standards for local networks can infer these services are unused, and automatically block any traffic flowing to each device on these ports. Furthermore, we showed that device behavior is distinct across device type and manufacturer, highlighting an opportunity for additional, automated network policy generation.

Informing Investigation with Real-World Data. Recent security research has focused on new home IoT devices, such as smart locks and home automation. Our results suggest that while these devices are growing in importance in western regions, they are far from the most common IoT devices around the world. Instead, home IoT is better characterized by smart TVs, printers, game consoles, and surveillance devices—devices that have been connected to our home networks for more than a decade. Furthermore, these are the kinds of devices that still support weak credentials for old protocols: work appliances are the device type with the highest fraction of weak FTP credentials; surveillance devices are the worst for telnet credentials. Improving the security posture of these devices remains just as important as ensuring that new technologies are secure—our home networks are only as secure as their weakest link.

APPENDIX A:

A.1 AVAST DATA SHARING POLICY

The first panel in Figure 3.1 presents users with a text blurb about WiFi Inspector's data sharing policy. For ease of reading, we have copied that text below here:

> Nearly every software product you use collects information about you. Search engines, games, everything. We do the same. This allows us to provide better products and services for you. But we promise to respect your privacy. We also promise that we will never publish or share any of your personal information outside Avast, nor allow anyone else to use it to contact you for marketing purposes without your consent.
>
> We do use the information that we collect to help us understand new and interesting trends. We may share this information with third parties outside Avast. However, before we do that, we will remove anything that identifies you personally. For more information, read our Privacy Policy.
>
> If after installing this product, you'd prefer not to participate in data sharing with Avast and third parties, you can opt-out at any time by unchecking the "participate in data-sharing" box in the settings.

A.2 DEVICE LANDSCAPE

	Routers		Gaming		Automation		Storage		Surveillance		Work		Assistant		Media	
N. America	16.4	Arris	39.2	Microsoft	44.2	Nest	24.9	W Digital	12.1	Hikvision	38.8	HP	63.2	Amazon	17.4	Roku
	8.1	Cisco	19.7	Nintendo	15.1	Belkin	14.1	Synology	7.3	Dahua	10.3	FoxConn	32.0	Google	10.2	Amazon
	5.2	Sagemcom	11.6	Azurewave	14.4	Phillips	5.9	Seagate	6.3	D-Link	8.4	Amazon	1.7	Unknown	9.9	Samsung
	4.6	Actiontec	9.4	Sony	9.8	ecobee	3.9	ICP	5.8	Suga	8.0	Epson	0.8	StreamU	5.9	Apple
	4.3	TP-Link	9.0	FoxConn	2.7	Enphase	3.0	WD	5.3	Flir	7.5	Canon	0.4	Apple	5.8	Google
S. America	22.2	TP-Link	43.7	Microsoft	33.5	Philips	25.0	W Digital	20.8	Hikvision	29.2	HP	39.1	Google	26.0	Samsung
	7.7	Arris	13.6	Sony	13.0	Belkin	14.7	Sagemcom	16.3	Dahua	18.0	Epson	27.5	Amazon	13.6	Arcadyan
	7.0	Technicolor	10.7	Azurewave	12.1	–	13.1	Synology	8.4	–	9.0	FoxConn	6.2	–	7.5	Google
	6.5	Huawei	9.6	FoxConn	5.9	SMA	9.7	D-Link	8.2	Intelbras	7.1	Brother	3.7	TI	6.3	LG
	4.6	Mitrastar	6.6	Nintendo	4.7	Enphase	8.5	Seagate	4.0	Cisco	5.7	Samsung	3.2	Dell	5.0	Intelbras
East Asia	12.9	NEC	45.9	Nintendo	49.0	Philips	37.2	Synology	28.5	Hikvision	13.4	Canon	56.2	Google	8.6	Panasonic
	11.9	Buffalo	21.9	Sony	7.0	Belkin	13.4	Buffalo	10.5	Dahua	11.1	Epson	32.6	Amazon	7.5	Amazon
	8.4	TP-Link	8.9	FoxConn	4.8	Belkin	12.1	ICP	8.6	Dahua	10.6	Moimstone	2.1	Xiaomi	6.9	FoxConn
	5.5	EFM	8.0	Azurewave	4.2	Gongjin Elec	8.8	I-OData	5.0	Panasonic	9.3	FoxConn	0.7	TCL	6.3	Google
	4.4	Huawei	4.9	Microsoft	4.2	SMA	8.2	QNAP	2.4	Bilian	9.2	HP	0.7	Onkyo	5.9	Sony
Central Asia	49.5	TP-Link	22.8	Microsoft	11.1	Fn-Link	37.4	Synology	43.2	Hikvision	23.7	HP	21.3	Amazon	37.2	Samsung
	16.6	Huawei	20.9	FoxConn	11.1	Cambridge	14.0	D-Link	16.2	Dahua	10.0	Yealink	17.0	Amazon	28.6	LG
	6.4	Cambridge	17.7	Azurewave	11.1	TP-Link	13.5	W Digital	11.0	Cisco	9.4	Canon	6.4	D-Link	6.9	FoxConn
	5.3	D-Link	12.5	Sony	–	–	7.7	ICP	6.2	Cisco	7.5	Epson	4.3	M-Cube	–	–
	3.0	ZTE	10.0	Liteon	–	–	4.1	QNAP	3.2	ICP	6.9	XEROX	4.3	TI	–	–
East Europe	23.9	TP-Link	37.3	Microsoft	40.3	Philips	26.7	Synology	20.6	Hikvision	27.7	HP	44.9	Google	30.8	Samsung
	7.3	ZTE	14.7	Sony	25.1	Philips	15.9	W Digital	18.7	Dahua	10.8	FoxConn	23.7	Amazon	17.0	LG
	7.1	Huawei	13.2	FoxConn	5.4	SMA	14.0	Sagemcom	12.0	Cisco	7.1	Canon	7.6	Amazon	5.4	FoxConn
	6.6	D-Link	11.0	Azurewave	3.2	eQ-3	9.7	ICP	4.3	Cisco	5.6	Epson	2.4	TI	4.7	Google
	3.8	Asus	9.5	Nintendo	3.2	Murata	7.6	QNAP	3.4	ICP	4.9	Samsung	2.3	Telemedia	3.3	Neweb
West Europe	18.0	Sagemcom	30.6	Microsoft	33.1	Philips	38.7	Synology	37.1	Free	39.0	HP	48.6	Amazon	15.7	Sagemcom
	16.1	Free	22.5	Nintendo	17.7	Alertme.com	17.7	W Digital	8.0	Hikvision	11.6	Canon	37.2	Google	14.1	Samsung
	5.7	AVM	14.9	Sony	6.1	eQ-3	7.2	ICP	7.0	Hikvision	9.2	FoxConn	6.4	Apple	9.3	Free
	5.2	Huawei	11.5	FoxConn	5.7	Hager	5.7	Technicolor	6.3	Dahua	9.0	Epson	0.7	Apple	8.4	Google
	3.8	TP-Link	8.3	Azurewave	4.8	SMA	4.5	QNAP	5.1	D-Link	4.1	Brother	0.6	Telemedia	6.2	Google

Table A.1: **Most Popular Vendor per Region per Device Type, 1**—We show the five most popular vendors per device type across the eleven regions in our dataset. We excluded two device types, wearable and home appliances, as they were barely present in our dataset and splitting up their vendor distribution by region provided only a handful of devices in each region.

	Routers		Gaming		Automation		Storage		Surveillance		Work		Assistant		Media	
East Europe	23.9	TP-Link	37.3	Microsoft	40.3	Philips	26.7	Synology	20.6	Hikvision	27.7	HP	44.9	Google	30.8	Samsung
	7.3	ZTE	14.7	Sony	25.1	Philips	15.9	W Digital	18.7	Dahua	10.8	FoxConn	23.7	Amazon	17.0	LG
	7.1	Huawei	13.2	FoxConn	5.4	SMA	14.0	Sagemcom	12.0	Cisco	7.1	Canon	7.6	Amazon	5.4	FoxConn
	6.6	D-Link	11.0	Azurewave	3.2	eQ-3	9.7	ICP	4.3	Cisco	5.6	Epson	2.4	TI	4.7	Google
	3.8	Asus	9.5	Nintendo	3.2	Murata	7.6	QNAP	3.4	ICP	4.9	Samsung	2.3	Telemedia	3.3	Neweb
West Europe	18.0	Sagemcom	30.6	Microsoft	33.1	Philips	38.7	Synology	37.1	Free	39.0	HP	48.6	Amazon	15.7	Sagemcom
	16.1	Free	22.5	Nintendo	17.7	Alertme.com	17.7	W Digital	8.0	Hikvision	11.6	Canon	37.2	Google	14.1	Samsung
	5.7	AVM	14.9	Sony	6.1	eQ-3	7.2	ICP	7.0	Hikvision	9.2	FoxConn	6.4	Apple	9.3	Free
	5.2	Huawei	11.5	FoxConn	5.7	Hager	5.7	Technicolor	6.3	Dahua	9.0	Epson	0.7	Apple	8.4	Google
	3.8	TP-Link	8.3	Azurewave	4.8	SMA	4.5	QNAP	5.1	D-Link	4.1	Brother	0.6	Telemedia	6.2	Google
South Asia	24.2	TP-Link	64.9	Microsoft	26.3	Philips	20.1	W Digital	34.3	Hikvision	33.1	HP	44.8	Google	17.1	FoxConn
	7.4	Huawei	8.7	FoxConn	24.1	SMA	14.5	Synology	18.4	Dahua	16.6	Canon	33.8	Amazon	16.9	Samsung
	7.4	D-Link	5.7	Azurewave	14.0	Matrix	14.5	Synology	18.4	Dahua	8.1	FoxConn	2.7	HP	8.3	LG
	7.3	Tenda	3.6	Sony	1.3	Espressif	10.6	Seagate	3.0	Cisco	6.0	Epson	2.5	Dell	6.1	Google
	2.7	Haier	2.0	Nintendo	1.3	Xiaomi	10.3	WD	2.1	ICP	3.6	Ricoh	1.8	Intel	5.5	Neweb
S.E. Asia	18.9	TP-Link	44.6	Microsoft	34.7	Inspur	36.4	Synology	24.7	Hikvision	15.4	HP	49.1	Google	19.7	Samsung
	14.3	Huawei	11.6	Nintendo	18.9	Philips	19.4	W Digital	17.2	Dahua	13.9	FoxConn	21.7	Amazon	10.8	FoxConn
	12.0	ZTE	11.5	FoxConn	18.6	Rf-Link	8.6	ICP	4.8	Cisco	9.7	Epson	2.7	TI	10.6	ZTE
	5.3	Fiberhome	10.2	Azurewave	8.2	SMA	7.5	QNAP	4.0	ICP	9.5	Canon	2.6	HP	10.5	LG
	4.3	Mikrotic	6.5	Sony	2.0	Belkin	6.6	D-Link	3.8	PLUS	7.3	Ricoh	2.3	Dell	4.1	Neweb
Oceania	19.3	Technicolor	43.7	Microsoft	30.3	Philips	21.0	Synology	16.8	Hikvision	23.5	HP	85.3	Google	17.7	Google
	15.4	Huawei	15.0	Nintendo	20.3	Belkin	15.9	HyBroad	13.9	Dahua	19.3	FoxConn	8.0	Amazon	12.2	Roku
	12.1	Sagemcom	11.1	FoxConn	16.4	Lifi	13.1	W Digital	3.7	D-Link	14.1	Epson	1.3	Apple	10.0	Apple
	7.6	TP-Link	10.3	Azurewave	10.1	Enphase	9.5	ICP	3.4	Baichuan	10.2	Canon	1.3	Apple	8.6	Samsung
	4.7	Netcomm	9.3	Sony	6.2	SMA	6.5	Seagate	3.0	Yealink	6.5	Brother	0.6	Liteon	6.8	Sonos
N. Africa, ME	25.6	Huawei	26.0	Microsoft	27.3	Philips	29.1	Askey	19.5	Hikvision	29.4	HP	27.6	Google	20.9	Samsung
	23.2	TP-Link	18.7	FoxConn	10.6	SMA	19.2	W Digital	15.3	Dahua	9.7	FoxConn	21.3	Amazon	17.2	LG
	8.4	ZTE	16.6	Sony	8.1	Lifi	9.7	ICP	5.4	Cisco	7.2	Canon	1.9	Dell	5.4	Vestel
	6.1	D-Link	12.2	Azurewave	3.2	Sercomm	9.1	Synology	4.3	Topwell	4.3	Samsung	1.9	Apple	3.8	Sagemcom
	4.7	Zyxel	7.7	Liteon	2.7	ZTE	7.7	VTech	4.0	ICP	3.9	Konika	1.8	HP	2.7	Apple
S-S Africa	19.7	Huawei	40.7	Microsoft	21.1	SMA	24.7	Synology	39.0	Hikvision	33.6	HP	33.8	Google	24.1	Samsung
	12.0	TP-Link	14.5	FoxConn	17.6	TI	19.2	W Digital	16.3	Dahua	8.5	Canon	28.8	Amazon	7.4	LG
	8.1	Ubiquiti	13.9	Sony	10.8	Philips	10.1	ICP	2.8	Cisco	8.4	Yealink	7.3	HP	7.4	LG
	6.7	Mikrotic	9.7	Azurewave	3.9	HP	9.3	QNAP	2.2	ICP	6.3	FoxConn	2.9	Dell	5.8	Apple
	6.5	D-Link	8.4	Nintendo	2.9	Hager	7.8	Seagate	1.7	PLUS	5.3	Ricoh	2.2	Apple	5.2	Sagemcom

Table A.2: **Most Popular Vendor per Region per Device Type, 2**—We show the five most popular vendors per device type across the eleven regions in our dataset. We excluded two device types, wearable and home appliances, as they were barely present in our dataset and splitting up their vendor distribution by region provided only a handful of devices in each region.

Region	FTP								Telnet			
	Work Appliance		Storage		Surveillance		Home Router		Surveillance		Home Router	
N. America	35.3	HP	40.1	ICP	49.8	Axis	63.8	TP-Link	42.9	Dahua	45.2	TP-Link
	13.9	Ricoh	25.2	W Digital	13.4	Vivotek	9.6	ZTE	22.7	PLUS	40.5	Zyxel
	9.3	FoxConn	10.0	QNAP	7.9	Trendnet	8.0	Mikrotic	9.3	Metrohm	4.4	–
	8.4	Kyocera	6.7	TP-Link	4.6	D-link	4.3	–	4.8	–	4.1	Belkin
	7.9	Sharp	5.5	WD	2.9	Creston	2.0	T&W	3.7	Cisco	0.7	Intelbras
S. America	39.6	Ricoh	24.2	W Digital	43.3	Vivotek	40.3	Technicolor	51.4	PLUS	44.5	Intelbras
	25.3	HP	20.2	ICP	30.6	Axis	28.5	TP-Link	10.3	Cisco	21.6	Huawei
	22.8	Kyocera	12.9	QNAP	6.9	Level One	11.1	D-Link	9.8	Metrohm	12.0	BluCastle
	3.3	Sharp	8.1	Cisco	6.5	D-Link	7.4	Mikrotic	8.8	Dahua	5.9	TP-Link
	1.2	Xerox	7.3	La Cie	2.2	Trendnet	4.7	Cameo	3.3	Ralink	5.7	Loopcomm
East Asia	39.9	Ricoh	49.0	I-O	44.3	Vivotek	62.4	TP-Link	43.8	PLUS	45.8	NEC
	17.6	Sharp	25.4	ICP	27.8	Axis	14.1	I-O	11.9	Metrohm	18.6	Hitron
	8.6	HP	7.9	QNAP	8.7	Logitec	6.9	DrayTek	10.8	Dahua	15.6	Huawei
	7.4	Kyocera	7.7	EFM	4.3	Imi	2.9	corega	9.2	ICP	4.9	Buffalo
	5.6	Xerox	1.7	inXtron	3.5	Buffalo	1.8	Mikrotic	4.3	Cisco	4.7	TP-Link
Central Asia	66.4	HP	66.7	ICP	39.1	Axis	92.6	TP-Link	36.0	PLUS	52.3	D-Link
	9.3	FoxConn	33.3	QNAP	17.4	Zhongxi	1.9	Huawei	16.9	Dahua	35.2	Huawei
	11.5	Kyocera	–	–	13.0	Ezvis	1.5	ZTE	11.2	–	8.8	Cambridge
	3.5	Ricoh	–	–	13.0	Vivotek	1.5	Mikrotic	10.1	Metrohm	2.4	TP-Link
	3.1	Xerox	–	–	8.7	–	1.1	Asus	5.6	iStor	0.6	Eltex
East Europe	42.4	Kyocera	53.1	ICP	31.6	Axis	45.8	TP-Link	35.0	PLUS	60.5	D-Link
	25.9	Ricoh	18.2	QNAP	20.3	Ezvis	14.6	ZTE	26.5	Dahua	18.9	Huawei
	23.6	HP	12.5	W Digital	12.5	Vivotek	11.6	Technicolor	12.3	Metrohm	8.6	TP-Link
	3.7	Sharp	3.0	WD	9.0	Zhongxi	8.3	Mikrotic	5.0	Cisco	2.8	Zyxel
	2.4	FoxConn	1.7	La Cie	4.3	D-Link	7.5	Sagemcom	2.5	iStor	1.8	ZTE
West Europe	27.9	Kyocera	49.2	ICP	49.7	Axis	40.8	TP-Link	35.5	PLUS	65.6	Zyxel
	22.2	HP	17.1	W Digital	11.0	Vivotek	20.6	Arcadyan	20.9	Dahua	15.1	TP-Link
	18.8	Ricoh	8.5	QNAP	10.8	Advance Vision	12.4	Technicolor	14.0	Metrohm	13.2	ZTE
	9.5	Sharp	4.1	WD	4.7	D-Link	6.9	AVM	5.8	iStor	0.9	–
	3.5	FoxConn	3.8	Synology	3.9	Hikvision	4.7	Mikrotic	5.0	–	0.8	Winstars

Table A.3: **Vendors with Weak FTP and Telnet Credentials by Region, 1**—We show the top five vendors in each device type by region that exhibit weak FTP or Telnet credentials. In most cases, a small handful of vendors are responsible for most of the weak devices.

Region	FTP								Telnet			
	Work Appliance		Storage		Surveillance		Home Router		Surveillance		Home Router	
South Asia	51.4	HP	32.4	W Digital	61.5	Matrix	34.7	ZTE	42.1	PLUS	40.3	Smartlink
	19.6	Ricoh	17.6	QNAP	11.5	Axis	26.4	TP-Link	18.4	Dahua	34.7	D-Link
	10.5	Canon	14.7	WD	10.3	D-Link	12.4	D-Link	11.3	Metrohm	5.4	Huawei
	5.6	Kyocera	14.7	ICP	3.8	3DSP	6.8	Fiberhome	8.8	–	2.9	Fida
	4.2	FoxConn	8.8	–	2.6	CardioMEMS	3.0	Binatone	4.6	Cisco	2.6	Zyxel
S.E. Asia	46.6	Ricoh	39.9	ICP	45.1	Vivotek	62.3	TP-Link	45.7	PLUS	36.6	Huawei
	19.5	HP	25.4	QNAP	39.6	Axis	16.6	Mikrotic	16.2	Metrohm	24.6	Zyxel
	6.3	Sharp	11.6	W Digital	3.0	–	7.6	DrayTek	12.6	Dahua	12.3	TP-Link
	6.3	Kyocera	6.5	WD	2.4	Matrix	3.3	Sagemcom	5.3	Cisco	7.5	ZTE
	4.4	Xerox	4.3	I-O	1.2	Level One	1.8	D-Link	5.1	–	5.0	RicherLink
Oceania	21.1	Kyocera	65.1	ICP	30.8	Axis	57.6	TP-Link	35.8	PLUS	91.4	TP-Link
	18.3	HP	15.1	W Digital	30.8	–	32.4	NetComm	18.9	Dahua	2.7	D-Link
	17.9	Ricoh	11.6	QNAP	30.8	Ezvis	3.6	D-Link	13.2	Metrohm	1.4	ZTE
	14.3	Xeros	2.3	–	7.7	Adaptive Recognition	1.8	Billion	9.4	–	0.9	NetComm
	8.7	Sharp	2.3	Cisco	0.0	UTC F&S	1.8	Billion	1.1	–	0.9	–
N. Africa, ME	35.1	Kyocera	58.7	ICP	34.8	Axis	81.9	TP-Link	48.7	PLUS	38.9	TP-Link
	24.1	HP	18.5	QNAP	19.1	Vivotek	5.7	ZTE	16.3	Metrohm	34.2	Zyxel
	23.7	Ricoh	11.4	W Digital	10.3	D-Link	4.8	Askey	11.8	Dahua	19.0	Huawei
	5.6	Sharp	4.9	WD	3.4	Level One	1.7	Boca	4.9	iStor	2.6	D-Link
	5.0	FoxConn	1.6	Xerox	2.9	SMD	0.8	Cameo	3.6	Cisco	1.1	AirTies
S-S Africa	32.1	HP	43.5	ICP	72.5	Axis	30.7	TP-Link	43.4	PLUS	60.0	Zyxel
	28.7	Kyocera	16.5	QNAP	16.7	Vivotek	28.0	D-Link	16.9	Dahua	17.4	Huawei
	26	Ricoh	11.8	W Digital	2.9	Hikvision	22.3	Mikrotic	14.0	Metrohm	7.3	TP-Link
	4.0	FoxConn	7.1	Xerox	2.0	Netcore	6.6	ZTE	5.9	–	4.9	Fida
	3.0	Sharp	5.9	Seagate	2.0	Bosch	4.2	Billion	4.4	iStor	2.2	ZTE

Table A.4: **Vendors with Weak FTP and Telnet Credentials by Region, 2**—We show the top five vendors in each device type by region that exhibit weak FTP or Telnet credentials. In most cases, a small handful of vendors are responsible for most of the weak devices.

REFERENCES

[1] "A guide to the internet of things," https://www.intel.com/content/dam/www/public/us/en/images/iot/guide-to-iot-infographic.png.

[2] "In-depth analysis of changes in world internet performance using the speedtest global index," https://www.speedtest.net/insights/blog/global-index-2019-internet-report/.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica et al., "A view of cloud computing," *Communications of the ACM*, 2010.

[4] "Over one third of us households will cut the cord by 2020," https://www.soda.com/news/over-one-third-of-us-households-will-cut-the-cord-by-2020/.

[5] H. Mohajeri Moghaddam, G. Acar, B. Burgess, A. Mathur, D. Y. Huang, N. Feamster, E. W. Felten, P. Mittal, and A. Narayanan, "Watching you watch: The tracking ecosystem of over-the-top tv streaming devices," in *ACM Conference on Computer and Communications Security*, 2019.

[6] D. Kumar, R. Paccagnella, P. Murley, E. Hennenfent, J. Mason, A. Bates, and M. Bailey, "Skill squatting attacks on Amazon Alexa," in *27th USENIX Security Symposium*, 2018.

[7] M. . Company, "Unlocking the potential of the internet of things," https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world,.

[8] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast Internet-wide scanning and its security applications," in *USENIX Security Symposium*, 2013.

[9] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A search engine backed by Internet-wide scanning," in *22nd ACM Conference on Computer and Communications Security*, 2015.

[10] "Shodan internet scanner," https://www.shodan.io/.

[11] R. D. Graham, "Masscan: Mass ip port scanner," https://github.com/robertdavidgraham/masscan.

[12] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring internet denial-of-service activity," *ACM Transactions on Computer Systems (TOCS)*, 2006.

[13] L. F. DeKoven, A. Randall, A. Mirian, G. Akiwate, A. Blume, L. K. Saul, A. Schulman, G. M. Voelker, and S. Savage, "Measuring security practices and how they impact security," in *ACM Internet Measurement Conference*, 2019.

[14] X. Feng, Q. Li, H. Wang, and L. Sun, "Acquisitional rule-based engine for discovering internet-of-things devices," in *27th USENIX Security Symposium*, 2018.

[15] D. Y. Huang, N. Apthorpe, G. Acar, F. Li, and N. Feamster, "Iot inspector: Crowdsourcing labeled network traffic from smart home devices at scale," *ACM Conference on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2020.

[16] T. OConnor, R. Mohamed, M. Miettinen, W. Enck, B. Reaves, and A.-R. Sadeghi, "Homesnitch: behavior transparency and control for smart home iot devices," in *Conference on Security and Privacy in Wireless and Mobile Networks*, 2019.

[17] S. Goutam, W. Enck, and B. Reaves, "Hestia: simple least privilege network policies for smart homes," in *Conference on Security and Privacy in Wireless and Mobile Networks*, 2019.

[18] B. Krebs, "Krebsonsecurity hit with record DDoS," https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/.

[19] O. Klaba, "Octave klaba Twitter," https://twitter.com/olesovhcom/status/778830571677978624.

[20] S. Hilton, "Dyn analysis summary of Friday October 21 attack," http://hub.dyn.com/dyn-blog/dyn-analysis-summary-of-friday-october-21-attack.

[21] A. Tellez, "Bashlite," https://github.com/anthonygtellez/BASHLITE.

[22] Internet Census 2012, "Port scanning/0 using insecure embedded devices," http://internetcensus2012.bitbucket.org/paper.html.

[23] M. Mimoso, "IoT botnets are the new normal of DDoS attacks," https://threatpost.com/iot-botnets-are-the-new-normal-of-ddos-attacks/121093/.

[24] @unixfreaxjp, "Mmd-0056-2016 - Linux/Mirai, how an old ELF malcode is recycled." http://blog.malwaremustdie.org/2016/08/mmd-0056-2016-linuxmirai-just.html.

[25] OVH, "The DDoS that didn't break the camel's VAC*," https://www.ovh.com/us/news/articles/a2367.the-ddos-that-didnt-break-the-camels-vac.

[26] B. Krebs, "Did the Mirai botnet really take Liberia offline?" https://krebsonsecurity.com/2016/11/did-the-mirai-botnet-really-take-liberia-offline/.

[27] B. Krebs, "Who is Anna-Senpai, the Mirai worm author?" https://krebsonsecurity.com/2017/01/who-is-anna-senpai-the-mirai-worm-author/.

[28] Anna-senpai, "[FREE] world's largest net:Mirai botnet, client, echo loader, CNC source code release," https://hackforums.net/showthread.php?tid=5420472.

[29] J. Blackford and M. Digdon, "TR-069 issue 1 amendment 5," https://www.broadband-forum.org/technical/download/TR-069_Amendment-5.pdf.

[30] B. Krebs, "New Mirai worm knocks 900k Germans offline," https://krebsonsecurity.com/2016/11/new-mirai-worm-knocks-900k-germans-offline/.

[31] BBC, "Router hacker suspect arrested at Luton airport," http://www.bbc.com/news/technology-37510502.

[32] EvoSec, "New IoT malware? anime/kami," https://evosec.eu/new-iot-malware/.

[33] P. Muncaster, "Massive Qbot botnet strikes 500,000 machines through WordPress," https://www.infosecurity-magazine.com/news/massive-qbot-strikes-500000-pcs/.

[34] G. Lyon, "Nmap network scanning," https://nmap.org/book/vscan-fileformat.html.

[35] N. Wells, "Busybox: A swiss army knife for linux."

[36] VirusTotal, "Virustotal - free online virus, malware, and url scanner," https://virustotal.com/en.

[37] A. Kountouras, P. Kintis, C. Lever, Y. Chen, Y. Nadji, D. Dagon, M. Antonakakis, and R. Joffe, "Enabling network security through active DNS datasets," in *19th International Research in Attacks, Intrusions, and Defenses Symposium*, 2016.

[38] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a dynamic reputation system for DNS," in *19th USENIX Security Symposium*, 2010.

[39] B. Krebs, "Who makes the IoT things under attack?" https://krebsonsecurity.com/2016/10/who-makes-the-iot-things-under-attack/.

[40] B. Herzberg, D. Bekerman, and I. Zeifman, "Breaking down mirai: An IoT DDoS botnet analysis," https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html.

[41] Level 3, "How the grinch stole IoT," http://www.netformation.com/level-3-pov/how-the-grinch-stole-iot.

[42] MalwareTech, "Mapping Mirai: A botnet case study," https://www.malwaretech.com/2016/10/mapping-mirai-a-botnet-case-study.html.

[43] C. Lever, P. Kotzias, D. Balzarotti, J. Caballero, and M. Antonakakis, "A Lustrum of malware network communication: Evolution and insights," in *38th IEEE Symposium on Security and Privacy*, 2017.

[44] WikiDevi, "Eltel et-5300," https://wikidevi.com/wiki/Eltel_ET-5300#Stimulating_port_5555_.28from_Internet.29.

[45] C. Lever, R. Walls, Y. Nadji, D. Dagon, P. McDaniel, and M. Antonakakis, "Domain-Z: 28 registrations later," in *37th IEEE Symposium on Security and Privacy*, 2016.

[46] Arbor Networks, "Worldwide infrastructure security report," https://www.arbornetworks.com/images/documents/WISR2016_EN_Web.pdf.

[47] C. Rossow, "Amplification hell: Revisiting network protocols for DDoS abuse." in *21st Network and Distributed System Security Symposium*, 2014.

[48] M. Karami, Y. Park, and D. McCoy, "Stress testing the booters: Understanding and undermining the business of DDoS services," in *25th International Conference on World Wide Web*, 2016.

[49] E. Wustrow, M. Karir, M. Bailey, F. Jahanian, and G. Huston, "Internet background radiation revisited," in *10th ACM Internet Measurement Conference*, 2010.

[50] M. Karami and D. McCoy, "Understanding the emerging threat of DDoS-as-a-service," in *6th USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 2013.

[51] Minecraft Modern Wiki, "Protocol handshaking," http://wiki.vg/Protocol#Handshaking.

[52] Gamepedia Minecraft Wiki, "Tutorials/setting up a server," http://minecraft.gamepedia.com/Tutorials/Setting_up_a_server.

[53] D. Moore, C. Shannon, and K. Claffy, "Code-Red: A case study on the spread and victims of an Internet worm," in *2nd ACM Internet Measurment Workshop*, 2002.

[54] H. Asghari, M. Ciere, and M. J. G. Van Eeten, "Post-mortem of a zombie: Conficker cleanup after six years," in *24th USENIX Security Symposium*, 2015.

[55] M. Finifter, D. Akhawe, and D. Wagner, "An empirical study of vulnerability rewards programs." in *22nd USENIX Security Symposium*, 2013.

[56] D. Pauli, "Netgear unveils world's easiest bug bounty," http://www.theregister.co.uk/2017/01/06/netgear_unveils_worlds_easiest_bug_bounty/.

[57] Z. Durumeric, J. Kasten, D. Adrian, J. A. Halderman, M. Bailey, F. Li, N. Weaver, J. Amann, J. Beekman, M. Payer et al., "The matter of Heartbleed," in *14th ACM Internet Measurement Conference*, 2014.

[58] F. Li, Z. Durumeric, J. Czyz, M. Karami, M. Bailey, D. McCoy, S. Savage, and V. Paxson, "You've got vulnerability: Exploring effective vulnerability notifications," in *25th USENIX Security Symposium*, 2016.

[59] F. Li, G. Ho, E. Kuan, Y. Niu, L. Ballard, K. Thomas, E. Bursztein, and V. Paxson, "Remedying web hijacking: Notification effectiveness and webmaster comprehension," in *25th International Conference on World Wide Web*, 2016.

[60] A. Froehlich, "8 IoT operating systems powering the future," http://www.informationweek.com/iot/8-iot-operating-systems-powering-the-future/d/d-id/1324464.

[61] Microsoft, "Support for Windows XP ended," https://www.microsoft.com/en-us/WindowsForBusiness/end-of-xp-support.

[62] E. Cooke, F. Jahanian, and D. McPherson, "The zombie roundup: Understanding, detecting, and disrupting botnets," in *1st USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop*, 2005.

[63] A. Zand, G. Vigna, X. Yan, and C. Kruegel, "Extracting probable command and control signatures for detecting botnets," in *29th ACM Symposium on Applied Computing*, 2014.

[64] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: Clustering analysis of network traffic for protocol-and structure-independent botnet detection." in *17th USENIX Security Symposium*, 2008.

[65] G. Gu, J. Zhang, and W. Lee, "Botsniffer: Detecting botnet command and control channels in network traffic," in *15th Network and Distributed System Security Symposium*, 2008.

[66] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From throw-away traffic to bots: Detecting the rise of DGA-based malware," in *21st USENIX Security Symposium*, 2012.

[67] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling, "Measurements and mitigation of peer-to-peer-based botnets: A case study on Storm worm," in *1st USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 2008.

[68] J. Wyke, "The ZeroAccess botnet: Mining and fraud for massive financial gain," *Sophos Technical Paper*, 2012.

[69] P. Porras, H. Saïdi, and V. Yegneswaran, "A foray into Conficker's logic and rendezvous points," in *2nd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More*, 2009.

[70] P. Sinha, A. Boukhtouta, V. H. Belarde, and M. Debbabi, "Insights from the analysis of the Mariposa botnet," in *5th Conference on Risks and Security of Internet and Systems*, 2010.

[71] P. Barford and V. Yegneswaran, *An Inside Look at Botnets*, 2007. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-44599-1_8

[72] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: analysis of a botnet takeover," in *16th ACM conference on Computer and Communications Security*, 2009.

[73] D. Wang, S. Savage, and G. M. Voelker, "Juice: A longitudinal study of an SEO campaign," in *20th Network and Distributed Systems Security Symposium*, 2013.

[74] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir, "A survey of botnet technology and defenses," in *Cybersecurity Applications & Technology Conference For Homeland Security*, 2009.

[75] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in *6th ACM Internet Measurement Conference*, 2006.

[76] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *SIGCOMM Computer Communications Review*.

[77] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher, *Internet Denial of Service: Attack and Defense Mechanisms (Radia Perlman Computer Networking and Security)*. Prentice Hall PTR, 2004.

[78] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring internet denial-of-service activity," *ACM Transactions on Computer Systems (TOCS)*, 2006.

[79] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir, "Taming the 800 pound gorilla: The rise and decline of NTP DDoS attacks," in *14th ACM Internet Measurement Conference*, 2014.

[80] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, "Exit from hell? reducing the impact of amplification DDoS attacks." in *23rd USENIX Security Symposium*, 2014.

[81] B. Schneier, "The Internet of Things is wildly insecure–and often unpatchable," https://www.schneier.com/essays/archives/2014/01/the_internet_of_thin.html.

[82] A. Costin, J. Zaddach, A. Francillon, and D. Balzarotti, "A large-scale analysis of the security of embedded firmwares," in *23rd USENIX Security Symposium*, 2014.

[83] A. Costin, A. Zarras, and A. Francillon, "Automated dynamic firmware analysis at scale: A case study on embedded web interfaces," in *11th ACM Asia Conference on Computer and Communications Security*, 2016.

[84] E. Fernandes, J. Jung, and A. Prakash, "Security analysis of emerging smart home applications," in *37th IEEE Symposium on Security and Privacy*, 2016.

[85] E. Ronen, C. O'Flynn, A. Shamir, and A.-O. Weingarten, "IoT goes nuclear: Creating a ZigBee chain reaction."

[86] C. O'Flynn, "A lightbulb worm? a teardown of the philips hue," Blackhat Security Conference.

[87] L. Franceschi-Bicchierai, "Hackers makes the first-ever ransomware for smart thermostats," https://motherboard.vice.com/en_us/article/internet-of-things-ransomware-smart-thermostat.

[88] Level 3, "Attack of things!" http://www.netformation.com/level-3-pov/attack-of-things-2.

[89] X. Mertens, "Analyze of a Linux botnet client source code," https://isc.sans.edu/forums/diary/Analyze+of+a+Linux+botnet+client+source+code/21305.

[90] M. Malik and M.-E. M. Léveillé, "Meet Remaiten–a Linux bot on steroids targeting routers and potentially other IoT devices," http://www.welivesecurity.com/2016/03/30/meet-remaiten-a-linux-bot-on-steroids-targeting-routers-and-potentially-other-iot-devices/.

[91] E. Fernandes, J. Paupore, A. Rahmati, D. Simionato, M. Conti, and A. Prakash, "Flowfence: Practical data protection for emerging IoT application frameworks," in *25th USENIX Security Symposium*, 2016.

[92] L. H. Newman, "An elaborate hack shows how much damage iot bugs can do," https://www.wired.com/story/elaborate-hack-shows-damage-iot-bugs-can-do/.

[93] O. Williams-Grut, "Hackers stole a casino's database through a thermometer in the lobby fish tank," https://www.businessinsider.com/hackers-stole-a-casinos-database-through-a-thermometer-in-the-lobby-fish-tank-2018-4.

[94] E. Fernandes, J. Jung, and A. Prakash, "Security analysis of emerging smart home applications," in *37th IEEE Symposium on Security and Privacy*, 2016.

[95] A. Muravitsky, V. Dashchenko, and R. Sako, "Iot hack: how to break a smart home again," https://securelist.com/iot-hack-how-to-break-a-smart-home-again/84092/.

[96] Y. J. Jia, Q. A. Chen, S. Wang, A. Rahmati, E. Fernandes, Z. M. Mao, A. Prakash, and S. J. Unviersity, "Contexlot: Towards providing contextual integrity to appified iot platforms." in *24th Network and Distributed Systems Security Symposium*, 2017.

[97] E. Fernandes, J. Paupore, A. Rahmati, D. Simionato, M. Conti, and A. Prakash, "Flowfence: Practical data protection for emerging iot application frameworks." in *25th USENIX Security Symposium*, 2016.

[98] W. He, M. Golla, R. Padhi, J. Ofek, M. Dürmuth, E. Fernandes, and B. Ur, "Rethinking access control and authentication for the home internet of things," in *27th USENIX Security Symposium*, 2018.

[99] Q. Wang, W. U. Hassan, A. Bates, and C. Gunter, "Fear and logging in the internet of things," in *25th Networking and Distributed Systems Symposium*, 2018.

[100] OPSWAT, "Windows anti-malware market share report," https://metadefender.opswat.com/reports/anti-malware-market-share.

[101] IEEE, "Registration authority," https://standards.ieee.org/products-services/regauth/oui/index.html.

[102] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis et al., "Understanding the Mirai botnet," in *26th USENIX Security Symposium*, 2017.

[103] L. DiCioccio, R. Teixeira, M. May, and C. Kreibich, "Probe and pray: Using UPnP for home network measurements," in *13th International Conference on Passive and Active Network Measurement*, 2012.

[104] Wikipedia, "ISO-3166-2," https://en.wikipedia.org/wiki/ISO_3166-2.

[105] Z. Durumeric, M. Bailey, and J. A. Halderman, "An Internet-wide view of Internet-wide scanning," in *23rd USENIX Security Symposium*, 2014.

[106] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A search engine backed by Internet-wide scanning," in *22nd ACM Conference on Computer and Communications Security*, 2015.

[107] J. Cohen, "Statistical power analysis for the behavioral sciences," 1998.

[108] Amazon, "All things alexa," https://www.amazon.com/Amazon-Echo-And-Alexa-Devices.

[109] Google, "Google home," https://store.google.com/au/product/google_home.

[110] Nest Labs, "Nest thermostat," https://nest.com/thermostats/.

[111] Belkin, "WeMo smart plug," https://www.belkin.com/us/p/P-F7C063/.

[112] Philips, "Philips hue," https://www2.meethue.com/en-us.

[113] Ecobee, "Ecobee 4," https://www.ecobee.com/ecobee4/.

[114] F. HALAIS, "Spectacle and surveillance in brazil," https://www.opendemocracy.net/opensecurity/flavie-halais/spectacle-and-surveillance-in-brazil.

[115] M. M. Kanashiro, "Surveillance cameras in brazil: exclusion, mobility regulation, and the new meanings of security," *Surveillance & Society*, 2008.

[116] D. M. Corey, W. P. Dunlap, and M. J. Burke, "Averaging correlations: Expected values and bias in combined pearson rs and fisher's z transformations," *The Journal of general psychology*, vol. 125, no. 3, 1998.

[117] L. H. Newman, "The ransomware meltdown experts warned about is here," https://www.wired.com/2017/05/ransomware-meltdown-experts-warned/.

[118] F. T. Commission, "FTC charges D-Link put consumers' privacy at risk due to the inadequate security of its computer routers and cameras," https://www.ftc.gov/news-events/press-releases/2017/01/ftc-charges-d-link-put-consumers-privacy-risk-due-inadequate.

[119] J. Wilson, R. S. Wahby, H. Corrigan-Gibbs, D. Boneh, P. Levis, and K. Winstein, "Trust but verify: Auditing the secure Internet of things," in *15th Annual International Conference on Mobile Systems, Applications, and Services*, 2017.

[120] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson, "Netalyzr: illuminating the edge network," in *10th Internet Measurement Conference*, 2010.

[121] M. Chetty, D. Haslem, A. Baird, U. Ofoha, B. Sumner, and R. Grinter, "Why is my internet slow?: making network speeds visible," in *29th SIGCHI Conference on human factors in computing systems*, 2011.

[122] L. DiCioccio, R. Teixeira, and C. Rosenberg, "Measuring home networks with homenet profiler," in *14th International Conference on Passive and Active Network Measurement*, 2013.

[123] M. A. Sánchez, J. S. Otto, Z. S. Bischof, and F. E. Bustamante, "Trying broadband characterization at home," in *14th International Conference on Passive and Active Network Measurement*, 2013.

[124] B. Agarwal, R. Bhagwan, T. Das, S. Eswaran, V. N. Padmanabhan, and G. M. Voelker, "Netprints: Diagnosing home network misconfigurations using shared knowledge." in *9th USENIX Networked Systems Design and Implementation Conference*, 2009.

[125] M. Chetty, J.-Y. Sung, and R. E. Grinter, "How smart homes learn: The evolution of the networked home and household," in *9th International Conference on Ubiquitous Computing*, 2007.

[126] S. Grover, M. S. Park, S. Sundaresan, S. Burnett, H. Kim, B. Ravi, and N. Feamster, "Peeking behind the NAT: an empirical study of home networks," in *13th ACM Internet Measurement Conference*, 2013.

[127] S. Sundaresan, S. Burnett, N. Feamster, and W. De Donato, "Bismark: A testbed for deploying measurements and applications in broadband access networks." in *19th USENIX Annual Technical Conference*, 2014.

[128] A. Sarabi and M. Liu, "Characterizing the internet host population using deep learning: A universal and lightweight numerical embedding," in *18th ACM Internet Measurement Conference*, 2018.

[129] X. Feng, Q. Li, Q. Han, H. Zhu, Y. Liu, J. Cui, and L. Sun, "Active profiling of physical devices at internet scale," in *25th International Conference on Computer Communication and Networks*, 2016.

[130] Z. Shamsi, A. Nandwani, D. Leonard, and D. Loguinov, "Hershel: single-packet os fingerprinting," in *6th ACM SIGMETRICS Conference*, 2014.

[131] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, and I. Ray, "Behavioral fingerprinting of iot devices," in *2nd ACM Workshop on Attacks and Solutions in Hardware Security*, 2018.

[132] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "Iot sentinel: Automated device-type identification for security enforcement in iot," in *37th International Conference on Distributed Computing Systems (ICDCS)*, 2017.

[133] G. Hernandez, O. Arias, D. Buentello, and Y. Jin, "Smart nest thermostat: A smart spy in your home," *Black Hat USA*, 2014.

[134] E. Ronen, A. Shamir, A.-O. Weingarten, and C. O'Flynn, "IoT goes nuclear: Creating a ZigBee chain reaction," in *38th IEEE Symposium on Security and Privacy (SP)*, 2017.

[135] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinattack: Inaudible voice commands," in *24th ACM Conference on Computer and Communications Security*, 2017.

[136] Y. Tian, N. Zhang, Y.-H. Lin, X. Wang, B. Ur, X. Guo, and P. Tague, "Smartauth: User-centered authorization for the Internet of things," in *26th USENIX Security Symposium*, 2017.

[137] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, "SoK: security evaluation of home-based iot deployments," in *40th IEEE Symposium on Security and Privacy*, 2019.

[138] A. Mirian, Z. Ma, D. Adrian, M. Tischer, T. Chuenchujit, T. Yardley, R. Berthier, J. Mason, Z. Durumeric, J. A. Halderman et al., "An internet-wide view of ics devices," in *14th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2016.

[139] B. Ghena, W. Beyer, A. Hillaker, J. Pevarnek, and J. A. Halderman, "Green lights forever: Analyzing the security of traffic infrastructure," in *8th USENIX Workshop on Offensive Technologies*, 2014.

[140] A. Bonkoski, R. Bielawski, and J. A. Halderman, "Illuminating the security issues surrounding lights-out server management," in *7th USENIX Workshop on Offensive Technologies*, 2013.

[141] D. Springall, Z. Durumeric, and J. A. Halderman, "FTP: The forgotten cloud," in *46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2016.

[142] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, "Exit from hell? reducing the impact of amplification ddos attacks," in *23rd USENIX Security Symposium*, 2014.

[143] N. Samarasinghe and M. Mannan, "Tls ecosystems in networked devices vs. web servers," in *International Conference on Financial Cryptography and Data Security*, 2017.

[144] J. Zhang, Z. Durumeric, M. Bailey, M. Liu, and M. Karir, "On the mismanagement and maliciousness of networks." in *Network and Distributed System Security Symposium*, 2014.

[145] M. Kührer, T. Hupperich, J. Bushart, C. Rossow, and T. Holz, "Going wild: Large-scale classification of open DNS resolvers," in *15th ACM Internet Measurement Conference*, 2015.

[146] M. Hastings, J. Fried, and N. Heninger, "Weak keys remain widespread in network devices," in *ACM Internet Measurement Conference*, 2016.

[147] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman, "Mining your Ps and Qs: Detection of widespread weak keys in network devices," in *21st USENIX Security Symposium*, 2012.

[148] D. Kumar, K. Shen, B. Case, D. Garg, G. Alperovich, D. Kuznetsov, R. Gupta, and Z. Durumeric, "All things considered: an analysis of iot devices on home networks," in {*USENIX*} *Security Symposium*, 2019.

[149] A. Sivanathan, H. H. Gharakheili, and V. Sivaraman, "Can we classify an iot device using tcp port scan?" in *IEEE International Conference on Information and Automation for Sustainability*, 2018.

[150] Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, N. O. Tippenhauer, and Y. Elovici, "Profiliot: a machine learning approach for iot device identification based on network traffic analysis," in *Symposium on applied computing*, 2017.

[151] H. Guo and J. Heidemann, "Detecting iot devices in the internet (extended)," *USC/ISI Technical Report ISI-TR-726 July*, 2018.

[152] M. H. Mazhar and Z. Shafiq, "Characterizing smart home iot traffic in the wild," *arXiv preprint arXiv:2001.08288*, 2020.

[153] Nmap, "Nmap security scanner," https://nmap.org/.

[154] Z. Durumeric, F. Li, J. Kasten, J. Amann, J. Beekman, M. Payer, N. Weaver, D. Adrian, V. Paxson, M. Bailey et al., "The matter of heartbleed," in *Internet Measurement Conference*, 2014.

[155] E. Al-Shaer, J. Wei, K. W. Hamlen, and C. Wang, "Honeyscope: Iot device protection with deceptive network views," in *Autonomous Cyber Deception*, 2019.

[156] Censys, "What does censys scan?" https://support.censys.io/hc/en-us/articles/360038762031.

[157] D. Mauro 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 *IEEE Security and Privacy Workshops (SPW)*, 2019.

[158] J. Cleary, S. Donnelly, I. Graham, A. McGregor, and M. Pearson, "Design principles for accurate passive measurement," in *Passive and Active Measurement Conference*, 2000.

[159] H. Martin, A. McGregor, and J. Cleary, "Analysis of internet delay times," in *Passive and Active Measurment Workshop*, 2000.

[160] G. Ho, A. Sharma, M. Javed, V. Paxson, and D. Wagner, "Detecting credential spearphishing in enterprise settings," in {*USENIX*} *Security Symposium*, 2017.

[161] V. Paxson, "Bro: a system for detecting network intruders in real-time," in *USENIX Security Symposium*, 1998.

[162] M. Bailey, E. Cooke, F. Jahanian, A. Myrick, and S. Sinha, "Practical darknet measurement," in *Annual Conference on Information Sciences and Systems*, 2006.

[163] N. Apthorpe, D. Y. Huang, D. Reisman, A. Narayanan, and N. Feamster, "Keeping the smart home private with smart (er) iot traffic shaping," *Proceedings on Privacy Enhancing Technologies*, 2019.

[164] Fingerbank, "Fingerbank," https://fingerbank.org/.

[165] G. Acar, D. Y. Huang, F. Li, A. Narayanan, and N. Feamster, "Web-based attacks to discover and control local iot devices," in *2018 Workshop on IoT Security and Privacy*.

[166] J. Chen, W. Diao, Q. Zhao, C. Zuo, Z. Lin, X. Wang, W. C. Lau, M. Sun, R. Yang, and K. Zhang, "Iotfuzzer: Discovering memory corruptions in iot through app-based fuzzing." in *Symposium on Networking and Distributed Systems Security*, 2018.

[167] Y. Zheng, A. Davanian, H. Yin, C. Song, H. Zhu, and L. Sun, "Firm-afl: high-throughput greybox fuzzing of iot firmware via augmented process emulation," in {*USENIX*} *Security Symposium*, 2019.

[168] J. Caballero, H. Yin, Z. Liang, and D. Song, "Polyglot: Automatic extraction of protocol message format using dynamic binary analysis," in *ACM conference on Computer and communications security*, 2007.

[169] A. M. Mandalari, R. Kolcun, H. Haddadi, D. J. Dubois, and D. Choffnes, "Towards automatic identification and blocking of non-critical iot traffic destinations," 2020.

[170] S. Demetriou, N. Zhang, Y. Lee, X. Wang, C. A. Gunter, X. Zhou, and M. Grace, "Hanguard: Sdn-driven protection of smart home wifi devices from malicious mobile apps," in *10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2017.