Understanding the Mirai Botnet

Manos Antonakakis[◇] Tim April[‡] Michael Bailey[†] Matthew Bernhard[◁] Elie Bursztein[○]
Jaime Cochran[▷] Zakir Durumeric[◁] J. Alex Halderman[◁] Luca Invernizzi[○]
Michalis Kallitsis[§] Deepak Kumar[†] Chaz Lever[◇] Zane Ma[†*] Joshua Mason[†]
Damian Menscher[○] Chad Seaman[‡] Nick Sullivan[▷] Kurt Thomas[○] Yi Zhou[†]

[‡]*Akamai Technologies* [▷]*Cloudflare* [◇]*Georgia Institute of Technology* [○]*Google*
[§]*Merit Network* [†]*University of Illinois Urbana-Champaign* [◁]*University of Michigan*

Abstract

The Mirai botnet, composed primarily of embedded and IoT devices, took the Internet by storm in late 2016 when it overwhelmed several high-profile targets with massive distributed denial-of-service (DDoS) attacks. In this paper, we provide a seven-month retrospective analysis of Mirai's growth to a peak of 600k infections and a history of its DDoS victims. By combining a variety of measurement perspectives, we analyze how the botnet emerged, what classes of devices were affected, and how Mirai variants evolved and competed for vulnerable hosts. Our measurements serve as a lens into the fragile ecosystem of IoT devices. We argue that Mirai may represent a sea change in the evolutionary development of botnets — the simplicity through which devices were infected and its precipitous growth, demonstrate that novice malicious techniques can compromise enough low-end devices to threaten even some of the best-defended targets. To address this risk, we recommend technical and non-technical interventions, as well as propose future research directions.

1 Introduction

Starting in September 2016, a spree of massive distributed denial-of-service (DDoS) attacks temporarily crippled Krebs on Security [46], OVH [43], and Dyn [36]. The initial attack on Krebs exceeded 600 Gbps in volume [46] — among the largest on record. Remarkably, this overwhelming traffic was sourced from hundreds of thousands of some of the Internet's least powerful hosts — Internet of Things (IoT) devices — under the control of a new botnet named Mirai.

While other IoT botnets such as BASHLITE [86] and Carna [38] preceded Mirai, the latter was the first to emerge as a high-profile DDoS threat. What explains Mirai's sudden rise and massive scale? A combination of factors — efficient spreading based on Internet-wide scanning, rampant use of insecure default passwords in IoT products, and the insight that keeping the botnet's behavior simple would allow it to infect many heterogeneous devices — all played a role. Indeed, Mirai has spawned many variants that follow the same infection strategy, leading to speculation that "IoT botnets are the new normal of DDoS attacks" [64].

In this paper, we investigate the precipitous rise of Mirai and the fragile IoT ecosystem it has subverted. We present longitudinal measurements of the botnet's growth, composition, evolution, and DDoS activities from August 1, 2016 to February 28, 2017. We draw from a diverse set of vantage points including network telescope probes, Internet-wide banner scans, IoT honeypots, C2 milkers, DNS traces, and logs provided by attack victims. These unique datasets enable us to conduct the first comprehensive analysis of Mirai and posit technical and non-technical defenses that may stymie future attacks.

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

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

[*]Denotes primary, lead, or "first" author

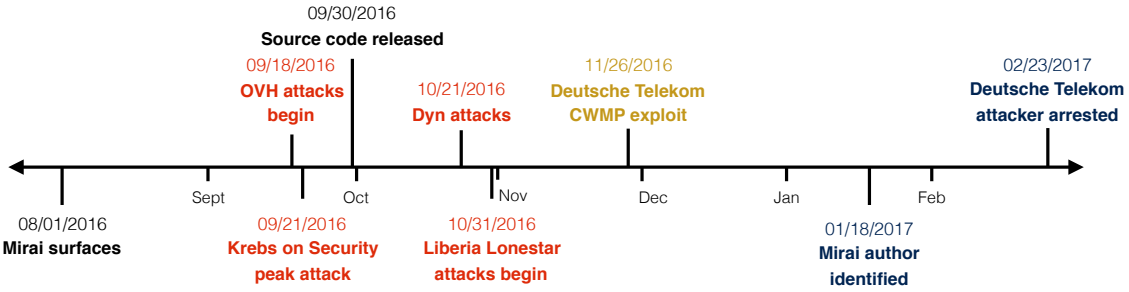


Figure 1: **Mirai Timeline**—Major attacks (red), exploits (yellow), and events (black) related to the Mirai botnet.

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

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

2 The Mirai Botnet

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

Timeline of events Reports of Mirai appeared as early as August 31, 2016 [89], though it was not until mid-September, 2016 that Mirai grabbed headlines with massive DDoS attacks targeting Krebs on Security [46] and OVH [74] (Figure 1). Several additional high-profile attacks later targeted DNS provider Dyn [36] and Lonestar Cell, a Liberian telecom [45]. In early 2017, the actors surrounding Mirai came to light as the Mirai author was identified [49]. Throughout our study, we corroborate our measurement findings with these media reports and expand on the public information surrounding Mirai.

Another significant event in this timeline is the public release of Mirai's source code on hackforums.net [4]. We rely on this code to develop our measurement methodology (Section 3). Furthermore, as we detail later (Section 5), this source code release led to the proliferation of Mirai variants with competing operators. One notable variant added support for a router exploit through CPE WAN Management Protocol (CWMP), an HTTP-based protocol that enables auto-configuration and remote management of home routers, modems, and other customer-premises equipment (CPE) [15]. This exploit led to an outage at Deutsche Telekom late November 2016 [47], with the suspected attacker later arrested in February 2017 [13]. In this work, we track Mirai's variants and examine how they influenced Mirai's propagation.

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

A separate *loader program* (③) asynchronously infected these vulnerable devices by logging in, determining the underlying system environment, and finally, downloading and executing architecture-specific malware (④). After a successful infection, Mirai attempted to conceal its presence by deleting the downloaded binary and obfuscating its process name in a pseudorandom alphanumeric string. As a consequence, Mirai infections did not persist across system reboots. In order to fortify itself, the malware additionally killed other processes bound to TCP/22 or TCP/23, as well as processes associated with competing infections, including other Mirai variants, `.anime` [25], and Qbot [72]. At this point, the bot

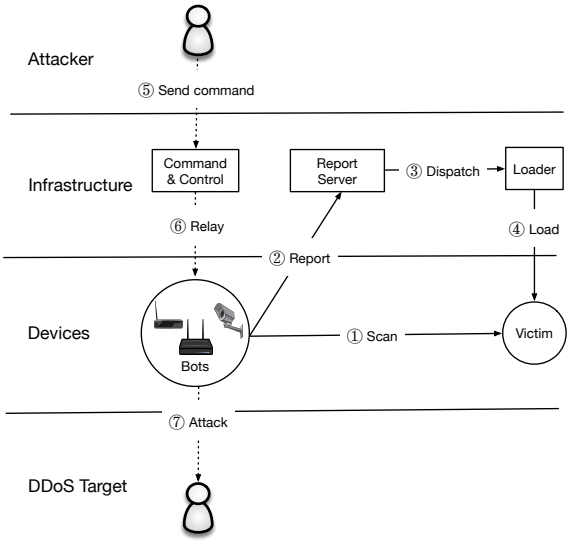


Figure 2: **Mirai Operation**—Mirai bots scan the IPv4 address space for devices that run telnet or SSH, and attempt to log in using a hardcoded dictionary of IoT credentials. Once successful, the bot sends the victim IP address and associated credentials to a report server, which asynchronously triggers a loader to infect the device. Infected hosts scan for additional victims and accept DDoS commands from a command and control (C2) server.

listened for attack commands from the command and control server (C2) while simultaneously scanning for new victims.

Malware phylogeny While not directly related to our study, the Mirai family represents an evolution of BASHLITE (otherwise known as LizardStresser, Torlus, Gafgyt), a DDoS malware family that infected Linux devices by brute forcing default credentials [86]. BASHLITE relied on six generic usernames and 14 generic passwords, while the released Mirai code used a dictionary of 62 username/password pairs that largely subsumed BASHLITE's set and added credentials specific to consumer routers and IoT devices. In contrast to BASHLITE, Mirai additionally employed a fast, stateless scanning module that allowed it to more efficiently identify vulnerable devices.

3 Methodology

Our study of Mirai leverages a variety of network vantage points: a large, passive network telescope, Internet-wide scanning, active Telnet honeypots, logs of C2 attack commands, passive DNS traffic, and logs from DDoS attack targets. In this section, we discuss our data sources and the role they play in our analysis. We provide a high-level summary in Table 1.

3.1 Network Telescope

Mirai's indiscriminate, rapid scanning strategy lends itself to tracking the botnet's propagation to new hosts. We monitored all network requests to a network telescope [9] composed of 4.7 million IP address operated by Merit Network over a seven month period from July 18, 2016 to February 28, 2017. On average, the network telescope received 1.1 million packets from 269,000 IP addresses per minute during this period. To distinguish Mirai traffic from background radiation [94] and other scanning activity, we uniquely fingerprinted Mirai probes based on an artifact of Mirai's stateless scanning whereby every probe has a TCP sequence number—normally a random 32-bit integer—equal to the destination IP address. The likelihood of this occurring incidentally is $1/2^{32}$, and we would expect to see roughly 86 packets demonstrating this pattern in our entire dataset. In stark contrast, we observed 116.2 billion Mirai probes from 55.4 million IP addresses. Prior to the emergence of Mirai, we observed only three IPs that perform scans with this fingerprint. Two of the IP addresses generated five packets; two on TCP/80 and three on TCP/1002. The third IP address belongs to Team Cymru [1], who conducts regular TCP/443 scans.

We caution that the raw count of IP addresses seen scanning over time is a poor metric of botnet size due to DHCP churn [87]. To account for this, we tracked the size of the botnet by considering the number of hosts actively "scanning" at the start of every hour. We detected scans using the methodology presented by Durumeric et al. [23], in which we group packets from a single IP address in a temporal window into logical scans. We specifically identified scans that targeted the IPv4 address space at an estimated rate of at least five packets per second, expiring inactive scans after 20 minutes. We geolocated IPs using Maxmind [61].

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

Table 2: **Devices Identified**—We identified device type, model, and/or vendor for 31.5% of active scan banners. Protocol banners varied drastically in device identifiability, with HTTPS certificates being most descriptive, and SSH prompts being the least.

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

Table 1: **Data Sources** — We utilized a multitude of data perspectives to empirically analyze the Mirai botnet.

3.2 Active Scanning

While Mirai is widely considered an IoT botnet, there has been little comprehensive analysis of infected devices over the botnet's entire lifetime. In order to determine the manufacturer and model of devices infected with Mirai, we leveraged Censys [22], which actively scans the IPv4 space and aggregates application layer data about hosts on the Internet. We focused our analysis on scans of HTTPS, FTP, SSH, Telnet, and CWMP between July 19, 2016 and February 28, 2017.

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

Post-filtering, our dataset included 1.8 million banners associated with 1.2 million Mirai-infected IP addresses (Table 2). We had the most samples for CWMP, and the least for SSH. We caution that devices with open services that are not closed by Mirai (e.g., HTTPS and FTP) can appear repeatedly in Censys banner scans during our measurement window (due to churn) and thus lead to over counting when compared across protocols. As such, we intentionally explored protocols in isolation from one another and limited ourselves to measurements that only consider relative proportions rather than absolute counts of infected hosts.

Finally, we processed each infected device's banner to identify the device manufacturer and model. We first applied the set of regular expressions used by Nmap service probes to fingerprint devices [58]. Nmap successfully handled 98% of SSH banners and 81% of FTP banners, but matches only 7.8% of the Telnet banners. In order to increase our coverage and also accommodate HTTPS and CWMP (which Nmap lacks probes for), we constructed our own regular expressions to map banners to device manufacturers and models. Unfortunately, we found that in many cases, there was not enough data to identify a model and manufacturer from FTP, Telnet, CWMP, and SSH banners and that Nmap fingerprints only provide generic descriptions. In total, we identified device type and/or model and manufacturer for 31.5% of banners (Table 2). We caution that this methodology is susceptible to misattribution in instances where port-forwarding and Universal Plug and Play (UPnP) are used to present multiple devices behind a single IP address, making the distinction between middlebox and end-device difficult.

3.3 Telnet Honeypots

To track the evolution of Mirai's capabilities, we collected binaries installed on a set of Telnet honeypots that masqueraded as vulnerable IoT devices. Mechanically, we presented a BusyBox shell [92] and IoT-consistent device banner. Our honeypots logged all incoming Telnet traffic and downloaded any binaries that attackers attempted to install on the host via `wget` or `tftp` (the methods of infection found in Mirai's original source). In order to avoid collateral damage, we blocked all other outgoing requests (e.g., scanning and DoS traffic).

We logged 80K connection attempts from 54K IP addresses between November 2, 2016 and February 28, 2017, collecting a total 151 unique binaries. We filtered out executables unrelated to Mirai based on a YARA signature that matched any of the strings from the original source code release, leaving us with 141 Mirai binaries. We supplemented this data with 293 binaries observed by

honeypots operated by Akamai, which served a similar purpose to ours, but were hosted on a different public cloud provider. As a final source of samples, we included 594 unique binaries from VirusTotal [90] that we scanned for using the YARA rules mentioned above. In total, we collected 1,028 unique Mirai samples.

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

3.4 Passive & Active DNS

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

From a large U.S. ISP, we obtained passive DNS data consisting of DNS queries generated by the ISP's clients and their corresponding responses. More specifically, we collected approximately 209 million resource records (RRs) — queried domain name, and associated RDATA — and their lookup volumes aggregated on a daily basis. For our active DNS dataset, we obtained 290 million RRs per day from Thales, an active DNS monitoring system [44]. Both datasets cover the period of August 1, 2016 to February 28, 2017.

Using both passive and active DNS datasets, we performed DNS *expansion* to identify shared DNS infrastructure by linking related historic domain names (RHDN) and related historic IPs (RHIPs) [5]. This procedure began with the seed set of C2 domains and IPs extracted during reverse engineering of our honeypotted binaries. For a given seed `foo.com`, we identified the IP addresses that `foo.com` previously resolved to and added them to a growing set of domains and IPs. We additionally performed the reverse analysis, starting from an IP and finding any domain names that concurrently resolved it. Thus, even from a single domain name, we iteratively expanded the set of related domain names and IP addresses to construct a graph reflecting the shared infrastructure used by Mirai variants. In total, we identified 33 unique DNS clusters that we explore in detail in Section 5.

3.5 Attack Commands

To track the DDoS attack commands issued by Mirai operators, Akamai ran a "milker" from September 27, 2016–February 28, 2017 that connected to the C2 servers found in the binaries uploaded to their honeypots. The service simulated a Mirai-infected device and communicated with the C2 server using a custom bot-to-C2 protocol, which was reverse engineered from malware samples prior to source code release. In total, Akamai observed 64K attack commands issued by 484 unique C2 servers (by IP address). We note that a naive analysis of attack commands overestimates the volume of attacks and targets: individual C2 servers often repeat the same attack command in rapid succession, and multiple distinct C2 servers frequently issued the same command. To account for this, we heuristically grouped attack commands along two dimensions: by shared C2 infrastructure and by temporal similarity. We collapsed matching commands (i.e., tuples of attack type, duration, targets, and command options) that occur within 90 seconds of each other, which yielded 15,194 attacks from 146 unique IP clusters. Our attack command coverage includes the Dyn attack [36] and Liberia attacks [45]. We did not observe attack commands for Krebs on Security and OVH, which occurred prior to the milker's operation.

3.6 DDoS Attack Traces

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

[1] We overlapped attack traces with every Mirai scanning IPs on our network telescope. The overlap may have been inflated by non-Mirai attack IPs being assigned to Mirai devices over time through DHCP churn.

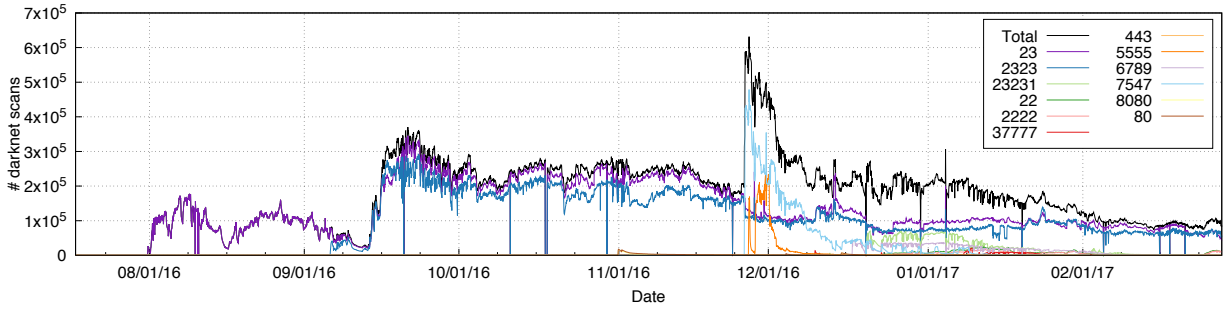


Figure 3: **Temporal Mirai Infections**—We estimate of the number of Mirai-infected devices over time by tracking the number of hosts actively scanning with Mirai fingerprint at the start of every hour. Mirai started by scanning Telnet, and variants evolved to target 11 additional protocols. The total population initially fluctuated between 200,000–300,000 devices before receding to 100,000 devices, with a brief peak of 600,000 devices.

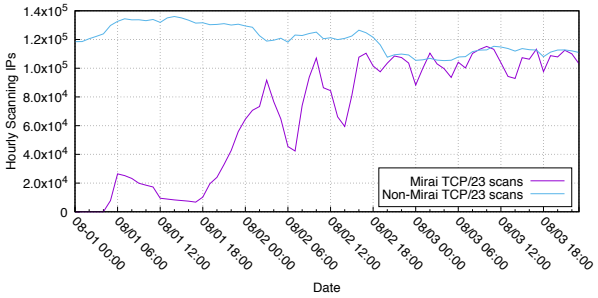


Figure 4: **Bootstrap Scanning**—Mirai scanning began on August 1, 2016 from a single IP address in a bulletproof hosting center. Mirai infection spread rapidly with a 76-minute doubling time and quickly matched the volume of non-Mirai Telnet scanning.

4 Tracking Mirai's Spread

As a first step towards understanding Mirai, we analyzed how the botnet bootstrapped its initial infections, what types of devices it targeted, and how it eventually infected an estimated 600K hosts. To contextualize the properties of Mirai, we compare it against prior botnets and worms.

4.1 Bootstrapping

We provide a timeline of Mirai's first infections in Figure 4. A single preliminary Mirai scan occurred on August 1, 2016 from an IP address belonging to DataWagon, a U.S.-based bulletproof hosting provider [48]. This bootstrap scan lasted approximately two hours (01:42–03:59 UTC), and about 40 minutes later (04:37 UTC) the Mirai botnet emerged. Within the first minute, 834 devices began scanning, and 11K hosts were infected within the first 10 minutes. Within 20 hours, Mirai infected 64,500 devices. Mirai's initial 75-minute doubling time is outstripped by other worms such as Code Red (37-minute doubling time [70]) and Blaster (9-minute doubling time [10]). Mirai's comparatively modest initial growth may be due to the low bandwidth and computational resources of infected devices, a consequence of the low-accuracy, brute-force login using a small number of credentials, or simply attributable to a bottleneck in loader infrastructure.

4.2 Steady State Size

We observed multiple phases in Mirai's life: an initial steady state of 200,000–300,000 infections in September 2016; a peak of 600,000 infections at the end of November 2016; and a collapse to roughly 100,000 infections at the end of our observation window in late February 2017 (Figure 3). Even though hosts were initially compromised via a simple dictionary attack, Mirai was able to infect hundreds of thousands of devices. This is similar in scale to historical botnets such as the prolific Srizbi spam botnet (400,000 bots [83]), which was responsible for more than half of all global botnet spam [35], and the Carna botnet (420,000 bots [38]), the first botnet of IoT devices compromised using default credentials.

While the original Mirai variant infected devices by attempting Telnet and SSH logins with a static set of credentials, later strains evolved to scan for other types of vulnerabilities. Most notably, Mirai-fingerprinted scans targeting TCP/7547, the standard port for CWMP, began appearing in our dataset on November 26, 2016. Mirai compromised CWMP devices through an RCE exploit in a SOAP configuration endpoint [41]. The new attack vector led to a renewed spike of infections (Figure 3). The decay that followed may be explained best by Deutsche Telekom patching routers soon after the attack [21]. The non-immediate decay may have been due to the devices requiring a reboot for the patch to take effect.

To better understand the decrease in Mirai bots from a steady state of 300,000 devices down to 100,000 devices, we examined the ASes in which raw population

decreased most significantly between September 21, 2016 and February 28, 2017. The ASes with the largest reduction in devices were: Telefónica Colombia (−38,589 bots, −98.5%), VNPT Corp (−16,791 bots, −90.2%), and Claro S.A. (−14,150 bots, −80.2%). This suggests potential action by certain network operators to mitigate Mirai. While a handful of ASes increased in prevalence over time, notably Telefónica de Argentina (+3,287 bots, 3,365.1%) and Ecuadorian telecom company CNT EP (+1,447 bots, 116.4%), the total increase (+10,500 bots) across all ASes is eclipsed by the overall decrease (−232,698 bots).

Country	Mirai Infections	Mirai Prevalence	Telnet Prevalence
Brazil	49,340	15.0%	7.9%
Colombia	45,796	14.0%	1.7%
Vietnam	40,927	12.5%	1.8%
China	21,364	6.5%	22.5%
S. Korea	19,817	6.0%	7.9%
Russia	15,405	4.7%	2.7%
Turkey	13,780	4.2%	1.1%
India	13,357	4.1%	2.9%
Taiwan	11,432	3.5%	2.4%
Argentina	7,164	2.2%	0.2%

Table 3: **Geographic Distribution**— We compare countries that harbored the most infections on 09/21/2016—when Krebs on Security was attacked—with countries that hosted the most telnet devices on 07/19/2016 prior to Mirai's onset. Mirai infections occurred disproportionately in South America and Southeast Asia, accounting for 50% of infections.

4.3 Global Distribution

In order to understand where Mirai infections were geographically concentrated, we calculated the geolocation of Mirai bots actively scanning at 00:00 UTC on September 21, 2016 (during the first Krebs on Security attack and Mirai's peak steady state infection period). As shown in Figure 3, the bulk of Mirai infections stemmed from devices located in Brazil (15.0%), Columbia (14.0%), and Vietnam (12.5%). Mirai also exhibited a concentrated network distribution—the top 10 ASes accounted for 44.3% of infections, and the top 100 accounted for 78.6% of infections (Table 4). Compared to the pre-Mirai global distribution of telnet hosts, Mirai consisted of a disproportionate number of devices concentrated in South America and Southeast Asia. This is possibly due to biases in manufacturer and market penetration in those regions. This is a stark contrast from many prior worms, which were primarily concentrated in the U.S., including CodeRed (43.9%), Slammer (42.9%), Witty (26.3%), and Conficker (34.5%) [82]. Mirai largely infected regions the black market considers to be low-quality hosts used for proxies and DDoS [88] and may have limited potential avenues for monetization.

AS	%	AS	%
Telefónica Colombia	11.9%	Türk Telekom	3.2%
VNPT Corp.	5.7%	Chunghwa Telecom†	2.9%
Claro S.A.	5.4%	FPT Group	2.8%
China Telecom†	4.0%	Korea Telecom†	2.6%
Telefônica Brasil	3.4%	Viettel Corporation	2.5%

Table 4: **AS Distribution**—We list the 10 ASes with the largest number of infections on 09/21/2016, the day Krebs on Security was attacked and the initial peak infection. The top 10 ASes accounted for 44.3% of infections, but only three of the top 10 are within the top 100 global ASes (denoted †) [16].

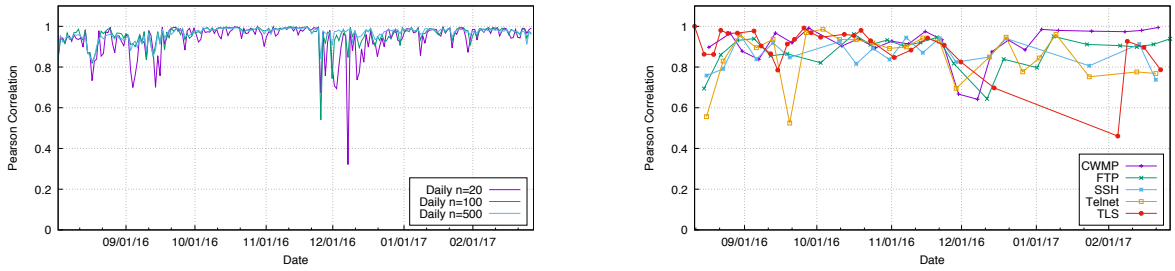
We explored the dynamism of Mirai's membership by examining the correlation between the top Mirai scanning ASes over time. We find that Mirai displayed general stability outside of the rapid growth phase in September 2016 and when CWMP exploits were introduced in late November (Figure 5a). During the September growth period, the number of IPs in each AS rose across the board with a few outliers. The growth of IPs belonging to Telefónica Colombia exceeded all other ASes and was eventually responsible for the largest number of Mirai infections. Other new introductions to the top 10 included India's Bharti Airtel and Bharat Sanchar Nigam Limited, Brazil's Claro S.A., and Korea Telecom.

CWMP emergence also disrupted general network distribution stability. Between November 25–27, 7 of the top 10 ASes decreased in rank to give rise to several previously unseen European ASes (e.g., Eircom and TalkTalk). Their appearance was short-lived; by December 10, 2016, these ASes fell back down in population. This suggests that the vulnerable population of the CWMP exploit were concentrated in Europe, but prompt patching returned Mirai back to its original concentration in South America and Southeast Asia. The longterm stability of Mirai ASes and geolocation demonstrates that Mirai has not expanded significantly in the scope and scale of devices that it infects. However, as the transient CWMP exploit demonstrates, new infection vectors had the potential to quickly add to Mirai's already sizable membership.

4.4 Device Composition

While cursory evidence suggested that Mirai targets IoT devices—Mirai's dictionary of default usernames and passwords included routers, DVRs, and cameras [50], and its source compiled to multiple embedded hardware configurations—we provide an in-depth analysis of both the intended device targets and successful infections.

To understand the types of devices that Mirai targeted,



(a) **AS Stability** (b) **Device Stability**

Figure 5: **Stability of Measured Properties**—From the temporal Pearson correlation of ASes (a) and device labels (b), we found that our measurements were largely stable despite external factors like DHCP churn. Rapid growth of CWMP-based infections in late November caused instability but calmed shortly thereafter.

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

Table 5: **Default Passwords**—The 09/30/2016 Mirai source release included 46 unique passwords, some of which were traceable to a device vendor and device type. Mirai primarily targeted IP cameras, DVRs, and consumer routers.

we analyzed the credentials hardcoded into the binaries we collected. We observed a total 371 unique passwords, and through manual inspection, we identified 84 devices and/or vendors associated with these passwords. Many passwords were too generic to tie to a specific device (i.e., "password" applies to devices from a large number of manufacturers), while others only provided information about underlying software (e.g., "postgres") and not an associated device. The devices we identified were primarily network-attached storage appliances, home routers, cameras, DVRs, printers, and TV receivers made by dozens of different manufacturers (Table 5).

Mirai's intended targets do not necessarily reflect the breakdown of infected devices in the wild. We leveraged the device banners collected by Censys to determine the models and manufacturers of infected devices. Our results across all five protocols indicate that security cameras, DVRs, and consumer routers represent the majority of Mirai infections (Table 6). The manufacturers responsible for the most infected devices we could identify are: Dahua, Huawei, ZTE, Cisco, ZyXEL, and MikroTik (Table 7).

We note that these results deviate from initial media reports, which stated that Mirai was predominantly composed of DVRs and cameras [34,53,60]. This is likely due to the evolution of the Mirai malware over time, which changed the composition of infected devices. Looking at the longitudinal Pearson correlation of top device vendors, we observe modest stability with the exception of two event periods: the rapid growth phase in mid-September 2016 and the onset of CWMP in late November 2016 (Figure 5b). During the rapid growth, the emergence of consumer routers manufactured by ASUS, Netgear, and Zhone supplanted D-Link routers and Controlbr DVRs in the top 20 devices. Dahua, Huawei, ZyXEL, and ZTE devices consistently remained in the Top 20.

Our data indicates that some of the world's top manufacturers of consumer electronics lacked sufficient security practices to mitigate threats like Mirai, and these manufacturers will play a key part in ameliorating vulnerability. Unfortunately, as discussed in the previous section, the menagerie of devices spanned both countries

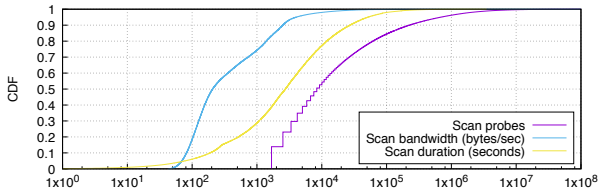


Figure 6: **Network Capacity Distribution**—Scan duration, probes, and bandwidth were extrapolated to reflect scanning network capacity across the full IPv4 Internet. A majority of probes scan below 250 Bps for over 2,700 seconds.

and legal jurisdictions, exacerbating the challenge of coordinating technical fixes and promulgating new policy to safeguard consumers in the future.

4.5 Device Bandwidth

As an additional confirmation of embedded composition, we examined the bandwidth of infected devices as gleaned from their scan rate, which is not artificially rate-limited by the original source code. Starting with the observed scanning rate and volume on our network telescope, we extrapolate across the entire IPv4 Internet by factoring in the size of our network telescope (4.7 million IPs) and the size of Mirai's default IP blacklist (340.2 million IPs). We found about half of the Mirai bots that scanned our network telescope sent fewer than 10,000 scan packets (Figure 6). We further note that the majority of bots scanned at an estimated rate below 250 bytes per second. We note however this is a strict underestimate, as Mirai may have interrupted scanning to process C2 commands and to conduct brute force login attempts. In contrast, SQL Slammer scanned at 1.5 megabytes/second, about 6000 times faster [68], and the Witty worm scanned even faster at 3 megabytes/second [81]. This additionally hints that Mirai was primarily powered by devices with limited computational capacity and/or located in regions with low bandwidth [3].

5 Ownership and Evolution

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

5.1 Ownership

In order to identify the structure of Mirai command and control servers, we turned to active and passive DNS data, which we used to cluster C2 IPs and domains based on

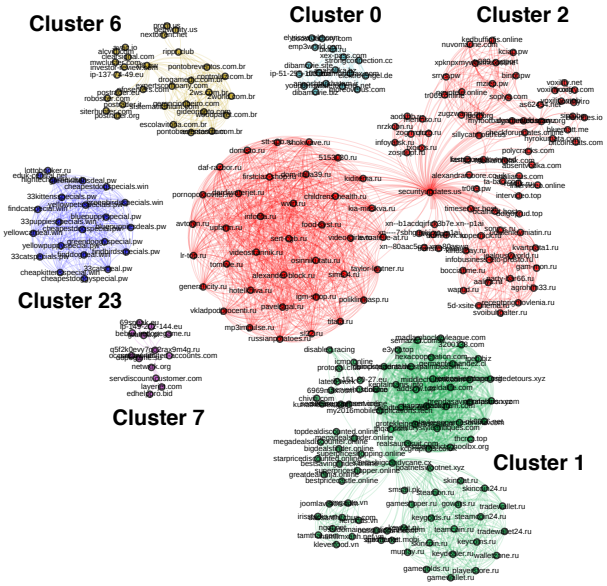


Figure 7: **C2 Domain Relationships**—We visualize related C2 infrastructure, depicting C2 domains as nodes and shared IPs as edges between two domains. The top six clusters by C2 domain count consisted of highly connected components, which represent agile, long-lived infrastructures in use by botmasters.

shared network infrastructure. Seeding DNS expansion with the two IPs and 67 domains that we collected by reverse engineering Mirai binaries, we identified 33 independent C2 clusters that shared no infrastructure. These varied from a single host to the largest cluster, which contained 112 C2 domains and 92 IP addresses. We show the connectivity of the top six clusters by number of C2 domains in Figure 7. The lack of shared infrastructure between these clusters lends credence to the idea that there are multiple active bot operators during our study period.

While Figure 7 provides a rough sense of Mirai C2 complexity, it does not indicate the number of bots that each cluster controlled. To estimate botnet membership, we measured the DNS lookup volume per cluster. In Figure 8, we show the top clusters of domains based on the volume of DNS lookups at a large, name-redacted ISP. This single perspective is not comprehensive, but it allows us to observe the rise and fall of different botnets over time, and may provide a hint of their relative sizes. A prime example is cluster 1, which was the initial version of the Mirai botnet involved in the early, high-profile attacks on Krebs on Security and OVH. Although it dominated in lookup volume in late September and early October, it gave way to newer clusters, 2 and 6, in mid-October. We provide a list of the largest clusters by lookup and their unique characteristics in Table 8.

While we cannot conclusively link each of these clusters to distinct operators, we note that each cluster utilized independent DNS infrastructure and evolving malware,

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

Table 6: **Top Mirai Device Types**—We list the top types of infected devices labeled by active scanning, as a fraction of Mirai banners found in Censys. Our data suggests that consumer routers, cameras, and DVRs were the most prevalent identifiable devices.

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

Table 7: **Top Mirai Device Vendors**—We list the top vendors of infected Mirai devices labeled by active scanning, as a fraction of Mirai banners found by Censys. The top vendors across all protocols were primarily camera, router, and embedded device manufacturers.

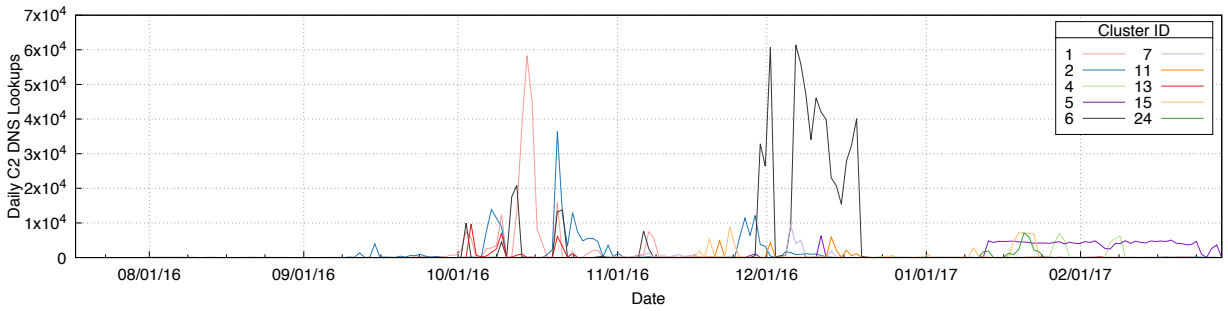


Figure 8: **C2 Cluster Lookup Volume**—The DNS lookup volume of C2 DNS clusters in a large U.S. ISP establishes the relative size of the botnet behind each cluster and chronicles its rise and fall. Note, for example, cluster 1 which represents the original botnet in use for the early high profile attacks on Krebs and OVH and the emergence of a myriad of clusters after the public source release.

underscoring the challenge of defending against these attacks through bespoke mitigations. Our results also confirm the recent findings of Lever et al., who observed that the naming infrastructure used by malware is often active weeks prior to its operation [54]. In all cases, the first occurrence of DNS/IP lookup traffic for a cluster far preceded the date that the domains were used as C2 infrastructure for the botnet. For example, even though the peak lookup for cluster 2 occurred on October 21, 2016, the first lookup of a C2 domain in this cluster occurred on August 1, 2016 (Table 8). This also significantly predated the first binary collected for this cluster (October 24, 2016), and the first attacks issued by the cluster (October 26, 2016). These results suggest that careful analysis of DNS infrastructure can potentially guide preventative measures.

5.2 Evolution

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

After the public release, we observed the rapid emergence of new features, ranging from improved infection capabilities to hardened binaries that slow reverse engineering efforts. Between November 2, 2016 and February 28, 2017, we observed 48 new sets of usernames and passwords, as well as changes to the IP blacklist. We note that while many actors modified the set of credentials, they often did so in different ways (Figure 9). This is true for other features as well. In one example, a variant evolved to remove U.S. Department of Defense blocks from the initial scanning blacklist. The malware further evolved to use new infection mechanisms. Most notably, in late November 2016, Mirai variants began to scan for TCP/7547 and TCP/5555, two ports commonly associated with CWMP [15, 93]. Additionally, one malware strain began to using domain generation algorithms (DGA) in the place of a hardcoded C2 domain, though this feature was short lived. By November 2016, packed binaries had

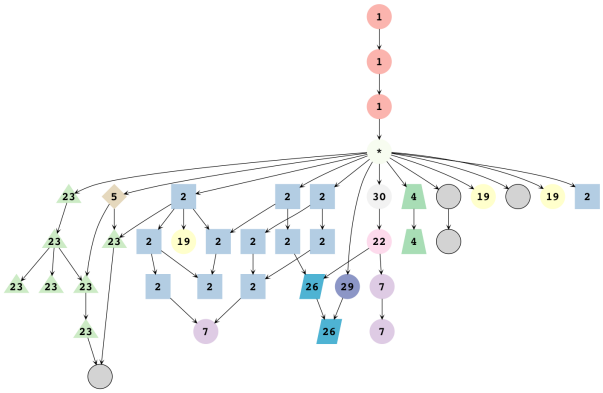


Figure 9: **Password Evolution**—The lineage of unique password dictionaries, labeled with their associated clusters, depicts many malware strains modifying the default credential list to target additional devices. The node marked (*) indicates the released source code password dictionary and serves as the foundation for the all divergent password variants

emerged.

Techniques to improve virulence and to aide in reliability were not simply limited to the client binaries. We found evidence of operators using DNS to avoid or attempt to evade detection as well. Recent work by Lever et al. demonstrated how attackers abuse the residual trust inherited by domains to perform many, seemingly unconnected types of abuse [55]. Mirai was no different from other types of malware—we found evidence that at least 17% of Mirai domains abused residual trust. Specifically, these domains expired and were subsequently re-registered before they were used to facilitate connections between bots and C2 servers. This serves as a reminder that although Mirai is unique in many ways, it still shares much in common with the many threats that came before it.

By combining the malware we observed with our DNS data, we can also measure the evolution of the C2 clusters in Table 8. We note that cluster 2—the third largest by lookup volume—evolved to support many new features, such as scanning new ports TCP/7547 and TCP/5555, adding DGA, and modifying the source code blacklist to exclude Department of Defense (DoD) blocks. This is not to say, however, that evolution guaranteed success. Cluster 23, which can be seen clearly in Figure 9, evolved very rapidly, adding several new passwords over its active time. Despite this evolution, this cluster was 19th out of 33 clusters in terms of lookup volume over time and was unable to capture much of the vulnerable population. We also note that not all successful clusters evolved either; for example, cluster 6, which showed no evolutionary trend from its binaries, received the highest lookup volume of all the clusters.

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

Table 9: **C2 Attack Commands**—Mirai launched 15,194 attacks between September 27, 2016–February 28, 2017. These include [A]pplication-layer attacks, [V]olumetric attacks, and TCP [S]tate exhaustion, all of which are equally prevalent.

6 Mirai's DDoS Attacks

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

6.1 Types of Attacks

Over the course of our five month botnet infiltration, we observed Mirai operators issuing 15,194 DDoS attack commands, excluding duplicate attacks (discussed in Section 3). These attacks employed a range of different resource exhaustion strategies: 32.8% were volumetric, 39.8% were TCP state exhaustion, and 34.5% were application-layer attacks (Table 9). This breakdown differs substantially from the current landscape of DDoS attacks observed by Arbor Networks [7], where 65% of attacks are volumetric, 18% attempt TCP state exhaustion, and 18% are higher-level application attacks. While amplification attacks [79] make up 74% of attacks issued by DDoS-for-hire booter services [40], only 2.8% of Mirai attack commands relied on bandwidth amplification, despite built-in support in Mirai's source code. This absence highlights Mirai's substantial capabilities despite the resource constraints of the devices involved.

6.2 Attack Targets

Studying the victims targeted by Mirai sheds light on its operators. We analyzed the attack commands issued by Mirai C2 servers (as detailed in Section 3) to examine who Mirai targeted. In total, we observed 15,194 attacks issued by 484 C2 IPs that overlapped with 24 DNS clusters (Section 5). The attacks targeted 5,046 victims, comprised of 4,730 (93.7%) individual IPs, 196 (3.9%) subnets, and 120 (2.4%) domain names. These victims ranged from game servers, telecoms, and anti-DDoS providers, to political websites and relatively obscure Russian sites (Table 10).

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

If we exclude targeted subnet (due to their unfocused blanket dispersion across many networks), we find that Mirai victims were distributed across 906 ASes and 85 countries. The targets were heavily concentrated in the U.S. (50.3%), France (6.6%), the U.K. (6.1%), and a long tail of other countries. Network distribution was more evenly spread. The top 3 ASes—OVH (7.8%), Cloudflare (6.6%) and Comcast (3.6%)—only accounted for 18.0% of victims.

The three most frequently targeted victims were Liberia's Lonestar Cell (4.1%), Sky Network (2.1%), and `1.1.1.1` (1.6%). We examine Lonestar Cell in depth in Section 6.3. Sky Network is a Brazilian company that operates servers for Minecraft (a popular game), which is hosted by Psychz Networks. The attacks against Psychz began on November 15, 2016 and occurred sporadically until January 26, 2017. `1.1.1.1` was likely used for testing [95]. Additional game targets in the top 14 victims included a former game commerce site longqikeji.com, and Runescape, another popular online game. The prevalence of game-related targets along with the broad range of other otherwise unrelated victims shares many characteristics with previously studied DDoS booter services [39].

For volumetric and TCP state exhaustion attacks, Mirai optionally specified a target port, which implied the type of service targeted. We find a similar prevalence of game targets—of the 5,450 attacks with a specified port, the most commonly attacked were 80 (HTTP, 37.5%), 53 (DNS, 11.5%), 25565 (commonly Minecraft

Attack Target	Date	Sample Size	Intersection
Akamai†	09/21/2016	12,847	96.4%
Project Shield†	09/25/2016	158,839	96.4%
Dyn◇	10/21/2016	107,464	70.8%

Table 11: **Mirai Attack IPs** — Client IPs from attacks on Krebs on Security (denoted †) and Dyn (denoted ◇) intersected significantly with Mirai-fingerprinted scanning our network telescope, confirming that both attacks were Mirai-based, but the lower Dyn intersection hints that other hosts may have been involved.

servers [31,65], 9.2%), 443 (HTTPS, 6.4%), 20000 (often DNP3, 3.4%), and 23594 (Runescape game server, 3.4%).

Interestingly, the 7th most common attack target was an IP address hosted by Voxility that was associated with one of the Mirai C2 servers, and we note that 47 of 484 Mirai C2 IPs were themselves the target of a Mirai DDoS attack. By clustering these 484 C2 IPs by attack command, we identified 93 unique clusters, of which 26 (28%) were targeted least once. This direct adversarial behavior reaffirms the notion of multiple, competitive botnet operators.

6.3 High Profile Attacks

Several high profile DDoS attacks brought Mirai into the limelight beginning in September 2016. We analyze the following three Mirai victims as case studies: Krebs on Security, Dyn, and the Liberian telecom provider Lonestar.

Krebs on Security The popular Krebs on Security blog has had a long history of being targeted by DDoS attacks (Figure 10), and on September 21, 2016 was subject to an unprecedented 623 Gbps DDoS attack — with Mirai as the prime suspect. Placing this attack in context, it was significantly larger than the previously reported largest publicly-disclosed DDoS attack victim (i.e., Spamhaus at 300+ Gbps [77]), but we note that attacks to non-disclosed targets of 500 Gbps and 800 Gbps were reported in 2015 and 2016 respectively [7]. To confirm the origin of the attack, we intersected a list of 12,847 attack IPs observed by Akamai with the Mirai IPs we saw actively scanning during that period. We found a 96.4% overlap in hosts. Google's Project Shield, who later took over DDoS protection of the site, separately maintained a larger sample of 158,839 attack IPs for an HTTP attack on September 25, 2016. When given the Mirai scanning IPs from that day, they found 96% of their attack IPs overlapped. Our results illustrate the potency of the Mirai botnet, despite its composition of low-end devices concentrated in Southeast Asia and South America. We also identified which C2 clusters were responsible for some of the largest attacks by correlating attack commands with naming infrastructure, and we note that cluster 1 (Figure 7) was

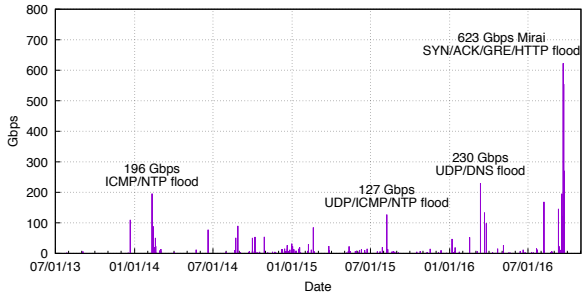


Figure 10: **Historical DDoS Attacks Targeting Krebs on Security** — Brian Krebs' blog was the victim of 269 DDoS attacks from 7/24/2012–9/22/2016. The 623 Gbps Mirai attack on 9/21/2016 was 35 times larger than the average attack, and the largest ever recorded for the site.

responsible for this attack.

Dyn On October 21, 2016, Dyn, a popular DNS provider suffered a series of DDoS attacks that disrupted name resolution for their clients, including high-traffic sites such as Amazon, Github, Netflix, PayPal, Reddit, and Twitter [71]. Consistent with Dyn's postmortem report [36], we observed 23 attack commands that targeted Dyn infrastructure, from 11:07–16:55 UTC. The first 21 attacks were primarily short-lived (i.e., 25 second) SYN floods on DNS port 53, along with a few ACK and GRE IP attacks, and followed by sustained 1 hour and 5 hour SYN attacks on TCP/53. We note a 71% intersection between the 107K IPs that attacked Dyn and Mirai scanning in our network telescope. This indicates that, while the attack clearly involved Mirai, there may have been other hosts involved as well.

Although the first several attacks in this period solely targeted Dyn's DNS infrastructure, later attack commands simultaneously targeted Dyn and PlayStation infrastructure, potentially providing clues towards attacker motivation. Interestingly, the targeted Dyn and PlayStation IPs are all linked to PlayStation name servers — the domain names `ns<00-03>.playstation.net` resolve to IPs with reverse DNS records pointing to `ns<1-4>.p05.dynect.net`, and the domain names `ns<05-06>.playstation.net` resolve to the targeted PlayStation infrastructure IPs.

The attacks on Dyn were interspersed amongst other attacks targeting Xbox Live, Microsoft DNS infrastructure, PlayStation, Nuclear Fallout game hosting servers, and other cloud servers. These non-Dyn attacks are either ACK/GRE IP floods, or VSE, which suggests that the targets were Valve Steam servers. At 22:17 UTC, the botnet issued a final 10 hour-long attack on a set of Dyn and PlayStation infrastructure. This pattern of behavior suggests that the Dyn attack on October 21, 2016 was not solely aimed at Dyn. The attacker was likely targeting

gaming infrastructure that incidentally disrupted service to Dyn's broader customer base. The attack was carried out by Cluster 6.

Lonestar Cell Attacks on Lonestar Cell, a large telecom operator in Liberia and the most targeted victim of Mirai (by attack account), have received significant attention due to speculation that Mirai substantially deteriorated Liberia's overall Internet connectivity [14, 42]. Others have questioned these claims [45]. We cannot provide insight into Liberia's network availability; instead, we analyze attack commands we observed. Beginning at 10:45 UTC on October 31, 2016 until December 13, 2016, a single botnet C2 cluster (id 2) issues a series of 341 attacks against hosts in the Lonestar AS. 87% of the attacks are SYN or ACK floods and targeted both full subnets and addresses within 168.253.25.0/24, 41.57.81.0/24, and 41.57.85.0/24, all of which belong to Lonestar Cell or its parent company, MTN Group.

In addition to IP targets, we observe an NXDOMAIN attack issued on November 8, 2016 that targeted `simregistration.lonestarcell.com`. A single C2 IP never seen previously or subsequently issued a single attack on December 14. Attacks on Lonestar infrastructure continued again at 09:24 UTC on January 16, 2017 and persisted until February 8, 2017, issuing 273 attacks from a single C2 IP address. In total there were 616 attacks, 102 of which used reflect traffic against Voxility, Google, Facebook, and Amazon servers towards Lonestar networks. The attack was carried out by C2 cluster 2 and used the C2 domains: "mufoscam.org", "securityupdates.us", "jgop.org", and "zugzwang.me".

As we have seen, Mirai primarily used direct, non-reflective attacks on a wide range of protocols including the less common GRE and VSE protocols. Even without relying on amplification attacks, Mirai was still able to inflict serious damage as evidenced by high-profile attacks against Krebs on Security, Dyn, and Lonestar Cell. Furthermore, the juxtaposition of attacker geography (largely Southeast Asia and South America) and victim geography (majority in the U.S.) places a spotlight on the importance of global solutions, both technical and non-technical, to prevent the rise of similar botnets. Otherwise, adversaries will continue to abuse the most fragile hosts to disrupt the overall Internet ecosystem.

7 Discussion

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

Security hardening The Mirai botnet demonstrated that even an unsophisticated dictionary attack could compromise hundreds of thousands of Internet-connected devices. While randomized default passwords would be a first step, it is likely that attacks of the future will evolve to target software vulnerabilities in IoT devices much like the early Code Red and Confickr worms [8, 70]. To mitigate this threat before it starts, IoT security must evolve away from default-open ports to default-closed and adopt security hardening best practices. Devices should consider default networking configurations that limit remote address access to those devices to local networks or specific providers. Apart from network security, IoT developers need to apply ASLR, isolation boundaries, and principles of least privilege into their designs. From a compliance perspective, certifications might help guide consumers to more secure choices as well as pressure manufacturers to produce more secure products.

Automatic updates Automatic updates—already canonical in the desktop and mobile operating system space—provide developers a timely mechanism to patch bugs and vulnerabilities without burdening consumers with maintenance tasks or requiring a recall. Automatic updates require a modular software architecture by design to securely overwrite core modules with rollback capabilities in the event of a failure. They also require cryptographic primitives for resource-constrained devices and building PKI infrastructure to support trusted updates. Apart from these challenges, patching also requires the IoT community to actively police itself for vulnerabilities, a potentially burdensome responsibility given the sheer diversity of devices. Bug bounties can help in this respect: roughly 25% of all vulnerabilities patched by Chrome and Firefox came from bug bounties in 2015 [28], while Netgear launched a bug bounty for its router software in January, 2017 [75]. In the event of a zero-day exploit that disables automatic updates, IoT developers must provide a secure fallback mechanism that likely requires physical access and consumer intervention.

The Deutsche Telekom infection and subsequent fix provide an excellent case study of this point. DT's routers had a vulnerability that enabled the botnet to spread via its update mechanism, which provides a reminder that basic security hardening should be the first priority. However, since DT did have an automatic update mechanism, it was also able to patch devices rather swiftly, requiring mini-

mal user intervention. Implementing automatic updates on IoT devices is not impossible, but does take care to do correctly.

Notifications Notifications via out-of-band channels serve as a fallback mechanism to bring devices back into security compliance or to clear infections. Recent examples include alerting device administrators via CERT bulletins, emailing the abuse contact in WHOIS records, and in-browser warnings to site owners that a page is compromised [24, 56, 57]. Notifications in the IoT space are complicated to say the least. IoT devices lack both a public indication of ownership and an established communication channel to reach consumers. Were consumers reachable, there must also be a clear and simple update path to address the problem. As a minimum alternative, IoT devices could be required to register an email address with the manufacturer or with a unified, interoperable monitoring platform that can alert consumers of serious issues. This is a space where IoT requires non-technical intervention. The usability challenge of acting on notifications remains an open research problem.

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

Defragmentation Fragmentation poses a security (and interoperability) risk to maintaining and managing IoT devices. We observed numerous implementations of Telnet, FTP, and HTTP stacks during scanning. The IoT community has responded to this challenge by adopting a handful of operating systems, examples of which include Android Thing, RIOT OS, Tock, and Windows for IoT [30]. This push towards defragmentation would abstract away the security nuances required of our prescriptive solutions.

End-of-life Even with security best practices in mind, end-of-life can leave hundreds of thousands of in-use IoT devices without support. Lack of long-term support will yield a two class system of protected and unprotected devices similar to the current state of Windows XP machines [63]. Over time, the risk that these devices pose to the Internet commons will only grow unless taken offline.

8 Related Work

Since as early as 2005, the security community has been working to understand, mitigate, and disrupt botnets [17]. For example, Zand et al. proposed a detection method based on identifying command and control signatures [97], and Gu et al. focused on analyzing network traffic to aid in detection and mitigation [32, 33]. Unfortunately, mitigation remains a difficult problem as botnets often evolve to avoid disruption [6].

This work follows in a long line studies that have analyzed the structure, behavior, and evolution of the botnet ecosystem [12, 37, 76, 84, 85, 91, 96]. Bailey et al. note that each technique used in understanding botnets has a unique set of trade offs, and only by combining perspectives can we fully analyze the entire picture [11]. This observation and the seminal work of Rajab et al., implicating botnet activity in 27% of all network telescope traffic, inspire our approach [2].

Botnets have historically been used to launch DDoS attacks, and there exists a parallel set of studies focusing on characterizing and defending against these attacks [66, 67], as well as estimating their effect [69]. In response to the recent growth of amplification attacks, there have been several studies investigating vulnerable amplifiers [20, 51, 79]. As DDoS attacks and infrastructure are becoming more commonplace, attention has turned to exploring the DDoS for hire ecosystem [40].

Since the emergence of IoT devices, security researchers have warned of their many inherent security flaws [80]. Researchers have found that IoT devices contain vulnerabilities from the firmware level [18, 19] up to the application level [26, 29, 73, 78]. Mirai is also not the first of its kind to target IoT devices — several precursors to Mirai exist, all of which exploit the weak password nature of these devices [38, 52, 59, 62, 72]. As a result of these widespread security failures, the security community has been quick to design systems to secure these kinds of devices. In one example, Fernandes et al. proposed Flowfence, which enables data flow protection for emerging IoT frameworks [27]. Much more work is needed if we are to understand and secure this new frontier.

In this work, we utilize a multitude of well-established botnet measurement perspectives, which substantiate concerns about IoT security. We demonstrate the damage that an IoT botnet can inflict upon the public Internet, eclipsing the DDoS capabilities of prior botnets. We use previously introduced solutions as guidelines for our own proposals for combating the Mirai botnet, and IoT botnets at large.

9 Conclusion

The Mirai botnet, composed primarily of embedded and IoT devices, took the Internet by storm in late 2016 when it overwhelmed several high-profile targets with some of the largest distributed denial-of-service (DDoS) attacks on record. In this work, we provided a comprehensive analysis of Mirai's emergence and evolution, the devices it targeted and infected, and the attacks it executed. We find that while IoT devices present many unique security challenges, Mirai's emergence was primarily based on the absence of security best practices in the IoT space, which resulted in a fragile environment ripe for abuse. As the IoT domain continues to expand and evolve, we hope Mirai serves as a call to arms for industrial, academic, and government stakeholders concerned about the security, privacy, and safety of an IoT-enabled world.

Acknowledgments

The authors thank David Adrian, Brian Krebs, Vern Paxson, and the Censys Team for their help and feedback. This work was supported in part by the National Science Foundation under contracts CNS-1345254, CNS-1409505, CNS-1518888, CNS-1505790, CNS-1530915, CNS-1518741 and through gifts from Intel and Google. The work was additionally supported by the U.S. Department of Commerce grant 2106DEK, Air Force Research Laboratory/Defense Advanced Research Projects Agency grant 2106DTX, the Department of Homeland Security Science and Technology Directorate FA8750-12-2-0314, and a Google Ph.D. Fellowship. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of their employers or the sponsors.

References

[1] Team Cymru. http://www.team-cymru.org/.

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

[3] Akamai. Q4 2016 state of the Internet - connectivity report. https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/q4-2016-state-of-the-internet-connectivity-report.pdf.

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

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

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

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

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

[9] M. Bailey, E. Cooke, F. Jahanian, and J. Nazario. The Internet Motion Sensor - A Distributed Blackhole Monitoring System. In *12th Network and Distributed Systems Security Symposium*, 2005.

[10] M. Bailey, E. Cooke, F. Jahanian, and D. Watson. The Blaster worm: Then and now. *IEEE Security & Privacy*, 2005.

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

[12] P. Barford and V. Yegneswaran. *An Inside Look at Botnets*. 2007.

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

[14] K. Beaumont. "Shadows Kill"−−Mirai DDoS botnet testing large scale attacks, sending threatening messages about UK and attacking researchers. https://medium.com/@networksecurity/shadows-kill-mirai-ddos-botnet-testing-large-scale-attacks-sending-threatening-messages-about-6a61553d1c7.

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

[16] CAIDA: Center for Applied Internet Data Analysis. AS ranking. http://as-rank.caida.org/?mode0=as-ranking&n=100&ranksort=3, 2017.

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

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

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

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

[21] Deutsche Telekom. Telekom-hilt. https://www.facebook.com/telekomhilft/photos/a.143615195685585.27512.122768271103611/1199966633383764/?type=&theater.

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

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

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

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

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

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

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

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

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

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

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

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

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

[35] K. J. Higgins. Srizbi botnet sending over 60 billion spams a day. http://www.darkreading.com/risk/srizbi-botnet-sending-over-60-billion-spams-a-day/d/d-id/1129480.

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

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

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

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

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

[41] kenzo2017. Eir's d1000 modem is wide open to being hacked. https://devicereversing.wordpress.com/2016/11/07/eirs-d1000-modem-is-wide-open-to-being-hacked/.

[42] S. Khandelwal. Someone is using mirai botnet to shut down internet for an entire country. http://thehackernews.com/2016/11/ddos-attack-mirai-botnet.html.

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

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

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

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

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

[48] B. Krebs. Spreading the DDoS disease and selling the cure. https://krebsonsecurity.com/2016/10/spreading-the-ddos-disease-and-selling-the-cure/.

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

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

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

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

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

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

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

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

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

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

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

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

[61] Maxmind, LLC. Geoip2. https://www.maxmind.com/en/geoip2-city.

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

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

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

[65] Minecraft Modern Wiki. Protocol handshaking. http://wiki.vg/Protocol#Handshaking.

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

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

[68] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer worm. *IEEE Security & Privacy*, 2003.

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

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

[71] S. Moss. Major DDoS attack on Dyn disrupts AWS, Twitter, Spotify and more. http://www.datacenterdynamics.com/content-tracks/security-risk/major-ddos-attack-on-dyn-disrupts-aws-twitter-spotify-and-more/97176.fullarticle.

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

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

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

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

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

[77] M. Prince. The DDoS that almost broke the internet. https://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet/.

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

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

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

[81] C. Shannon and D. Moore. The spread of the Witty worm. *IEEE Security & Privacy*, 2004.

[82] S. Shin and G. Gu. Conficker and Beyond: A Large-scale Empirical Study. In *26th Annual Computer Security Applications Conference*, 2010.

[83] S. S. C. Silva, R. M. P. Silva, R. C. G. Pinto, and R. M. Salles. Botnets: A survey. 2013.

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

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

[86] A. Tellez. Bashlite. https://github.com/anthonygtellez/BASHLITE.

[87] K. Thomas, R. Amira, A. Ben-Yoash, O. Folger, A. Hardon, A. Berger, E. Bursztein, and M. Bailey. The abuse sharing economy: Understanding the limits of threat exchanges. In *19th Symposium on Research in Attacks, Intrusions and Defenses*, 2016.

[88] K. Thomas, D. Y. Huang, D. Wang, E. Bursztein, C. Grier, T. J. Holt, C. Kruegel, D. McCoy, S. Savage, and G. Vigna. Framing dependencies introduced by underground commoditization. In *14th Workshop on the Economics of Information Security*, 2015.

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

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

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

[92] N. Wells. Busybox: A swiss army knife for linux.

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

[94] E. Wustrow, M. Karir, M. Bailey, F. Jahanian, and G. Houston. Internet Background Radiation Revisited. In *10th ACM Internet Measurement Conference*, 2010.

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

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

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

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

Table 8: **Cluster Size Estimate and Characteristics** — We highlight the top five clusters by max single-day lookup volume within a large U.S. ISP, which provides an indicator of their relative size. Each cluster is additionally labeled with observed evolutionary patterns and associated attacks.

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

Table 10: **Mirai DDoS Targets** — The top 14 victims most frequently targeted by Mirai run a variety of services. Online games, a Liberian cell provider, DDoS protection services, political sites, and other arbitrary sites match the victim heterogeneity of booter services. Many clusters targeted the same victims, suggesting a common operator.