The Evolution of Bashlite and Mirai IoT Botnets

Artur Marzano*, David Alexander*, Osvaldo Fonseca*, Elverton Fazzion*†, Cristine Hoepers‡,
Klaus Steding-Jessen‡, Marcelo H. P. C. Chaves‡, Ítalo Cunha*, Dorgival Guedes*, Wagner Meira Jr.*
*Department of Computer Science – Universidade Federal de Minas Gerais (UFMG)
†Department of Computing – Universidade Federal de São João del-Rei (UFSJ)
‡CERT.br - Brazilian National Computer Emergency Response Team
NIC.br - Brazilian Network Information Center

***Abstract*—Vulnerable IoT devices are powerful platforms for building botnets that cause billion-dollar losses every year. In this work, we study Bashlite botnets and their successors, Mirai botnets. In particular, we focus on the evolution of the malware as well as changes in botnet operator behavior. We use monitoring logs from 47 honeypots collected over 11 months. Our results shed new light on those botnets, and complement previous findings by providing evidence that malware, botnet operators, and malicious activity are becoming more sophisticated. Compared to its predecessor, we find Mirai uses more resilient hosting and control infrastructures, and supports more effective attacks.**

I. Introduction

Distributed Denial of Service (DDoS) attacks attempt to exhaust resources such as CPU, memory, and bandwidth of devices in the Internet to degrade services, and may target servers or network equipment. For example, DDoS attacks may flood web servers with spurious requests, causing legitimate requests to be delayed or dropped [1]. It has been reported that DDoS attacks lead to financial losses on the order of 2 billion dollars per year [2]. Attacks performed from distributed infrastructures are more effective, as they can dedicate more resources to overload targets and require more advanced mitigation mechanisms [3].

DDoS attacks are frequently launched from *botnets*, a set of of network devices infected with *malware*, known as *bots* or *zombies*. Botnets include *command and control servers* (C&C), which maintain connections with active bots and allow the botnet's operator to broadcast commands to them. Bots can perform a wide range of tasks, including scan other devices for vulnerabilities, infect vulnerable devices, send spam e-mail messages, or perform different types of attacks.

The growth of the Internet of Things (IoT), combined with widespread vulnerabilities found in its devices, has attracted the attention of malicious agents interested in subverting those devices. Today, IoT devices are a powerful platform for creating large-scale botnets with significant computational power and network bandwidth. Flooding DDoS attacks perpetrated by botnets based on IoT devices have exceeded 1.2 Tbps [4]; these attacks have successfully disrupted basic Internet services like DNS, impacting millions of users, and have been used to extort money from attacked networks. Researchers and security experts have dedicated efforts to characterize those botnets and develop countermeasures [3], [5], [6], [7], [8].

Our study complements previous research and sheds new light into the evolution of botnet malware and the behavior of botnet operators. In this paper we characterize two families of IoT botnets: Bashlite (introduced in 2015 [3]) and its successor, Mirai (introduced in 2016 [6]) (Section II). In particular, we focus on the evolution of the set of supported attacks and which attacks are perpetrated (operator behavior).

Our study analyzes 11 months of malicious activity monitored by 47 low-interactivity *honeypots* deployed around Brazil (Section III). The honeypots emulate vulnerable devices and allow us to monitor infection attempts by botnet malware. We also collect data from monitors that connect to C&Cs to receive broadcast commands. Our dataset captured scan and infection attempts from millions of infected devices and monitored hundreds of C&Cs.

We focus on the evolution of IoT botnets, comparing Bashlite with Mirai and discussing their differences. Overall, we find that Mirai is more sophisticated than Bashlite. We make the following contributions:

- We characterized Bashlite's and Mirai's support infrastructures (Section IV) and found that although C&Cs and malware servers are concentrated in few ASes, most of them infrastructure providers, this concentration is decreasing and Mirai botnets' infrastructures are hosted in networks more diverse than Bashlite's.
- We characterized available attacks and how they are used (Section V-A). Compared with Bashlite, Mirai supports more sophisticated, application-layer attacks and Mirai operators effectively use such application-layer attacks.
- We characterized attack targets (Section V-B) and, although the set of most attacked ASes changed, attacks target similar services (*e.g.*, content providers and online game servers).
- We characterized how operators manage and use their botnets (Sections V-C and V-D). We found that Mirai operators coordinate attacks across multiple botnets more often than Bashlite operators, and that Bashlite operators have to deal with significantly higher management complexity.

Our results broaden our understanding of the Bashlite and Mirai botnets, specially how they evolved. Understanding the evolution of botnets will aid researchers and operators maintain up-to-date and effective countermeasures, ultimately benefiting end users and businesses.

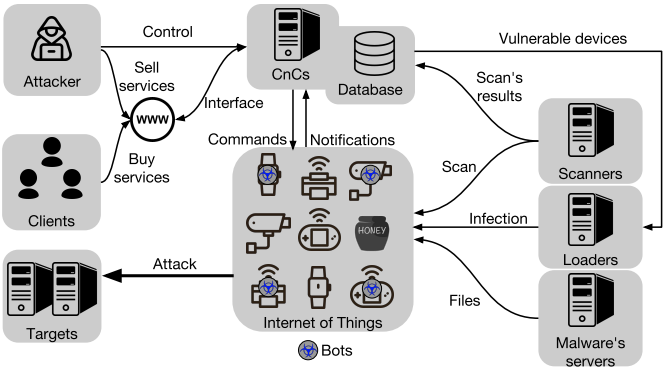

Figure 1. Overview of an IoT botnet.

II. The Bashlite and Mirai IoT Botnets

In this section we summarize the characteristics of the Bashlite and Mirai botnets. Mirai's source code is based on Bashlite's, and the botnets share many similarities. In particular, their main feature is that they infect IoT devices (*e.g.*, broadband modems and surveillance cameras) accessible with vulnerable, known authentication credentials. Figure 1 shows an overview of the ecosystem in which those botnets operate.

- **Command and control servers (C&C)** are the operators' interface to the botnet. C&Cs receive commands from operators and maintain connections with infected devices to broadcast commands.
- **Bots** are infected devices that are part of the botnet. Bots report their state to C&Cs and execute the received commands.
- **Scanners** probe devices to find telnet and SSH servers to attempt login and identify vulnerable devices.
- **Loaders** login to vulnerable devices to download and run the botnet malware, creating a new bot.
- **Malware servers** host resources used by the botnet such as shell scripts and executable binaries.
- **Database** (potentially distributed) stores information collected by the botnet, *e.g.*, active bots and scan results.

The infection of devices is performed in two steps. Scanners first identify vulnerable devices and report to a central database. Loaders then connect to the vulnerable devices to download and run the malware. During the infection process, loaders access servers to download and run malware binaries on the vulnerable device. Once infected, a bot connects to the botnet's C&C and awaits commands. To prevent subsequent infection attempts from other botnets, Bashlite and Mirai disable the infected device's telnet and SSH services.

Mirai improves on Bashlite on multiple fronts. While some variants of the Bashlite malware have extensions to allow bots to scan for vulnerable devices, Mirai has this functionality built-in. While Bashlite malware specifies C&Cs as hardcoded IP addresses, Mirai resolves C&Cs IP addresses using DNS; indirection through DNS makes it harder to take down C&Cs. While Bashlite commands are specified in plain text and transmitted unencrypted,[1] Mirai's communication uses a compact binary protocol.

Finally, the operator may sell botnet services (*e.g.*, denial of service attacks), usually through a Web interface that clients can access [9].

III. IoT Botnet Dataset

The data analyzed in this work were collected by two monitoring infrastructures, allowing us to monitor the scanning and infection of vulnerable devices as well as the behavior of botnet operators. The data were collected from January 1st to November 13th, 2017. As our monitoring platform does not resolve hostnames in real time, we attempted to resolve all hostnames in the dataset (C&Cs, malware servers, and attack targets) on December 18th, 2017.

Scanning and Infection Activity. We monitor scanning and infection activity using 47 honeypots distributed across Brazil. The honeypots emulate SSH and telnet servers accessible through known vulnerable credentials used by IoT device vendors and exploited by Bashlite and Mirai. The honeypots capture both stages in the infection process: (i) host scans followed by authentication attempts using dictionary attacks, and (ii) logins using vulnerable credentials followed by a sequence of commands to infect the device.

The honeypots never execute commands. Instead, honeypots interpret the commands and return the expected responses to mimic vulnerable devices and elicit additional commands. This is possible because the infection process is automated and uses a pre-defined sequence of commands.

All received commands are logged and sent to a server which post-processes them to identify attempts to download malware (*e.g.*, using wget, curl, or scp) and extract URLs. The URLs are then downloaded and their payloads stored in an isolated system.

During the collection period, our honeypots interpreted 342,001,071 commands received from 2,385,460 IP addresses, associated with 12,842 autonomous systems.

Botnet Operator Behavior. We observe the behavior of botnet operators using monitors that emulate an infected device. As with honeypots, monitors never execute commands. Instead, monitors send forged notifications and preprogrammed responses to emulate a bot, maintain a connection with C&Cs, and receive commands.

During the collection period, we identified and monitored activity in C&Cs hosted on 566 distinct IP addresses. Bashlite and Mirai C&Cs launched 126,296 attacks against 40,449 distinct targets distributed in 2,855 autonomous systems.

IV. Botnet Infrastructure

In this section we discuss the evolution of Bashlite and Mirai support infrastructures, *i.e.*, C&C (Section IV-A) and malware server (Section IV-B) hosting.

[1] Bashlite C&Cs operate similar to Internet Relay Chat (IRC) channel, which allows Bashlite operators to interact while connected to a C&C.

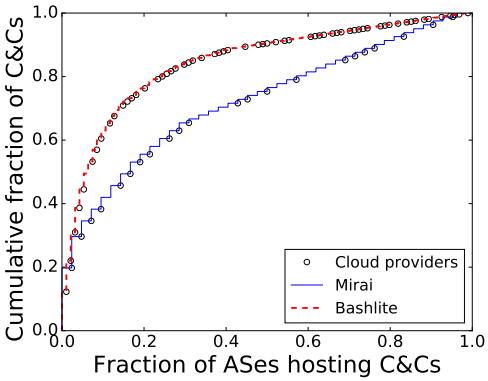

Figure 2. C&C locations.

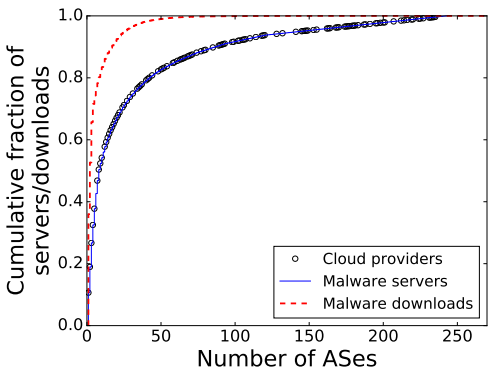

Figure 3. Malware server locations.

Table I
DISTRIBUTION OF C&C, MALWARE SERVER, AND ATTACK TARGET LOCATIONS BY AS TYPE

ENTITY		CDN/Hosting	Transit/Access	Enterprise	Others
Bashlite	C&Cs	73.39%	16.01%	3.12%	7.48%
	Targets	14.84%	82.20%	1.75%	1.20%
Mirai	C&Cs	72.84%	16.05%	1.23%	9.88%
	Targets	26.80%	67.94%	3.92%	1.34%
Malware servers		71.27%	18.56%	3.03%	7.15%

A. Command and Control Servers (C&C)

We identify C&C IP addresses to connect to by reverse engineering the malware URLs observed by honeypots during infection attempts. We were able to resolve 50.8% of the of Mirai C&C hostnames to IP addresses. We may fail to resolve a C&C hostname if, *e.g.*, its authoritative DNS server has been taken down. The fact that we can resolve a significant fraction of C&C hostnames illustrates the difficulty of taking down DNS servers, even if malicious.

We map Bashlite and Mirai C&C IP addresses to autonomous systems (AS) and country codes (CC) using Team Cymru's IP-to-AS database. We map 486 Bashlite C&C IP addresses to 93 ASes and 32 country codes, and 90 Mirai C&C IP addresses to 41 ASes and 21 country codes. We verified that all instances of IP addresses that mapped to multiple ASes were controlled by DigitalOcean, and mapped those IP addresses to DigitalOcean's most frequent AS (AS14061).

Figure 2 shows the cumulative distribution function of C&Cs as a function of the ASes where they are hosted. We observe a concentration of C&Cs in a few ASes. Although similar, the distributions show that Mirai C&Cs are spread across more ASes (less concentrated on the left), which may complicate takedowns and indicates more diversified hosting.

Table I shows the fraction of C&Cs hosted on different types of ASes, as classified by CAIDA's ASRank. We find that most C&Cs are hosted on cloud providers. Previous work have reported that some cloud providers do not cooperate in taking down C&Cs even after being notified of the malicious activity, also known as 'bulletproof' hosting providers [10], [11], [12]. When we look at the top 10 ASes that host most Bashlite and Mirai C&Cs, we find 7 ASes in common. This intersection may be a result of Bashlite and Mirai botnets being operated by the same groups, or a preference toward cloud providers more lenient towards malicious activity.[2]

B. Malware Servers

We also characterize the location of 1,955 IP addresses and 136 hostnames observed hosting malware, *i.e.*, appearing in malware payload URLs. We successfully resolved 42.7% of the malware server hostnames.[3]

Figure 3 shows the distribution of malware servers as a function of the ASes where they are located (blue line with circles). As with C&Cs, we observe that malware servers are concentrated in a few ASes, most of them cloud providers or CDNs (Table I). We also plot the distribution of malware download requests as a function of ASes where the server is located (dashed red line). We find download requests are concentrated in few servers (not shown) and ASes, which can be explained by larger botnets attempting infection and downloads from its malware servers more often.

V. EVOLUTION OF BOTNET ATTACKS

In this section we analyze the commands broadcasted by C&Cs (Section V-A), attack targets (Section V-B), coordination between different C&Cs (Section V-C), and operators's interactive sessions (Section V-D).

A. Attack Commands

Bashlite commands are transmitted in plain text. We observed 583 different command names in Bashlite; this large number of commands can be explained by different code variants renaming commands [5] (*e.g.*, from meaningful mnemonics to offensive words) or by operator typing errors (most

[2]The most notable infrastructure providers in the top 10 providers hosting the largest number of C&Cs are OVH.com and DigitalOcean. Providers such as Google, Amazon, and Microsoft host few C&Cs.

[3]The hostnames we could not resolve (57.3%) account for only a small fraction (0.1%) of malware download attempts in our dataset.

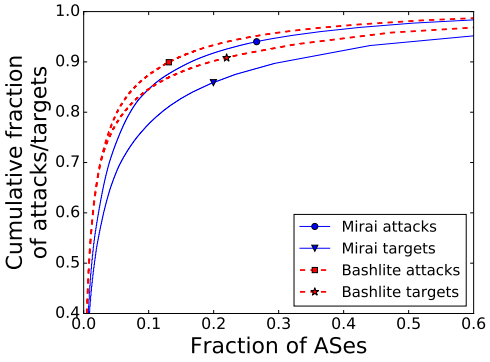

Figure 4. Distribution of targeted servers among ASes.

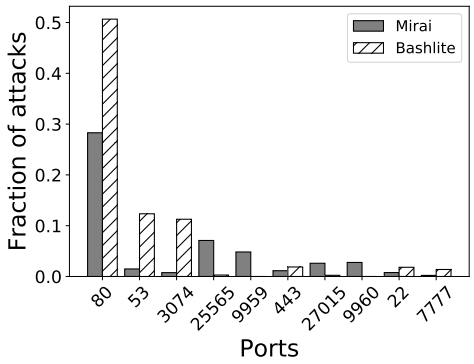

Figure 5. Fraction of attacks toward top-10 attacked ports.

commands occur only once in our data). To make the problem tractable, we consider only the top 80 most frequent commands, which account for 98.9% of all observed Bashlite commands. We could identify the semantics behind 72 of the top 80 commands. We found 48 commands in the source code of 39 Bashlite variants available in public online repositories, 17 commands had names equivalent to one of the previous 48 commands, and 7 commands were described in C&C login banners. We classify Bashlite commands into six classes:

- **Attack (66.4%)** commands start DDoS attacks against select targets: !* TCPFLOOD 192.168.0.1 80 120 32 syn
- **Management (18.4%)** commands, *e.g.*, to update the malware binary, remove bots from the botnet, or enable scanning: !* UPDATE, !* BOTKILL, or !* SCAN ON
- **Queries (1.27%)** retrieve information about the botnet and its state: !* HELP and !* STATUS
- **Interrupt (13.1%)** commands stop ongoing attacks, if any: !* KILLATTK
- **Other (0.70%)** commands that do not fit in any of the previous classes: !* CLEAR

Mirai C&Cs only broadcast attack commands. Botnet management is handled by separate services. Mirai supports 10 different attacks and transmits them using a binary protocol with no changes between variants, which allow us to accurately identify the semantics of each attack by looking at source code available on public online repositories.

Table II shows the distribution of Bashlite and Mirai attack commands available and executed. We group attacks into three classes similar to Antonakakis et al. [6]. Volumetric attacks are the simplest and attempt to exhaust bandwidth at the targeted device. TCP-related attacks exploit the TCP protocol to increase load at the targeted device's operating system (*e.g.*, SYN floods). The most sophisticated attacks are application-layer and attempt to overload a target application (*e.g.*, by submitting expensive queries). Comparing Mirai to Bashlite, we observe a significant shift from volumetric (flooding) attacks to TCP or application-aware attacks. These attacks are more effective and require fewer botnet resources to successfully degrade the target's quality of service.

Not only are Mirai attacks more sophisticated, but they

Table II
DISTRIBUTION OF COMMAND & ATTACK CLASSES FOR MIRAI COMMANDS & TOP 20 BASHLITE ATTACK COMMANDS

Entity		Volumetric	TCP-related	Application
Bashlite	Commands	40%	45%	15%
	Attacks	73.4%	13.6%	13%
Mirai	Commands	30%	20%	50%
	Attacks	30.9%	38.4%	30.6%

support a wider range of parameters. For example, the ACK attack allows configuration of 17 different fields and options of the IP and TCP headers. We find that 70.5% of Mirai attack commands use less than 20% of the available parameters, indicating that default parameters are meaningful, reducing the technical load on botnet operators, and possibly that the malware's capabilities are not yet used to their full potential.

B. Attack Targets

Bashlite attack commands may specify targets by IP address (94.9%) or hostname (5.1%). A reason to specify targets by IP address is to prevent geographically-distributed bots from resolving hostnames to different IP addresses. Mirai attack commands always specify targets by IP address. We were able to resolve 90.3% of target hostnames.[4]

Figure 4 shows the distributions of attacks and their targets (multiple attacks can have the same target) across different ASes. (We cropped both axes to improve legibility.) We observe that targets are concentrated on a few ASes and, as expected, attacks are even more concentrated. We also observe that Mirai targets are more evenly distributed across ASes (diverse) than Bashlite targets.

Table I shows the distribution of attack targets by AS type. We find that Transit/Access networks are the most attacked; this can be due to DDoS attacks targeting transit networks or their clients. However, we point out that Mirai operators more frequently attack CDN/Hosting and Enterprise networks, which are more directly related to services and businesses,

[4]The hostnames we could not resolve account for just 0.4% of attacks.

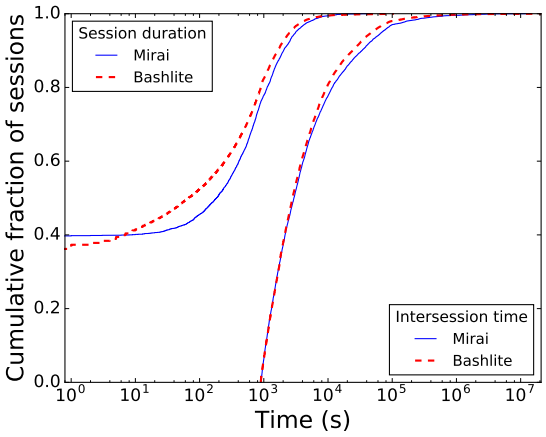

Figure 6. Session durations and intersession times.

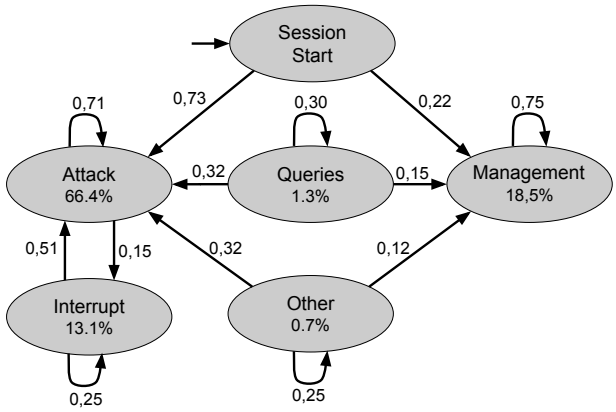

Figure 7. Transition graph for command classes within sessions.

respectively. As an example, OVH.com is the AS most frequently attacked by Mirai and the second most frequently attacked by Bashlite in our dataset.[5]

Figure 5 shows the fraction of attacks targeting each of the top 10 ports most frequently attacked by Bashlite and Mirai. Basic and common service ports are frequently attacked by Bashlite and Mirai. For example, HTTP, HTTPS, DNS, and SSH all figure in the top 10 attacked ports for the two families of botnets. Other notable attack targets are ports related to computer games. For example, the top 10 attacked ports also include those used by Xbox Live (3074), Minecraft (25565), Valve (27015), and TeamSpeak (9987). Although both Bashlite and Mirai botnets attack those ports (with different frequencies), it is important to note that Bashlite attacks are UDP or TCP SYN floods, while Mirai performs application-layer attacks specific to each service. For example, Mirai uses the TCP Stomp attack against Minecraft servers, and submits application-level queries to Valve servers. These observations are aligned with previous results showing that game servers are frequent attack targets [6], [14].

Finally, Mirai allows operators to target a network prefix, which causes bots to attack random computers in that prefix, or target multiple addresses in a single attack. Although these options are seldom used (0.1% and 1.1% of attacks, respectively), it illustrates available functionality that could be employed to perform attacks that are harder to detect and mitigate (as attack traffic is spread across multiple destinations). We also note that 2.82% commands exhibit operator errors which prevent the correct execution of the attack, such as not specifying the domain in DNS/HTTP attacks, supporting our view that some operators do not fully comprehend basic requisites of the botnet operation.

C. Coordinated Attacks

We define a pair of attacks from different C&Cs as *coordinated* if they have a common target and are issued within a time window of 60 seconds (results for time windows between 5 and 120 seconds are qualitatively similar). We find that 3.5% of Bashlite and 7.4% of Mirai attacks are coordinated. Although attacks may be issued from multiple C&Cs for different reasons, including coincidence, we argue that the increase in the fraction of coordinated attacks is another indication of more sophisticated botnet use and management. For example, coordination allows partitioning of a large botnet into smaller ones, facilitating multitasking (*e.g.*, configuring subsets of bots to attack different targets) and improving resistance against takedowns.

[5]OVH has reported to be a common target of DDoS attacks by botnets [13].

D. Botnet Operation Sessions

To understand the process by which operators manage and use botnets, we group commands issued by operators into *sessions*. We define a session as a sequence of commands separated by no more than a (configurable) period of δ seconds without activity. We plot the number of sessions as a function of δ and choose a value of δ close to the 'knee' of the curve, i.e., a value after which increasing δ has small impact on the number of sessions. In particular, we set $\delta = 900$ seconds.

Figure 6 shows the distributions of operator session durations (left side) for Bashlite and Mirai. We find that sessions are usually short, with approximately 40% of sessions comprising a single command ($x = 0$), and approximately 80% of sessions lasting less than 15 minutes. As expected, we find that the number of commands is correlated with session duration (not shown). Figure 6 also shows the distribution of the intersession times (right side). By definition, the intersession times start at $\delta = 900$ seconds and are reasonably long (note the logarithmic scale on the x-axis). This behavior indicates that botnet operators, in general, work in bursts, issuing several commands in a short session followed by long periods of inactivity.

To better understand the activities in a session, Figure 7 shows a transition graph among different command types for Bashlite sessions (Mirai sessions only contain attack commands and are not considered in this analysis). The nodes in the graph correspond to Bashlite command classes (Section V-A), and to the start of a session. Edges represent

command sequences. We compute the weight of the directional edge from node u to node v as the probability of a command of class u being succeeded by a command of class v. Edges from the initial state indicate the first commands in a session. We omit edges with weight less than 0.1 to improve readability.

The graph shows that 73% of sessions start with an attack or management command, indicating those are by far the most common reasons to start interaction with the botnet. We note that there is no command to terminate sessions, but sessions usually (19.7%) end with an interrupt command. We see that 71% of attack commands are followed by attack commands. Manual inspection indicates that those sequences of attack commands implement concurrent attacks against multiple targets or a long-duration attack against a single target. More than half of interrupt commands are followed by new attacks. Management commands are usually followed by management commands, which may indicate some difficulty in managing Bashlite botnets (*e.g.*, updating the malware, scanning for vulnerable devices to grow the botnet) and explain the implementation of dedicated management services in Mirai. Finally, attack commands and management commands, although common, are mostly unrelated. In particular, less than 1% of attack commands are followed by management commands.

VI. Related Work

IoT security. Many IoT devices execute special-purpose software that vendors rarely update. Even if vendors do provide updates, end users are frequently not interested in or lack the technical skills to install them. Moreover, some embedded devices have weak or leaked passwords that allow remote access. These and other factors have motivated the development of malware software to build botnets [7], [5].

Botnet characterization. A common challenge is observing the behavior of botnets (frequently using honeypots) without contributing to their operation or attacks [7], [5], [8]. In this work, we used low-interactivity honeypots and monitors that never contribute to malicious activities.

Previous work have characterized botnets and proposed attack mitigation mechanisms. More related to our work are studies characterizing the Bashlite and Mirai botnets [5], [6], [3] (we summarize some of these results in Section II). Our characterization complements previous work and improves our understanding. We use a different dataset and, more importantly, focus on the evolution of the malware, executed attacks, and operator practices, which have not been thoroughly discussed in previous work.

DDoS attacks. DDoS attacks are a real threat, and have caused significant impact to end users and businesses [3]. Researchers have studied DDoS attacks for more than a decade, classifying and creating taxonomies [15], [16], [1]; characterizing their impact [17], [18]; or revealing the DDoS attack underground market [9], [3], [5]. Our work helps us better understand DDoS attacks and our results may support the development of novel countermeasures.

VII. Conclusion

We have studied the behavior of the Bashlite and its successor Mirai IoT botnets using data collected by 47 low-interactivity honeypots deployed across Brazil. Overall, our results show that these botnets and their operators have evolved significantly. In particular, we show that botnet infrastructure has become more resilient and easier to manage. We also show that the software has integrated more effective attacks. Finally, we provide indication that botnet operators are becoming more proficient at managing and exploiting the capabilities of their botnets, *e.g.*, choosing effective application-layer attacks for each target, automating management, and coordinating attacks.

References

[1] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks," *IEEE communications surveys & tutorials*, vol. 15, no. 4, 2013.

[2] Neustar, "Worldwide DDoS Attacks & Cyber Insights Research Report," Online, May 2017. [Online]. Available: https://hello.neustar.biz/201705-Security-Solutions-DDoS-SOC-Report-LP.html

[3] K. Angrishi, "Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets," 2017.

[4] Symantec, "Internet Security Threat Report, Volume 22," Online, April 2017. [Online]. Available: https://www.symantec.com/security-center/threat-report

[5] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[6] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the mirai botnet," in *Proc. of USENIX Security Symposium*, 2017.

[7] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "Iotpot: Analysing the rise of iot compromises," in *Proc. of USENIX Workshop on Offensive Technologies*, 2015.

[8] Y. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "Iotpot: A novel honeypot for revealing current iot threats," *Journal of Information Processing*, vol. 24, no. 3, pp. 522–533, 2016.

[9] J. J. Santanna, R. van Rijswijk-Deij, R. Hofstede, A. Sperotto, M. Wierbosch, L. Z. Granville, and A. Pras, "Booters: An analysis of ddos-as-a-service attacks," in *Proc. of IEEE/IFIP International Symposium on Integrated Network Management (IM)*, 2015.

[10] A. K. Sood and R. J. Enbody, "Crimeware-as-a-service—a survey of commoditized crimeware in the underground market," *International Journal of Critical Infrastructure Protection*, vol. 6, no. 1, 2013.

[11] M. Goncharov, "Criminal hideouts for lease: Bulletproof hosting services," 2015.

[12] M. Konte, R. Perdisci, and N. Feamster, "Aswatch: An as reputation system to expose bulletproof hosting ases," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, 2015.

[13] "The ddos that didn't break the camel's vac*," https://www.ovh.com/world/news/articles/a2367.the-ddos-that-didnt-break-the-camels-vac, accessed: 2018-02-16.

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

[15] J. Mirkovic and P. Reiher, "A taxonomy of ddos attack and ddos defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, 2004.

[16] E. Cooke, F. Jahanian, and D. McPherson, "The zombie roundup: Understanding, detecting, and disrupting botnets." *Proc. of the Steps to Reducing Unwanted Traffic on the Internet Workshop*, 2005.

[17] S. Behal and K. Kumar, "Characterization and comparison of ddos attack tools and traffic generators: A review." *IJ Network Security*, vol. 19, no. 3, 2017.

[18] A. Wang, A. Mohaisen, W. Chang, and S. Chen, "Delving into internet ddos attacks by botnets: Characterization and analysis," in *Proc. of IEEE/IFIP International Conference on Dependable Systems and Networks*, 2015.