Honor Among Thieves: Towards Understanding the Dynamics and Interdependencies in IoT Botnets

Jinchun Choi[1,2], Ahmed Abusnaina[1], Afsah Anwar[1], An Wang[3], Songqing Chen[4], DaeHun Nyang[2], Aziz Mohaisen[1]
[1]University of Central Florida [2]Inha University [3]Case Western Reserve University [4]George Mason University

***Abstract*—In this paper, we analyze the Internet of Things (IoT) Linux malware binaries to understand the dependencies among malware. Towards this end, we use static analysis to extract endpoints that malware communicates with, and classify such endpoints into targets and dropzones (equivalent to Command and Control). In total, we extracted 1,457 unique dropzone IP addresses that target 294 unique IP addresses and 1,018 masked target IP addresses. We highlight various characteristics of those dropzones and targets, including spatial, network, and organizational affinities. Towards the analysis of dropzones' interdependencies and dynamics, we identify dropzones chains. Overall, we identify 56 unique chains, which unveil coordination (and possible attacks) among different malware families. Further analysis of chains with higher node counts reveals centralization. We suggest a centrality-based defense and monitoring mechanism to limit the propagation and impact of malware.**

***Keywords*-Internet of Things, Malware, Static Analysis, Distributed Denial of Service**

I. Introduction

The use of Internet of Things (IoT) devices in everyday life has been increasing significantly. Gartner, a global research and advisory firm, predicts that the number of IoT devices will grow to 25 billion by 2021, from 14.2 billion devices in 2019 [1]. As the number of applications of IoT devices is significantly increasing, so is the number of malicious software (malware) targeting IoT, which have seen a consistent increase in the past few years as well [2]. In part due to their sheer number, as well as the constrained operation environments (e.g., limited standards, lack of maintenance and updates, etc.), IoT devices are more likely to be a target for malware infections, bringing about botnets used often in launching catastrophic Distributed-Denial-of-Service (DDoS) attacks [3]–[6].

Bashlite (also known as Gafgyt, LizardStresser, Lizkebab, Qbot, and Torlus) is a malware family that uses default login IDs and passwords to propagate and infect targets [7]. Based on this infection, the Bashlite malware family made a large botnet that is capable of launching large DDoS attacks [8]. Among many capabilities that Bashlite has, it can update bots payload, in continuous evolution and morphing, and—more interestingly–remove competing botnets from infected hosts [9]. Mirai is the most notorious malware successor of the Bashlite, and has been used for demonstrating some significant damages by launching DDoS attacks several times on several pieces of critical infrastructure and prominent services [10]. Among other incidents, Mirai's attacks temporarily disrupted "Krebs on Security" (security blog) [11], OVH (cloud computing company) [12], and Dyn (domain service company) [13]. In the case of the Dyn attack, many other services, such as Airbnb, Github, and Twitter, among others, were indirectly affected. The same botnet was able to take down the Internet of the African nation of Liberia [14].

Competition and coordination among botnets are not well explored, although recent reports have highlighted the potential of such competition as demonstrated in adversarial behaviors of Gafgyt towards competing botnets [9]. Understanding a phenomenon in behavior is important for multiple reasons. First, such an analysis would highlight the competition and alliances among IoT botnets (or malactors; *i.e.,* those who are behind the botnet), which could shed light on cybercrime economics. Second, understanding such a competition would necessarily require appropriate analysis modalities, and such a competition signifies those modalities, even when they are already in use. Third, the signified modalities may shed light on possible effective defenses; e.g., a piece of infrastructure used by a majority of infected hosts in an IoT botnet makes an excellent candidate for a botnet takedown.

To address IoT security, malware behavior analysis methods are employed, including dynamic and static analysis. Dynamic analysis is concerned with understanding malware by inspecting runtime artifacts of IoT executables (typically running in a restricted environment; e.g., sandbox, virtual machine), in search of malicious behaviors. Despite many advantages, dynamic analysis has several drawbacks. For example, recent malware families have been shown to utilize randomized behaviors that make analysis difficult. Evading dynamic analysis techniques is yet another major shortcoming, often demonstrated by inserting fake code fragments, separate processes, etc. [15]. Dynamic analysis is also time-consuming since a successful analysis needs the malware to run for a significant amount of time before such a dynamic behavior is exposed. Static analysis, on the other hand, is concerned with analyzing the contents and the structure of the executables. Through this analysis, we can find features of malware such as execution-flow as well as the strings without having to execute the binaries [16]–[18], making this approach safer and faster than dynamic analysis [19], although subjecting to static analysis circumvention techniques, such as code obfuscation [20], [21], typically addressed with deobfuscators.

In this paper, we attempt to understand the dynamics between different IoT botnets through the lenses of static analysis hoping to unveil competitive behaviors among those botnets. By obtaining endpoints from the residual strings of IoT malware binaries upon static analysis, we proceed to categorize those endpoints based on the context in which they appear into dropzones and targets. `Dropzones` are Internet Protocol (IP) addresses used along with specific words, such as `wget`, and are used by the malware to control bots and to retrieve updated malware binaries (payloads) or scripts from an external server. `Targets`, on the other hand, are IP addresses subjected to an attack by the malware sample being analyzed, as indicated by several keywords.

Our analysis shows that most target IP addresses are masked (16-bit), in which a malware dynamically specifies targets at runtime. However, we found some target addresses that are not masked, identified as static IP addresses. Those are IP addresses that the attacker considers explicitly as targets in the malware code. Given the certainty around those IP addresses, we focused on them and analyzed the interactions between dropzones and the statically targeted IP addresses in our static analysis. Through our analysis, we found that many of the statically targeted IP addresses are identical to the dropzone IP addresses obtained from other binaries, while the majority of the remaining targeted IP addresses are marked as malicious in our VirusTotal [22] scan. Such a discovery is very interesting, highlighting a non-arbitrary behavior among botnets and other malactor's infrastructure. Given that the number of targets is a negligible percentage of the IPv4 space, those targets cannot be random, and that highlights the potential adversarial relationship between the dropzones.

Contributions. We make the following contributions:

1) We conduct string static analysis over IoT malware binaries to extract communicated and referred IP addresses, and keywords reflecting the malware behavior.
2) We conduct a spatial distribution analysis on the extracted dropzones and targets, where region dependencies within the extracted addresses are shown.
3) We identify the interdependent dropzone IP addresses, by extracting dropzones "chains", capturing the dynamics between different botnets and other malicious infrastructure. We propose a centrality-based modality of analysis (and defense) to limit the propagation and impact of malware based on those dynamics.

Organization. The organization of the rest of this paper is as follows. We describe the background in section II. Next, the analyzed dataset is listed in subsection II-B. In section III, we provide our analysis and results. The discussion of this paper is provided in section IV. In section V, we review the related work and, and draw concluding remarks in section VI.

II. Background, Dataset, and Static Artifacts

A. Static Analysis and Endpoints

Static Analysis. In this paper, we employ static analysis for extracting residual strings in the IoT malware binaries, and use those strings as an analysis space from which we obtain endpoints, classified as targets and dropzones. In static analysis, reverse-engineering tools are utilized to understand circumvention methods in use by the adversary, and to extract static artifacts, such as strings, function calls, structures (such as control flow graph), etc. However, techniques such as packing or obfuscation can be used to avoid static analysis or to increase the effort and resources required for conducting it. In a separate project, we developed various in-house heuristics and tools to address obfuscation, and to obtain faithful strings representation for IoT malware, which we use in our analysis in this paper. The results of our analysis are further in §III-A.

TABLE I: The distribution of malware families in the dataset.

Family	Count	Percentage
Gafgyt	4,264	88.76%
Mirai	507	10.55%
Tsunami	29	0.60%
Singleton	2	0.04%
Pilkah	1	0.02%
Sambashell	1	0.02%
Total	4,804	100%

Dropzones and Targets. IP addresses extracted from the malware binary through static analysis are classified into two categories. If the IP address was used with `wget`, `tftp`, `get`, or `post`, which are commands used to send files such as script, malware binary, etc., we mark the remote location of this IP address as a dropzone. The remaining IP addresses that are not used with these commands are called targets, which are the IP addresses attacked by the malware (confirmed through the manual inspection). The target IP addresses consist of 16-bit masked addresses and static IP addresses. Only the static IP addresses are used in our analysis. An additional explanation for this is given in Section III.

B. Dataset

In this paper, we relied on a dataset of IoT malicious binaries obtained from CyberIOCs [23]. The binaries are recent, and consist of samples that were collected in the CyberIOCs feed in the period of January 2018 to late January of 2019.

Dataset Creation. Our IoT dataset is a set of 4,804 malware samples, randomly selected from CyberIOCs [23]. We reverse-engineered the samples using *Radare2* [24], a reverse engineering framework that provides various capabilities including disassembly, which we use for the IoT malware samples.

IoT Malware Family. To better understand the collected samples, we uploaded the samples to *VirusTotal* [22] and gathered the scanning results corresponding to each sample. Then, we used AVClass [25] to match the samples with their corresponding IoT malware families. Table I shows the distribution of malware families in the dataset; as shown Gafgyt and Mirai represent the majority of our dataset, 99.31%.

III. Analysis

A. IoT Malware Static Analysis

We reverse-engineered the malware binaries to extract the IP addresses communicated with or referred by the malicious binaries. To scale up the analysis, we automate the process using Radare2. We analyzed the strings in the entry point and the function calls to extract the IP addresses, where two types of IP addresses can be extracted: (1) C2 servers communicated by the malware for instructions, such as targets list, malware binaries execution, etc. Such IP addresses can be identified by

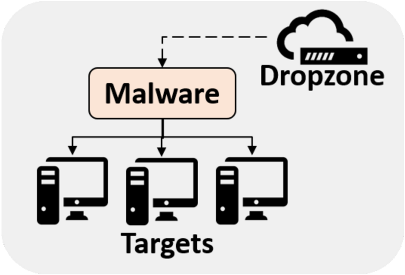


Fig. 1: General structure of the malware-IP relationship. Malicious binaries are obtained from the remote dropzone and are accessible using *wget*, *GET*, etc.

command keywords, namely `wget`, `tftp`, `post`, and `get`. These IP addresses are designated as dropzone IP addresses. (2) IP addresses referred by the malware, e.g., the malware communicates with the IP address to infiltrate where successful infiltration causes propagation of the malware, recruiting an additional bot. These IP addresses are called *target* IPs.

1) Dropzone: Controlled by the attackers, a dropzone is a remote location often storing the malware binaries and infection capabilities. Upon gaining access to a device, a malware instance accesses the dropzone, via a dropzone IP address, to download the file on the host device. The mentioned remote addresses are our artifacts of interest, and we study relationships between different dropzones' IP addresses.

2) Target: Upon successfully infecting a device, the malware uses the infected host to propagate the infection by setting a list of IP addresses to infect in the future. We refer to these IP addresses as target IP addresses.

We collected the dropzones and target IPs from each IoT malware sample, to be analyzed in the next sections. Figure 1 shows a general structure of the malware-IP relationship.

B. IP Addresses Analysis

To better understand the relationship between the dropzones, we start by analyzing the IP addresses. In our dataset, we observed 1,457 unique dropzone IP addresses and only 294 unique target IP addresses. Moreover, there were 1,018 unique 16-bit masked target IP addresses. These IP addresses are generated at runtime using a random number generator, particularly, the *SRAND* C library, or by looping over all possible IP addresses within the specified network. Typically, masked IP addresses are used to infect and compromise vulnerable IoT devices within a network. Figure 2 shows the distribution of the dropzones, unique targets, and masked targets. Notice that most of the dropzones are located in the US and Europe. However, most of the masked targets are located in Southeast Asia, Brazil and the Eastern Coast of the US.

1) Unique Target IP Addresses: Our analysis focuses on the unique target IP addresses referred to in the IoT malware. These addresses are more meaningful than the masked IP addresses as they are hardcoded within the malware. In the dataset, we extracted 294 unique target IP addresses. In which, 134 of the addresses are dropzones of other malware samples in our dataset. We scanned the remaining IP addresses using VirusTotal, where 129 of them were identified as malicious. These malicious IP addresses might be potential dropzones not existing in our dataset, or infrastructure utilized by other malactors. In addition, the benign IP addresses might be dropzones not yet discovered by VirusTotal, or DDoS attacks targets, which is more plausible given that the dataset is new, and blacklists against which our IP addresses were scanned take time to populate with the malicious addresses. Table II shows the distribution of the unique target IP addresses.

TABLE II: The distribution of the unique target IP addresses.

Type	Count	Percentage
Dropzone	134	45.58%
Malicious	129	43.88%
Benign	31	10.54%
Total	294	100%

TABLE III: Organization distribution of the benign target IPs.

Organization	**Type**
Amateur Radio Digital Comm.	Nonprofit
Apple Inc.	Technology
Bank of America, N.A.	Financial
Ford Motor Company	Automaker
Hewlett Packard Enterprise	Technology
Lockheed Martin Corporation	Aerospace/Defense
University of Michigan	Academic
Information Center	**Locality**
AFRINIC	Africa
APNIC	Asia Pacific
DoD NIC	USA
RIPE NCC	Europe

2) Targeted Organizations: We scanned the benign IP addresses extracted from the IoT malware (31 addresses) and gathered the organizations they belong to. We found that the IP addresses belong to companies such as Apple, Bank of America, Ford, etc. Moreover, one IP address belonged to the University of Michigan, while some IP addresses belonged to endpoints in different information centers. Table III shows the distribution of the benign target IP addresses over organizations and information centers, highlighting a wide distribution.

3) Dropzones Malware Family: Dropzones are remote locations storing malware binaries, among other artifacts by the adversary. When a new device is infected, it will communicate with the dropzone to obtain the malicious binaries, along with the infection capabilities, which vary for each family. We analyze the malware family of each dropzone. Table IV shows the distribution of the dropzone malware families, with Gafgyt malware binaries existing in 89.63% of the dropzones, followed by Mirai (10.57%). In addition, we found that different families of malware binaries contained the same dropzone. In other words, some dropzones contain more than one family binaries; Table V shows the distribution of the

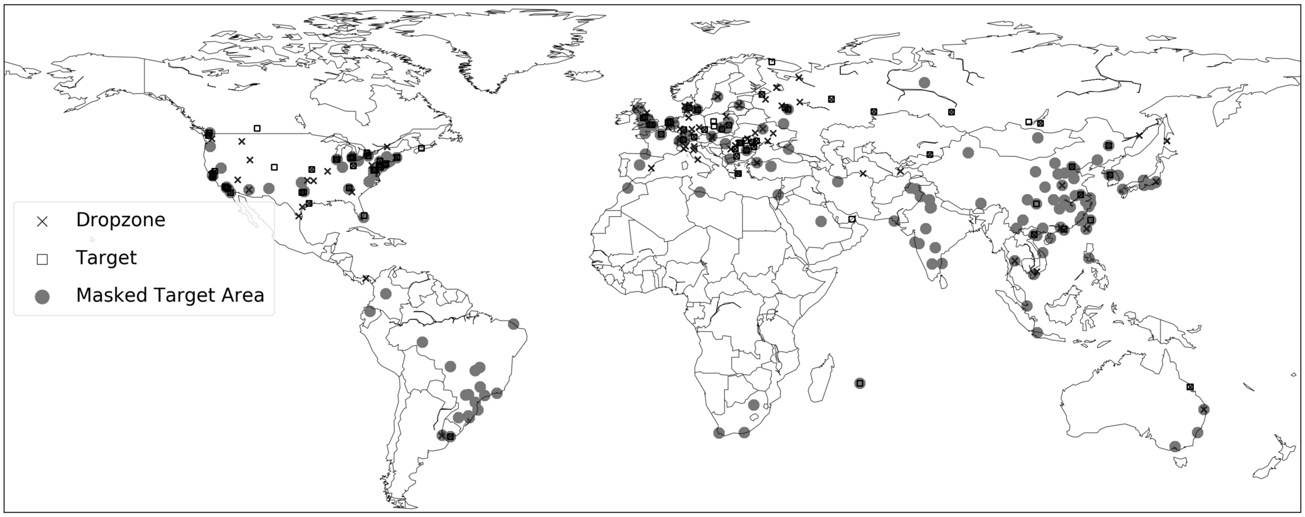


Fig. 2: The distribution of the extracted components worldwide. Here, the target refers to the location of the unique target IP address extracted from the string analysis of the IoT malware. The masked target is represented by its center. To estimate the locations of the masked target area, we convert the masked part to zeros (*i.e.,* 183.229.%d.%d to 183.229.0.0).

TABLE IV: The distribution of dropzone malware families.

Family	# Dropzones	% Dropzones
Gafgyt	1,306	89.63%
Mirai	154	10.57%
Tsunami	14	0.96%
Singleton	2	0.14%
Pilkah	1	0.07%
Sambashell	1	0.07%
Total	1,457	100%

TABLE V: Malware families distribution per dropzone.

# Families	# Dropzones	% Dropzones
1	1,437	98.63%
2	19	1.30%
3	1	0.07%
Total	1,457	100%

malware families per dropzone. We notice that one dropzone contains the malicious binaries of Gafgyt, Mirai and Tsunami families, highlighting the shared infrastructure.

4) Dropzones Distribution: Dropzones have spatial localities in their distribution, as shown in Figure 2. Moreover, Figure 3 shows a heatmap of the country distribution of the dropzones, where the United States, Netherlands, Denmark, Romania, and Russia are hosting 77.82% of the dropzones. Table VI shows the top dropzones hosting countries, highlighting—not surprisingly—a heavy-tailed distribution.

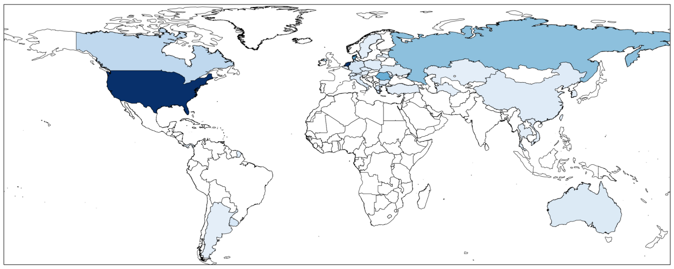


Fig. 3: The distribution of dropzones over the countries. The color shade reflects the number of dropzones within the country, where darker shade represents more dropzones.

TABLE VI: The distribution of the dropzones over countries.

Country	# Dropzones	% Dropzones
United States	553	37.95%
Netherlands	283	19.42%
Denmark	113	7.75%
Romania	103	7.07%
Russia	82	5.63%
Others	323	22.18%

To this end, we have analyzed the IP addresses as independent entities. However, our static analysis shows that different IP addresses are communicating and being referred by each other. As such, it is important to study the relationship between them, and among dropzones in particular.

C. Malware-Target Relationship

The IP addresses referred by malicious binaries are the next targets of the malware for either infection or attack. The malware samples communicate with a target to achieve one of the following:

- **Infection.** Malware search for vulnerable devices and compromise them, leading to a new bot. Afterward, malicious binaries with infection capabilities are downloaded from the remote dropzone to the infected host. The IP address is typically generated at runtime and referred to in the code as the masked IP address.

- **Attack.** After infecting a large number of bots, malware samples attack the target by flooding its servers or network infrastructure with packets, resulting in a DDoS attack. Typically, these IP addresses are not masked as the bot should be aware of the exact target IP address prior to the DDoS attack.
- **Communication.** A malware sample might communicate with an infected bot for many reasons, e.g., checking its status, updating/pushing files, sending a command message, etc. The communication is very important for the malware to assess the resources and coordinate for future attacks.

Limitation. Static analysis is useful to understand the behavior without the need to run the malicious binaries. In the malicious binaries, we observe keywords such as `Infect`, `wget`, `post`, `push`, `http`, and `get`. These keywords indicate the relationship between the malware and the targets. However, besides `wget`, it is hard to match the exact IP address to a certain behavior if more than one keyword is used. Therefore, we assume that all IP addresses might contain the behaviors provided by the keywords associated with them.

D. Dropzones Chains

In subsection III-B, we classified the IP addresses and found that 134 target IP addresses are also dropzones existing in our dataset. A malware may control a dropzone, and targets another dropzone; we refer to this phenomenon as dropzones chain. Figure 4 shows the general structure of a dropzones chain of length two. Understanding the chains is important, as malware may access a dropzone to distribute and update its binaries on other dropzones. Moreover, a dropzone may control several dropzones, forwarding commands and managing attacks. Using static analysis alone is not sufficient to understand the exact role of each dropzone within the chain. However, the behavior of the chain is toward propagating information, which plays a major role in the success of the malicious attacks. Figure 5 shows the dropzone to dropzone chain links visualization worldwide. Notice that some dropzones are directly linked to several dropzones.

1) Chains Length: We extracted 56 possible chains from our dataset. The majority of the dropzones (62.5%) are of length 2, and most of the dropzones are of a length less than 10 (96.43%). However, the longest chain has 42 dropzones. Table VII shows the distribution of the dropzones chains length. We observe that centralization exists in chains with a high number of dropzones. All of the extracted chains belong to the Gafgyt and Mirai families. We found one Mirai chain of length 2 and 52 Gafgyt chains. However, there were 3 chains containing both Mirai and Gafgyt dropzones. One possible explanation for such a characteristic is that Mirai is considered an evolution of the Gafgyt malware family [10], [26].

2) Chains Region Distribution: Dropzones within the chain have locality characteristics. Figure 6 shows the country distribution of the links within the chains. Notice that a darker shade indicates more dropzones within the chain are from the specified country. Table VIII shows the top five countries hosting dropzones within chains. Notice that the countries are the same as Table VI. It can be seen that 24.95% of the United States dropzones are within chains, with an overall 20.66% of the dropzones are within chains. However, the chains are depending on the collected dataset, meaning that the remaining dropzones (79.34%) may be part of chains not observed by the collected samples.

3) Chains Centrality: A common observation we make is that large chains usually have one or a few central dropzones. For instance, one remote location is a dropzone of a large number of other dropzones. This indicates the importance of that dropzone for the malware to successfully operate. Removing or monitoring the central dropzones highly affects the malicious operation of various samples, and improving our understanding of the malware behavior/defense. Figure 7 shows the effect of removing the central dropzone from the chain. Here, the chain contains 42 dropzones, connected by 44 edges (links), with a central remote location acting as a dropzone for 34 dropzones. In this figure, a directed edge indicates that a remote location is a dropzone (start of the arrow) to another dropzone (end of the arrow). Therefore, removing the central dropzone decrease the number of edges from 44 to 10 (77.27% decrease). Removing the dropzone from the network can be done by the Internet Service Provider (ISP), as the dropzones have static IP addresses, with known home ISPs. Moreover, another feasible option is to monitor

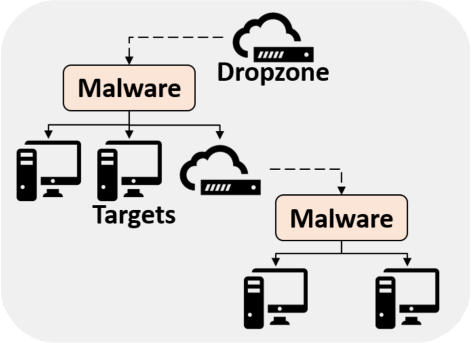


Fig. 4: Structure of the dropzones chain of length two, where malware access dropzone, and targets another dropzone.

TABLE VII: The distribution of the dropzones chains length.

Chain Length	Count
2	35
3	10
4	5
6	3
8	1
17	1
42	1
Total	56

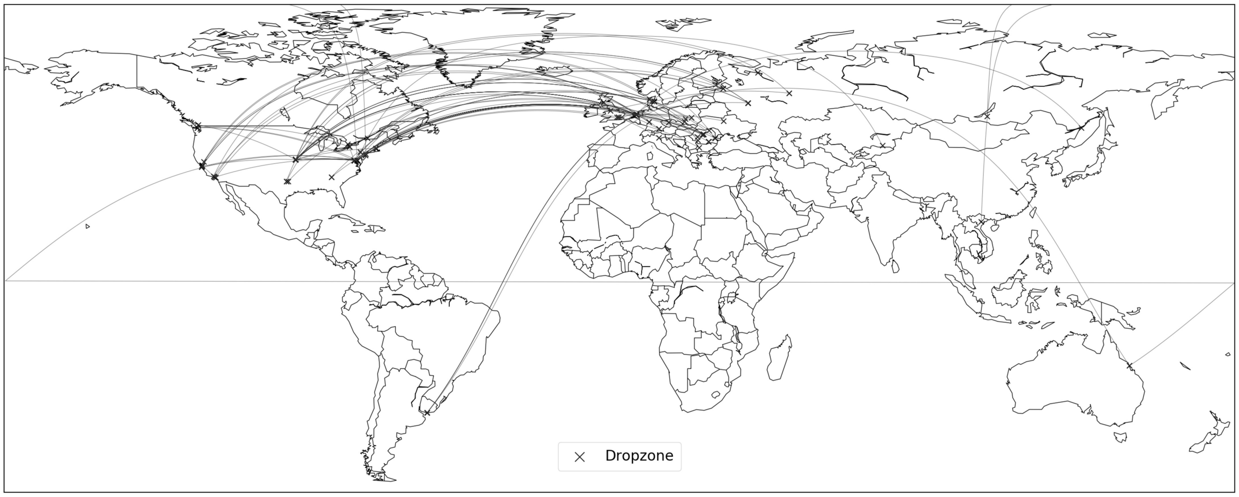


Fig. 5: Dropzone to dropzone links visualization. Here, a remote location may be a dropzone of several dropzones. Links connect the location of the dropzone with the location of the targeted dropzone.

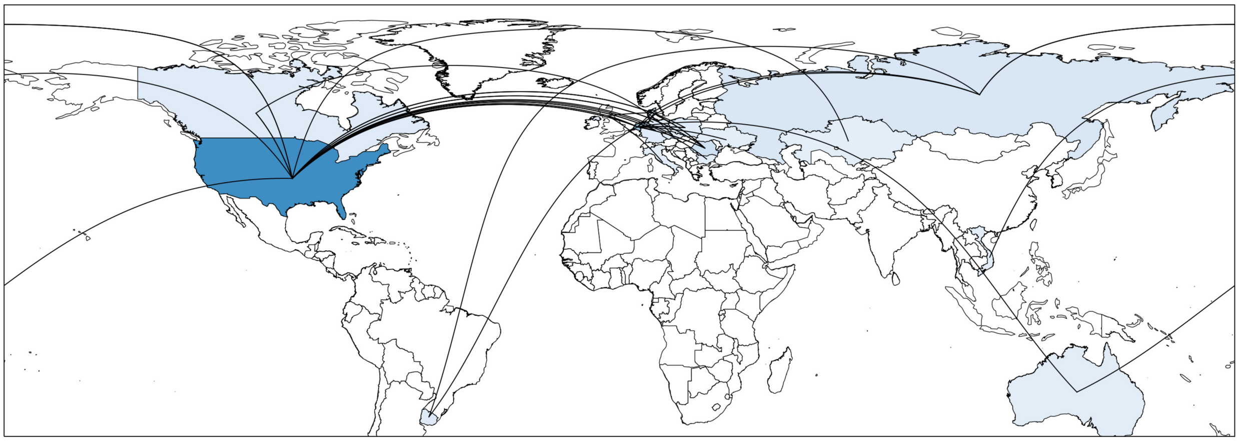


Fig. 6: Country level distribution of the dropzones within the chains. Color shade indicates the number of dropzones within the country. Here, link connect the country of the dropzone with the country of the targeted dropzone.

TABLE VIII: The distribution of the chains dropzones over countries. Here, # Dropzones is the number of dropzones within the chain, whereas, % Dropzones is the percentage of the dropzones within the country that is within a chain.

Country	# Dropzones	% Dropzones
United States	138	24.95%
Netherlands	71	25.09%
Romania	27	26.21%
Denmark	16	14.16%
Russia	14	17.07%
Others	35	10.83%
Total	301	20.66%

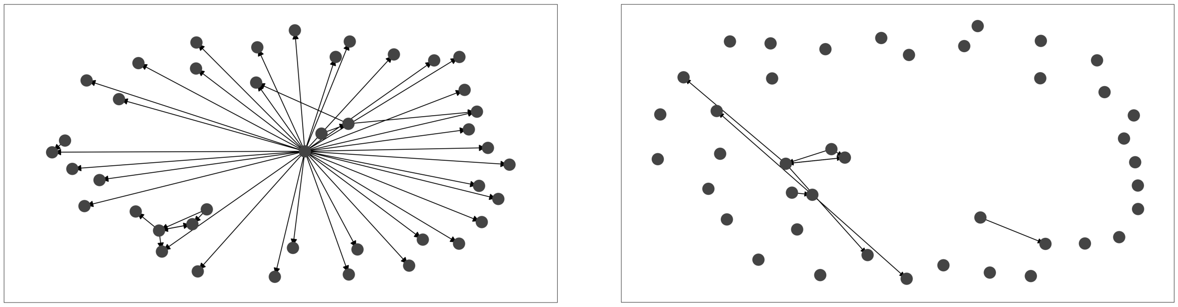
the traffic from and to the central dropzone of each chain, as monitoring all bots or dropzones might not be possible. In a related analysis, we show that dropzones are accessed to obtain malicious binaries, infection capabilities, and attacks coordinating. Therefore, monitoring the traffic of the central dropzone gives an overview of the malware behavior as it acts as a dropzone for a large number of dropzones and samples.

IV. Discussion

A. Key Findings and Implications

We note that all the analysis previously done depends on the collected dataset. By reverse-engineering the IoT malicious binaries and conducting string static analysis, it has been shown that some IP addresses are specified within the code prior to the execution of the program. These IP addresses are within two groups, dropzones, and targets.

1) Dropzone-Target Relationship: A remote location may be used as a dropzone for a set of dropzone targets. This indicates a relationship between the dropzones. The nature and role of the relationship may vary, as malware may use a master dropzone to update the binaries of a set of local dropzones or control the dropzones in order to coordinate future attacks or large area infections. In addition, large chains are usually centralized, where there is one or a few central dropzones acting as a dropzone for most of the chain dropzones. Analyzing the traffic of the central dropzones may



(a) Before

(b) After

Fig. 7: The effect of removing the central dropzone over the chain. Figure 7a is the chain before the removal of the dropzone, and Figure 7b is the same chain after the removal of the dropzone. Notice that most dropzones became disconnected and isolated, thus removed from the chain.

help us understand the behavior of the malware, while shutting down the central dropzone highly affects the operation of the malware, especially at early stages. This can be done with the coordination with ISPs as the dropzone IP is known. Shutting down the central dropzone cuts the communication between it and other dropzones, disrupting the functionalities in §III-C

2) Families Intersection: It has been observed that one remote location may act as a dropzone for multiple IoT malicious families. In the dataset, this relationship exists within Gafgyt, Mirai, and Tsunami. Researchers reported a new type of IoT threat known as KTN-RM or Remaiten which targets IoT devices by combining the capabilities of Linux malware known as Tsunami, Gafgyt [27], [28]. In addition, it has been reported that Mirai is an evolution of Bashlite malware, including Gafgyt [10], [26]. Our analysis, although not concerned with the capabilities, hint on such an evolution from an infrastructure standpoint.

B. Limitations

Our study is not without limitations. The key shortcomings in our work have to do with the dataset and target behavior, as seen from the static analysis artifacts.

1) Dataset: Table I shows the distribution of the families in the dataset utilized in this paper. The dataset is biased toward Gafgyt and Mirai, where 99.31% of the samples belong to these families. This bias is reflected in the dropzones chains, as 92.86% of the extracted chains belong to Gafgyt family. Collecting more samples over longer period of time will enhance the quality of the analysis, and the extracted chains, which stands as our future work. We note, however, that those two families are the most popular by far, and the findings in this work are of significant value given their prevalence.

2) Target Behavior: String static analysis may still not address some ambiguity of the exact behavior of a certain target. To help understand the role of the referred IP addresses, we analyzed the samples with specific keywords. However, it is not possible to match the behavior to a certain target. Therefore, we assume that the behavior applies to all referred targets, as a form of extrapolation from a few tested samples. Establishing statistical confidence in the findings, though a larger baseline, would be our future work.

3) Dropzones Chains: All extracted chains are limited by the collected dataset. The chain quality and size might increase with the number of collected samples. In this dataset, all chains belong to Gafgyt and Mirai, although we speculate that other IoT malware families will have similar behavior. Moreover, as chains with various families may exist due to the combined malicious capabilities, we expect the limited number of families analyzed in this study will not affect the generality of the findings; a confirmation of the above anecdote.

V. Related Work

Malware Analysis. Malware analysis helps to understand the behavior of the malware, thus defending against it. Dynamic analysis executes malware in a monitored environment and observes its behavior and functionality [29], [30]. In contrast, Static analysis inspects the executable files without executing them. It analyzes the malware through the strings and function calls necessary for malware operation and structure. Cozzi *et al.* [31] performed static analysis on Linux malware. Also, they discussed how Linux malware build their malicious acts. Kendall *et al.* [19] described the static analysis in-depth, a malware executable file might be disclosing only basic properties such as file type, the checksum for file fingerprinting, simple extractable strings and Dynamic-Link Library (DLL) import information, or fully disassembled with powerful tools and specialized knowledge. One of the obstacles to static analysis of malware is code obfuscation. Moser *et al.* [21] examined the limitations of the static analysis in the detection of malicious code. Soliman *et al.* [32] set a taxonomy for the tools and analysis method. They analyze the pros and cons of each static and dynamic analysis approaches. The prior work focused on analyzing IoT malware and evolution. However, it is equally important to understand the communication and relationships between bots to fully understand the operation of a malware, thus defensing against them.

DDoS Attack with IoT Malware. Several IoT malware have the capability to launch DDoS attacks [28]. Mirai is one of the notorious IoT malware that is targeting vulnerable IoT devices such as Digital Video Recorders (DVRs), security cameras, routers [10]. Wang *et al.* analyzed multiple IoT malware and categorized them by the approach in which they infect targets, such as, brute-forcing the weak user credentials, and exploiting vulnerabilities found in the devices. They found that Mirai brute-force the target based on the dictionary of popular usernames and passwords [33]. Sinanović presented the result of the dynamic and static analysis of Mirai. They set up a virtual environment for dynamic analysis to replicate controlled DDoS attack [30]. In addition, Antonakakis *et al.* [10] investigated Mirai botnet and how they appeared. In particular, they studied the evolution of Mirai over time, along with the type of devices affected by it using static analysis. Similarly, Ceron *et al.* [7] studied DDoS capable malware, such as Mirai and Bashlite by handling the network traffic. They utilized Software-Defined Networking to control the network environment. Furthermore, De Donno *et al.* [34] studied the taxonomy of DDoS attacks in the different subject of IoT. They did a detailed analysis of how Mirai's design and components perform their attacks.

VI. Conclusion

In this work, we analyzed IoT malware binaries to understand the dependencies and relationships among malware. We conduct static analysis to extract the addresses communicated to or referred by the malware. Among a large number of endpoints (dropzones and targets) in static malware artifacts, we identified dependencies between dropzones, in which we coin the dropzones chain. We identified 56 unique chains and unveiled interactions among Gafgyt and Mirai families. Further analysis showed the existence of centralization within chains with higher node counts, where a central dropzone communicates with several dropzones in a decentralized fashion. We suggest central dropzone monitoring and removal, in order to understand and limit the impact of the malware.

Acknowledgment. This research was supported by Korea National Research Foundation under grant 2016K1A1A2912757 and a collaborative seed research grant from Cyber Florida.

References

[1] Priyanka, Dua. (2018) 25 billion connected things will be in use by 2021: Gartner. Available at [Online]: https://bit.ly/2JBcmwq.

[2] E. Leloglu, "A review of security concerns in Internet of Things," *Journal of Computer and Communications*, vol. 5, no. 01, p. 121, 2016.

[3] 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.

[4] N. Vlajic and D. Zhou, "IoT as a land of opportunity for DDoS hackers," *Computer*, vol. 51, no. 7, pp. 26–34, 2018.

[5] M. E. Ahmed and H. Kim, "DDoS attack mitigation in Internet of Things using software defined networking," in *Proceedings of the IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE, 2017, pp. 271–276.

[6] H. Mustapha and A. M. Alghamdi, "DDoS attacks on the internet of things and their prevention methods," in *Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*. ACM, 2018, p. 4.

[7] J. M. Ceron, K. Steding-Jessen, C. Hoepers, L. Z. Granville, and C. B. Margi, "Improving IoT Botnet investigation using an adaptive network layer," *Sensors*, vol. 19, no. 3, p. 727, 2019.

[8] G. Masters. (2016) Millions of IoT devices enlisted into DDoS bots with Bashlite malware. Available at [Online]: https://bit.ly/2LV1ew9.

[9] V. Mark, G. Byron, and R. Augusto. (2019) Bashlite IoT Malware updated with mining and backdoor commands, targets WeMo devices. Available at [Online]: https://bit.ly/2OKfirI.

[10] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis *et al.*, "Understanding the Mirai Botnet," in *Proceedings of 26th USENIX Security Symposium (USENIX Security 17)*, 2017, pp. 1093–1110.

[11] B. Krebs. (2016) KrebsOnSecurity hit with record DDoS. Available at [Online]: https://bit.ly/2dn9If6.

[12] G. M. Graff. (2017) How a dorm room minecraft scam brought down the Internet. Available at [Online]: https://bit.ly/2j2RTCO.

[13] J. A. Jerkins, "Motivating a market or regulatory solution to IoT insecurity with the Mirai botnet code," in *Proceedings of the IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2017, pp. 1–5.

[14] Nicky, Woolf. (2016) Massive cyber-attack grinds liberia's internet to a halt. Available at [Online]: https://bit.ly/2elh92w.

[15] K. K. Ispoglou and M. Payer, "malWASH: Washing malware to evade dynamic analysis," in *Proceedings of the 10th USENIX Workshop on Offensive Technologies (WOOT 16)*, 2016.

[16] H. Alasmary, A. Khormali, A. Anwar, J. Park, J. Choi, A. Abusnaina, A. Awad, D. Nyang, and A. Mohaisen, "Analyzing and Detecting Emerging Internet of Things Malware: A Graph-based Approach," *IEEE Internet of Things Journal*, 2019.

[17] A. Abusnaina, A. Khormali, H. Alasmary, J. Park, A. Anwar, and A. Mohaisen, "Adversarial learning attacks on graph-based IoT malware detection systems," in *Proceedings of the 39th IEEE International Conference on Distributed Computing Systems, (ICDCS)*, vol. 10, 2019.

[18] H. Alasmary, A. Anwar, J. Park, J. Choi, D. Nyang, and A. Mohaisen, "Graph-based comparison of IoT and android malware," in *International Conference on Computational Social Networks (CSoNet)*. Springer, 2018, pp. 259–272.

[19] K. Kendall and C. McMillan, "Practical malware analysis," in *Black Hat Conference, USA*, 2007, p. 10.

[20] Developers. (2019) The ultimate packer for executables. Available at [Online]: https://upx.github.io/.

[21] A. Moser, C. Kruegel, and E. Kirda, "Limits of static analysis for malware detection," in *Proceedings of the 23rd Annual Computer Security Applications Conference (ACSAC 2007)*. IEEE, 2007, pp. 421–430.

[22] Developers. (2019) VirusTotal. Available at [Online]: https://www.virustotal.com.

[23] ——. (2019) Cyberiocs. Available at [Online]: https://freeiocs.cyberiocs.pro/.

[24] ——. (2019) Radare2. Available at [Online]: https://https://rada.re/r/.

[25] M. Sebastián, R. Rivera, P. Kotzias, and J. Caballero, "AVclass: A tool for massive malware labeling," in *Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses, (RAID)*, 2016, pp. 230–253.

[26] A. Marzano, D. Alexander, O. Fonseca, E. Fazzion, C. Hoepers, K. Steding-Jessen, M. H. Chaves, Í. Cunha, D. Guedes, and W. Meira, "The evolution of Bashlite and Mirai IoT botnets," in *Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2018, pp. 813–818.

[27] C. Aggarwal and K. Srivastava, "Securing IoT devices using SDN and edge computing," in *Proceedings of the 2nd International Conference on Next Generation Computing Technologies (NGCT)*, 2016, pp. 877–882.

[28] M. De Donno, N. Dragoni, A. Giaretta, and A. Spognardi, "Analysis of DDoS-Capable IoT Malwares," in *Proceedings of the Federated Conference on Computer Science and Information Systems, FedCSIS.*, 2017, pp. 807–816.

[29] C. Willems, T. Holz, and F. C. Freiling, "Toward automated dynamic malware analysis using cwsandbox," *IEEE Security & Privacy*, vol. 5, no. 2, pp. 32–39, 2007.

[30] H. Sinanović and S. Mrdovic, "Analysis of Mirai malicious software," in *Proceedings of the 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 2017, pp. 1–5.

[31] E. Cozzi, M. Graziano, Y. Fratantonio, and D. Balzarotti, "Understanding Linux malware," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 161–175.

[32] S. W. Soliman, M. A. Sobh, and A. M. Bahaa-Eldin, "Taxonomy of malware analysis in the IoT," in *Proceedings of the 12th International Conference on Computer Engineering and Systems (ICCES)*. IEEE, 2017, pp. 519–529.

[33] A. Wang, R. Liang, X. Liu, Y. Zhang, K. Chen, and J. Li, "An inside look at IoT malware," in *International Conference on Industrial IoT Technologies and Applications*. Springer, 2017, pp. 176–186.

[34] M. De Donno, N. Dragoni, A. Giaretta, and A. Spognardi, "DDoS-capable IoT malwares: Comparative analysis and mirai investigation," *Security and Communication Networks*, vol. 2018, 2018.